Министерство образования и науки Челябинской области

Государственное бюджетное профессиональное образовательное учреждение

**«Южно-Уральский государственный технический колледж»**

|  |  |
| --- | --- |
| ЗАЩИЩЕНО | |
| ОЦЕНКА: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
|  | Женихова И.Ю. |
|  | Рявкина А.В. |
|  | Ильина О.С. |
|  | Шибанова В.А. |

**ОТЧЕТ**

о прохождении учебной практики УП.05

по ПМ.05 Проектирование и разработка информационных систем

специальность

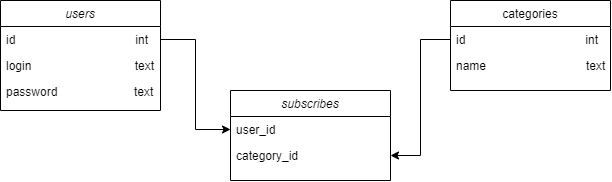
09.02.07 Информационные системы и программирование

(квалификация: разработчик веб и мультимедийных приложений**)**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Разработал: | студент группы ВБ-396/б |
|  | И.В. Иванников |

Челябинск, 2023

В ходе УП была выполнена работа по созданию БД для сервиса выдачи новостей по подпискам пользователя. База включает три таблицы: users (с данными пользователей), categories (с новостными категориями), subscribes (с информацией о подписках пользователя). Схема БД представлена на рисунке.



Сервис позволяет пользователю зарегистрироваться, авторизоваться зарегистрированному пользователю. Независимо от наличия учетной записи, пользователь может добавлять или удалять категории. Авторизованный пользователь может просмотреть свои подписки, подписаться на новостную категорию или отписаться от нее. Также была реализована возможность удаления своей учетной записи.

import sqlite3

connect = sqlite3.connect('bot.db')

cursor = connect.cursor()

# удаление

def checkDel(login, password):

    return cursor.execute('''

        SELECT login, password

        FROM "users"

        WHERE login = ? AND password = ?''', (login, password)).fetchone()

#проверка на пользователя

def check(login):

    return cursor.execute('''

        SELECT login, password

        FROM "users"

        WHERE login = ?''', (login,)).fetchone()

#удаление юзера

def delete(login, password):

    cursor.execute('''

            DELETE FROM users

            WHERE login = ? AND password = ?''', (login, password))

    connect.commit()

#регистрация

def registr(login, password):

    cursor.execute('''

            INSERT INTO "users" (login, password)

            VALUES (?, ?)''', (login, password))

    connect.commit()

#создание таблицы users

resull = cursor.execute('''

CREATE TABLE IF NOT EXISTS

"users"

("id" INTEGER NOT NULL,

"login" TEXT NOT NULL,

"password" TEXT NOT NULL,

primary key("id" AUTOINCREMENT)

)''')

#создание таблицы categories

resull = cursor.execute('''

CREATE TABLE IF NOT EXISTS

"categories"

("id" INTEGER NOT NULL,

"name" TEXT NOT NULL,

primary key("id" AUTOINCREMENT)

)''')

#создание таблицы subscribes

resull = cursor.execute('''

CREATE TABLE IF NOT EXISTS

"subscribes"

("id\_user" INTEGER NOT NULL,

"id\_category" TEXT NOT NULL,

FOREIGN KEY (id\_user) REFERENCES users(id) ON DELETE CASCADE,

FOREIGN KEY (id\_category) REFERENCES categories(id) ON DELETE CASCADE

)''')

connect.commit()

#вывести все категории

def allCategory():

    resull = cursor.execute('''SELECT \*

    FROM categories''').fetchall()

    for category in resull:

        print(f"{category[0]}  - {category[1]}")

#вывести все подписки

def allSub(idPolzovat):

    return cursor.execute('''SELECT subscribes.id\_category, categories.name

    FROM subscribes

    INNER JOIN categories ON subscribes.id\_category=categories.id

    WHERE id\_user = ?''', (idPolzovat,)).fetchall()

#поиск категории по имени

def findCategory(name):

    return cursor.execute('''SELECT \* FROM categories

        WHERE name = ?''', (name,)).fetchone()

#добавление категории

def addCategory(name):

    cursor.execute('''INSERT INTO categories (name)

    VALUES(?)''', (name,))

    connect.commit()

    print(f"Категория {name} добавлена")

#удаление категории

def deleteCategory(name):

    cursor.execute('''DELETE FROM categories

    WHERE id = ?''', (name,))

    connect.commit()

    print("Категория удалена")

#поиск подписки по id

def findSubscribe(idPolzovat, idCategory):

    return cursor.execute('''SELECT \* FROM subscribes

    WHERE id\_user=? AND id\_category = ?''', (idPolzovat, idCategory)).fetchone()

#добавление подписки

def subscribe(idPolzovat, idCategory):

    categories = cursor.execute('SELECT \* FROM categories').fetchall()

    spisok=[]

    for category in categories:

        spisok.append(category[0])

    if(idCategory in spisok):

        cursor.execute('''INSERT INTO subscribes (id\_user, id\_category)

        VALUES (?, ?)''', (idPolzovat, idCategory))

        connect.commit()

        print("Вы подписались")

    else:

        print("Такой категории не существует")

#удаление подписки

def deleteSubscribe(idLogin, idCategory):

    cursor.execute('''DELETE FROM subscribes

    WHERE id\_user = ? AND id\_category = ?''', (idLogin, idCategory))

    connect.commit()

    print("Вы отписались")

#поиск id пользователя

def id(login):

    for i in cursor.execute('''SELECT id FROM users

    WHERE login = ?''', (login,)).fetchone():

        return int(i)

idPolzovat = 0

authorization = int(input('''

1 - авторизоваться

2 - зарегистрироваться

3 - удалить акк '''))

while authorization != -1:

    if (authorization == 1):

        login = input("Введите логин: ")

        password = input("Введите пароль: ")

        resull = check(login)

        if (resull != None):

            if (password == resull[1]):

                idPolzovat = id(login)

                print("Вход выполнен")

                if (idPolzovat > 0):

                    process = int(input('''

                    1 - посмотреть подписки

                    2 - подписаться на категорию

                    3 - отписаться от категории

                    4 - добавить категорию

                    5 - удалить категорию

                    6 - выйти:'''))

                    while (process != -1):

                        if (process == 1):

                            print(allSub(idPolzovat))

                        if (process == 2):

                            allCategory()

                            category = int(input('Вы хотите подписаться на - '))

                            find = findSubscribe(idPolzovat, category)

                            print(idPolzovat,category,find)

                            if find is None:

                                subscribe(idPolzovat, category)

                            else:

                                print('Вы уже подписаны на данную категорию')

                        if (process == 3):

                            print(allSub(idPolzovat))

                            category = int(input("Выберите категорию: "))

                            deleteSubscribe(idPolzovat, category)

                        if (process == 4):

                            category = input("Название: ")

                            find = findCategory(category)

                            if(find==None):

                                addCategory(category)

                            else:

                                print('Категория уже добавлена')

                        if (process == 5):

                            category = input("Номер категории: ")

                            deleteCategory(category)

                        if (process == 6):

                            idPolzovat = 0

                            actionAuth = 0

                            continue

                        process = int(input('''

                            1 - посмотреть подписки

                            2 - подписаться на категорию

                            3 - отписаться от категории

                            4 - добавить категорию

                            5 - удалить категорию

                            6 - выйти:'''))

            else:

                print("Пароль неверный")

        else:

            print("Вы не зарегистрированы")

    if (authorization == 2):

        login = input("Введите логин: ")

        password = input("Ведите пароль: ")

        register = check(login)

        if (register != None):

            print("Пользователь уже существует")

        else:

            registr(login, password)

            print("Вы зарегистрировались")

    if (authorization == 3):

        login = input("Введите логин:")

        password = input("Введите пароль: ")

        delete = checkDel(login, password)

        if (delete != None):

            delete(login, password)

            print("Пользователь удален")

        else:

            print("Пароль или логин не верны")

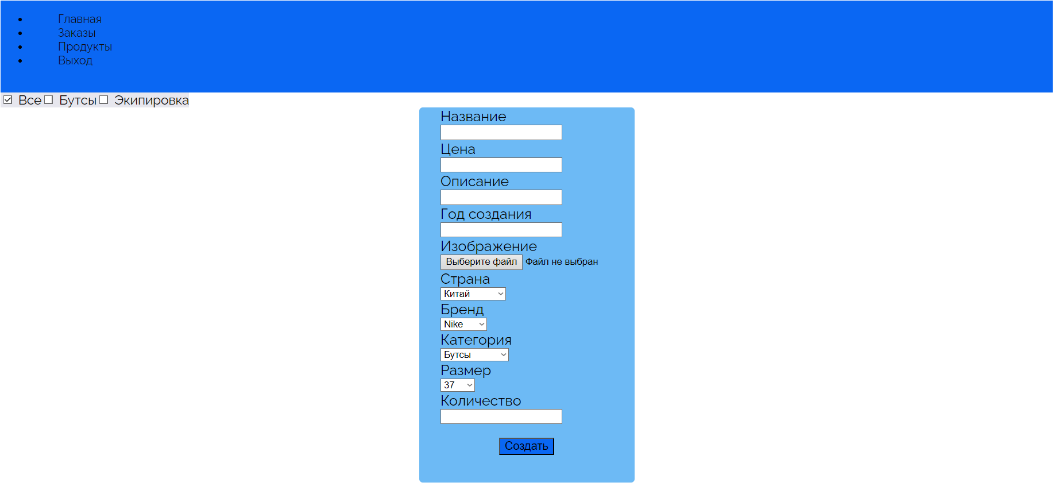
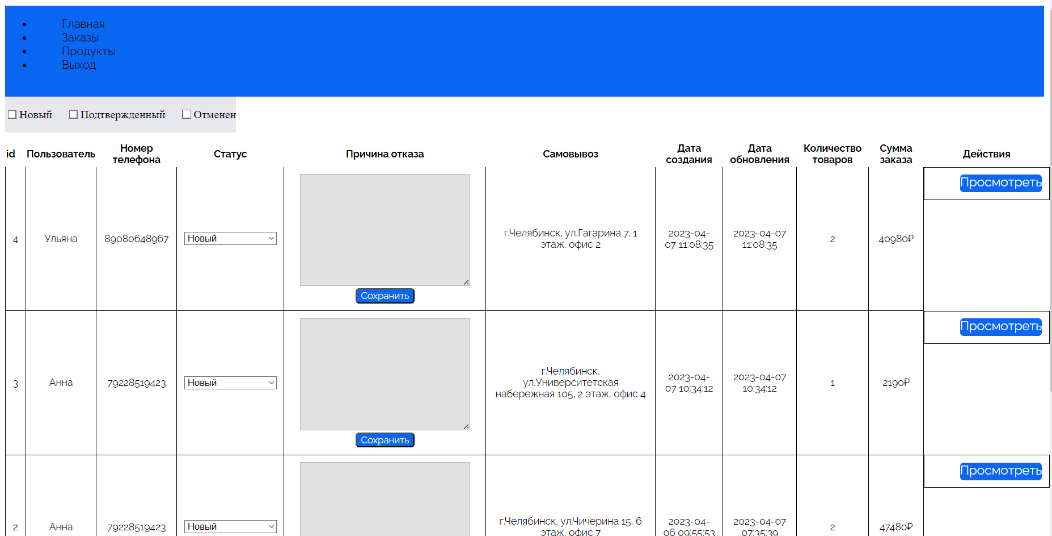
cursor.close()

**Тема: разработка приложений для работы с ИС**

**Задание**: разработать панель администратора сайта для интернет-ресурса footballStreat.

**Решение**:

1. Вид панели администратора (скриншоты различных страниц)



1. Программный код

**auth.view.php**

<?php require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/views/templates/header.admin.php'; ?>

<br>

<form class="createF" action="/app/admin/autnCheck.php" method="POST">

    <label for="phone">

        Введите логин

        <input type="phone" class="form-control" name="phone">

    </label>

    <label for="password" class="form-label">Введите пароль

        <input type="password" name="password" class="form-control" id="password">

    </label>

    <?php if (!empty($\_SESSION["error"])) : ?>

        <p class="error"><?= $\_SESSION["error"] ?></p>

    <?php endif ?>

    <br><br>

    <div class="form-group">

        <button name="admin">Войти</button>

    </div>

</form>

<?php unset($\_SESSION["error"]); ?>

</body>

</html>

**orders.view.php**

<?php require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/views/templates/header.admin.php'; ?>

<div class="conteiner">

    <div class="cont cont\_panel">

        <div class="nav-order">

            <div class="chec-flex">

                <form action="" class="chec">

                    <?php foreach ($statuses as $status) : ?>

                        <div class="input">

                            <input type="checkbox" <?= isset($\_GET['status']) && $\_GET['status'] == $status->id ? 'checked' : '' ?> onchange="this.form.submit()" name="status" value="<?= $status->id ?>"></input>

                            <p for=""><?= $status->name ?></p>

                        </div><?php endforeach ?>

                </form>

            </div>

        </div>

    </div>

</div>

<div class="conteiner\_orders">

    <table>

        <tr>

            <th>id</th>

            <th>Пользователь</th>

            <th>Номер телефона</th>

            <th>Статус</th>

            <th>Причина отказа</th>

            <th>Самовывоз</th>

            <th>Дата создания</th>

            <th>Дата обновления</th>

            <th>Количество товаров</th>

            <th>Сумма заказа</th>

            <th>Действия</th>

        </tr>

        <?php foreach ($orders as $order) : ?>

            <tr class="block">

                <td><?= $order->id ?></td>

                <td><?= $order->person\_name ?></td>

                <td><?= $order->phone ?></td>

                <td><select class="select" data-order-id="<?= $order->id ?>" name="" id="select<?= $order->id ?>">

                        <?php foreach ($statuses as $status) : ?>

                            <option class="state" value="<?= $status->id ?>" <?= $status->id == $order->status\_id ? "selected" : "" ?>><?= $status->name ?></option>

                        <?php endforeach ?>

                    </select></td>

                <td>

                    <form action="/app/admin/change.php" method="POST">

                        <textarea disabled name="reason\_cancel" data-textarea="<?= $order->id ?>" id="order\_id" class="reason\_cancel" cols="30" rows="10"><?= $order->reason\_cancel ?></textarea>

                        <button value="<?= $order->id ?>" name="saveAS" class="save" data-order-id="<?= $order->id ?>">Сохранить</button>

                </td>

                </form>

                <td><?= $order->delivery ?></td>

                <td><?= $order->date\_order ?></td>

                <td><?= $order->updated\_at ?></td>

                <td><?= $order->count ?></td>

                <td><?= $order->sum ?>₽</td>

                <td class="doing">

                    <a href="/app/admin/productsByOrder.php?id=<?= $order->id ?>" class="pros">Просмотреть</a>

                </td>

            </tr>

        <?php endforeach; ?>

    </table>

</div>

<script src="/accers/js/fetch.js"></script>

<script src="/accers/js/LoadOrders.js"></script>

**products.view.php**

<?php

session\_start();

if (!isset($\_SESSION["admin"]) || !$\_SESSION["admin"]) {

    header("Location: /app/admin/auth.php");

    die();

}

require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/views/templates/header.admin.php'; ?>

<div class="conteiner">

    <div class="cont cont\_panel">

        <div class="block">

            <form action="/app/admin/search.php" method="POST" class="conteiner\_pos category">

                <div class="form-check">

                    <input class="form-check-input" type="checkbox" name="all" id="all" checked value="all">

                    <label class="form-check-label" for="all">

                        Все

                    </label>

                </div>

                <?php foreach ($categories as $category) : ?>

                    <div class="form-check">

                        <input class="form-check-input" type="checkbox" name="category" value="<?= $category->id ?>" id="<?= $category->id ?>">

                        <label class="form-check-label" for="<?= $category->id ?>">

                            <?= $category->name ?>

                        </label>

                    </div><?php endforeach ?>

            </form>

        </div>

    </div>

</div>

</div>

</form>

<div class="conteiner\_pos">

    <form action="/app/admin/create.php" method="POST">

        <div class="new">

            <div class="new\_group">

                <label for="name">Название</label>

                <input class="inp" type="text" name="name">

            </div>

            <div class="new\_group">

                <label for="price">Цена</label>

                <input class="inp" type="text" name="price">

            </div>

            <div class="new\_group">

                <label for="description">Описание</label>

                <input class="inp" type="text" name="description">

            </div>

            <div class="new\_group">

                <label for="release\_year">Год создания</label>

                <input class="inp" type="number" name="year\_release">

            </div>

            <div class="new\_group">

                <label for="photo">Изображение</label>

                <input class="inp" type="file" name="photo">

            </div>

            <div class="new\_group">

                <label for="country">Страна</label>

                <select class="inp" name="country\_id" id="country">

                    <?php foreach ($countries as $country) : ?>

                        <option value="<?= $country->id ?>"><?= $country->name ?></option>

                    <?php endforeach; ?>

                </select>

            </div>

            <div class="new\_group">

                <label for="brand">Бренд</label>

                <select class="inp" name="brand\_id" id="brand">

                    <?php foreach ($brands as $brand) : ?>

                        <option value="<?= $brand->id ?>"><?= $brand->name ?></option>

                    <?php endforeach; ?>

                </select>

            </div>

            <div class="new\_group">

                <label for="category">Категория</label>

                <select class="inp" name="category\_id" id="category">

                    <?php foreach ($categories as $category) : ?>

                        <option value="<?= $category->id ?>"><?= $category->name ?></option>

                    <?php endforeach; ?>

                </select>

            </div>

            <div class="new\_group">

                <label for="table\_size\_id">Размер</label>

                <select class="inp" name="table\_size\_id" id="size">

                    <?php foreach ($sizes as $size) : ?>

                        <option value="<?= $size->id ?>"><?= $size->size ?></option>

                    <?php endforeach; ?>

                </select>

            </div>

            <div class="new\_group">

                <label for="quantity">Количество</label>

                <input class="inp" type="number" name="quantity">

            </div><br>

            <div class="new\_group btn\_prod">

                <button class="create btn" name="create">Создать</button>

            </div>

            <br><br>

        </div>

    </form>

</div>

<table>

    <tr>

        <th>id</th>

        <th>Название</th>

        <th>Цена</th>

        <th>Год создания</th>

        <th>Изображение</th>

        <th>Страна</th>

        <th>Категория</th>

        <th>Бренд</th>

        <th>Действие</th>

    </tr>

    <tbody class="block cont\_product">

    </tbody>

</table>

</div>

<script src="/accers/js/fetch.js"></script>

<script src="/accers/js/loadProducts.js"></script>

<script src="/accers/js/loadProductsTovar.js"></script>

**productsByOrder.view.php**

<?php require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/views/templates/header.admin.php'; ?>

<table class="conteiner-orderss">

    <tr>

        <th>№</th>

        <th>Пользователь</th>

        <th>Изображение</th>

        <th>Название</th>

        <th>Размер</th>

        <th>Количество</th>

        <th>Цена</th>

    </tr>

    <tbody>

        <?php foreach ($productsInOrder as $product) : ?>

            <tr>

                <td><?= $product->order\_id ?></td>

                <td><?=$product->pers\_name?><br> <?=$product->mail?></td>

                <td><img src="/upload/<?= $product->image ?>" alt="image" width="50"></td>

                <td><?= $product->name ?></td>

                <td><?= $product->size ?></td>

                <td><?= $product->product\_count ?></td>

                <td><?= $product->price\_product\_order ?>₽</td>

            </tr>

        <?php endforeach ?>

    </tbody>

</table>

<?php require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/views/templates/footer.php'; ?>

**profile.view.php**

<?php require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/views/templates/header.admin.php'; ?>

<div class="conteiner">

    <div class="cont cont\_panel">

    </div>

</div>

</div>

**update.view.php**

<?php

use App\models\Quantity;

require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/views/templates/header.admin.php'; ?>

<form action="/app/admin/update.php" class="updateProduct" method="POST" enctype="multipart/form-data">

    <input type="hidden" class="form-control" name="id" value="<?= $product->id ?>">

    <div class="div-update-input">

        <label class="label-form" for="name">Введите название: </label>

        <input type="text" class="form-control" name="name" value="<?= $product->name ?>">

    </div>

    <div class="div-update-input">

        <label class="label-form" for="price">Введите цену: </label>

        <input type="number" class="form-control" name="price" value="<?= $product->price ?>">

    </div>

    <div class="div-update-input">

        <label class="label-form" for="description">Введите описание: </label>

        <input type="text" class="form-control" name="description" value="<?= $product->description ?>">

    </div>

    <div class="div-update-input">

        <label class="label-form" for="release">Введите год релиза: </label>

        <input type="number" class="form-control" name="year\_release" value="<?= $product->year\_release ?>">

    </div>

    <div class="div-update-input">

        <label class="label-form" for="image">Выберите картинку: </label>

        <input type="file" class="form-control" name="photo" id="file">

        <img src="/upload/<?= $product->photo != null ? $product->photo : '' ?>" alt="" id="loadedImg" width="80" height="80">

    </div>

    <div class="div-update-input">

        <label class="label-form" for="country">Выберите страну: </label><select name="country\_id" class="form-control" id="">

            <?php foreach ($countries as $country) : ?>

                <option value="<?= $country->id ?>" <?= $country->id == $product->country\_id ? 'selected' : ' ' ?>><?= $country->name ?></option>

            <?php endforeach ?>

        </select>

    </div>

    <div class="div-update-input">

        <label class="label-form" for="category">Выберите категорию: </label><select name="category\_id" class="form-control" id="">

            <?php foreach ($categories as $category) : ?>

                <option value="<?= $category->id ?>" <?= $category->id == $product->category\_id ? 'selected' : ' ' ?>> <?= $category->name ?></option>

            <?php endforeach ?>

        </select>

    </div>

    <div class="div-update-input">

        <label class="label-form" for="category">Выберите бренд: </label><select name="brand\_id" class="form-control" id="">

            <?php foreach ($brands as $brand) : ?>

                <option value="<?= $brand->id ?>" <?= $brand->id == $product->brand\_id ? 'selected' : ' ' ?>> <?= $brand->name ?></option>

            <?php endforeach ?>

        </select>

    </div>

    <div class="div-update-input">

        <label class="label-form" for="category">Выберите размер: </label><select name="size\_id" class="form-control" id="">

            <?php foreach ($sizes\_product as $size) : ?>

                <option value="<?= $size->id ?>"> <?= $size->size ?> </option>

            <?php endforeach ?>

        </select>

        <?php foreach ($sizes\_product as $size) : ?>

            <?php foreach (Quantity::quantity($product->id, $size->ptr) as $count) : ?>

                <?php var\_dump($count->quantity) ?>

                <input type="number" class="form-control" name="price" value="<?= $count->quantity ?>">

            <?php endforeach ?>

        <?php endforeach ?>

    </div>

    <!-- <?php foreach ($counts as $count) : ?>

        <div class="div-update-input">

        <label class="label-form" for="price">Введите количество: </label>

        <?php endforeach ?> -->

    </div>

    <button class="button-save-up" name="save" for="image" value="<?= $product->id ?>">Сохранить</button>

</form>

**create.php**

<?php

use App\models\Product;

require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/bootstrap.php';

if (isset($\_POST['create'])) {

    $product = Product::add($\_POST['name'], $\_POST['price'], $\_POST['photo'], $\_POST['description'], $\_POST['year\_release'], $\_POST['category\_id'], $\_POST['brand\_id'], $\_POST['country\_id'], $\_POST['quantity'], $\_POST['table\_size\_id']);

}

header('Location: /app/admin/products.php');

**orders.php**

<?php

use App\models\Order;

$style = '/accers/css/admin.css';

require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/bootstrap.php';

$orders = Order::all();

$statuses = Order::AllOrderStatuses();

if (!empty($\_GET) && isset($\_GET["status"])) {

    if ($\_GET["status"] !== 'all') {

    $orders = Order::ordersByManyStatuses($\_GET["status"]);

    }

    } else {

    }

require\_once $\_SERVER['DOCUMENT\_ROOT'] . '/views/admin/orders.view.php';

**products.php**

<?php

use App\models\Category;

use App\models\Product;

use App\models\Country;

use App\models\Brand;

use App\models\Size;

$style = '/accers/css/admin.css';

include $\_SERVER["DOCUMENT\_ROOT"] . "/bootstrap.php";

$stream = file\_get\_contents("php://input");

if ($stream != null) {

 $id = json\_decode($stream)->id;

 $delete = Product::delete($id);

 echo json\_encode( $delete, JSON\_UNESCAPED\_UNICODE);

}

$categories = Category::all();

$countries = Country::all();

$brands = Brand::BrandAll();

$sizes = Size::sizeAll();

include $\_SERVER['DOCUMENT\_ROOT'] . '/views/admin/products.view.php';

**productsByOrder.php**

<?php

use App\models\Order;

$style = '/accers/css/admin.css';

require\_once $\_SERVER["DOCUMENT\_ROOT"] . "/bootstrap.php";

$productsInOrder = Order::productsByOrder(($\_GET["id"]));

require\_once $\_SERVER["DOCUMENT\_ROOT"] . "/views/admin/productsByOrder.view.php";

**update2.php**

<?php

use App\models\Category;

use App\models\Country;

use App\models\Product;

use App\models\Brand;

use App\models\Size;

use App\models\Quantity;

$style = '/accers/css/admin.css';

include $\_SERVER["DOCUMENT\_ROOT"] . "/bootstrap.php";

if (!isset($\_SESSION["admin"]) || !$\_SESSION["admin"]) {

    header("Location: /app/admin/auth.php");

    die();

}

$product = Product::find($\_GET["id"]);

$categories = Category::all();

$countries = Country::all();

$brands = Brand::BrandAll();

$sizes = Size::sizeAll();

$sizes\_product=Size::sizeProduct($product->id);

require\_once $\_SERVER["DOCUMENT\_ROOT"] . "/views/admin/update.view.php";

**update.php**

<?php

use App\models\Product;

include $\_SERVER["DOCUMENT\_ROOT"] . "/bootstrap.php";

$extensions = ["jpg", "jpeg", "png", "webp", "jfif"];

$types = ["photo/jpeg", "photo/png", "photo/webp", "photo/jfif"];

if ($\_FILES["photo"]["name"] != "") {

    $name = $\_FILES["photo"]["name"];

    $tmpName = $\_FILES["photo"]["tmp\_name"];

    $error = $\_FILES["photo"]["error"];

    $size = $\_FILES["photo"]["size"];

    $path\_parts = pathinfo($name);

    $img = $name;

    if ($error == 0) {

        if ($size > 3000000) {

            $\_SESSION["error"] = "Файл слишком большой";

        } else {

            if (!move\_uploaded\_file($tmpName, "D:/OSPanel/domains/football-streat/upload/" . $img)) {

                $\_SESSION["error"] = "Файл не загрузился";

                if (isset($\_POST["save"])) {

                    if ($\_POST["id"] != null) {

                        Product::update($\_POST, $img);

                    }

                }

            };

        }

    } else {

        $\_SESSION["error"] = "Ошибка";

    };

} else {

    Product::updateNoImage($\_POST);

};

header("Location: /app/admin/products.php");

search.php

<?php

use App\models\Product;

include $\_SERVER["DOCUMENT\_ROOT"] . "/bootstrap.php";

if (isset($\_GET["category"])) {

    $categories = json\_decode($\_GET["category"]);

    if (empty($categories) || $categories == "all") {

        $products = Product::all();

    } else {

        $products = Product::productsByManyCategories($categories);

    }

    echo json\_encode($products, JSON\_UNESCAPED\_UNICODE);

}

**loadProductsTovar.js**

document.addEventListener('DOMContentLoaded', () => {

    let productsConteiner = document.querySelector('.cont\_product')

    let categoryChecks = document.querySelectorAll("[name='category']")

    let allProduct = []

    categoryChecks.forEach(item => {

        allProduct.unshift(item.value)

    })

    getProductsByCategory(allProduct)

    //отрабатываем выбор категории

    categoryChecks.forEach(item => {

        item.addEventListener('change', async (e) => {

            //формируем массив по id категорий по выбранным флажкам

            let categoriesCheck = [...categoryChecks].filter(item => item.checked).map(item => item.value)

            //получаем товары выбранных категорий

            await getProductsByCategory(categoriesCheck)

        })

    });

    async function getProductsByCategory(categories) {

        let param = new URLSearchParams()

        param.append('category', JSON.stringify(categories))

        let products = await getData('/app/admin/search.php', param);

        showProducts(products)

    }

    //вывод карточек на страницу

    function showProducts(products) {

        productsConteiner.innerHTML = ''

        products.forEach(product => {

            productsConteiner.insertAdjacentHTML('beforeend', getOneCard(product))

        })

    }

    //формирование одной карточки

    function getOneCard({ id, name, photo, price,year\_release, country, category,brand}) {

        return `<tr class='tr\_block'>

        <td>${id}</td>

        <td> ${name}</td>

        <td> ${price}</td>

        <td> ${year\_release}</td>

        <td> <img class="img-tov-adm" src=/upload/${photo}></td>

        <td> ${country}</td>

        <td> ${category}</td>

        <td> ${brand}</td>

        <td class="doing">

            <button class="delete" data-image="${photo}"  data-id="${id}">Удалить</button>

            <a href="/app/tables/products/show.php?id=${id}" class="btn btn-primary delete">Подробнее</a>

            <a href="/app/admin/update2.php?id=${id}" class="btn btn-primary delete update">Редактировать</a>

        </td>

        </tr>`

    }

})

**Вывод**: за время учебной практики был выполнен ряд работ:

* разработан дружественный графический интерфейс;
* выполнено программирование в соответствии с требованиями технического задания;
* реализован многопользовательский режим работы приложения;
* выполнены тестирование и отладка приложения.

https://www.figma.com/files/recent?fuid=1155110030346316277

Разработка конфигурации для учета посещений клиентами гаражей

