

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ» (ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук  
Кафедра информационных систем и технологий

Сайт продуктового онлайн-магазина

Курсовой проект

09.03.02 Информационные системы и технологии  
Информационные системы в телекоммуникациях

Зав. кафедрой \_\_\_\_\_ доц., канд. техн. наук Д.Н. Борисов

Руководитель \_\_\_\_\_ В.С. Тарасов

Обучающиеся:

\_\_\_\_\_ Митин В.С.  
\_\_\_\_\_ Шкурей А.А.

Воронеж 2022

## Оглавление

Введение .....	3
1. Постановка задачи.....	4
2. Анализ предметной области .....	5
2.1. Глоссарий.....	5
2.2. Анализ существующих решений .....	5
2.3. Анализ задачи.....	6
2.3.1. Варианты использования приложения .....	6
2.3.2. Взаимодействие компонентов системы.....	7
2.3.3. Варианты состояния системы .....	8
2.3.4. Развёртывание приложения .....	8
3. Анализ средств реализации .....	9
4. Реализация.....	10
4.1. Сущности.....	10
4.2. Графический интерфейс.....	12
5. Тестирование.....	13
5.1. Ограничения целостности .....	13
5.2. Обработка введенных данных .....	13
Заключение.....	14
Список использованных источников .....	15
Приложения.....	16
Приложение А.....	16
Приложение В.....	17

## **Введение**

Информационные технологии стали неотъемлемой частью повседневной жизни. С каждым днем в России становится все больше людей, которые начинают пользоваться Интернетом. А если есть спрос, то появляются и предложения. Интернет-технологии позволяют развивать и интернет-услуги.

Сервис и качество обслуживания в интернет-магазинах набирает свою силу. Сервис, который предлагают современные интернет магазины оставляет самые положительные эмоции у каждого покупателя. Каждый уважающий себя интернет магазин стремиться к индивидуальному подходу к каждому клиенту и к гибкости в работе, что позволит вам совершать покупки в интернет-магазине быстро и удобно.

В настоящее время подавляющее большинство магазинов имеют свой сайт, который позволяет заказывать товары дистанционно. Однако ни один из рассмотренных нами сайтов не имеет функции доставки продукции каждую неделю в определенный день. В разрабатываемом нами сайте будет реализована данная особенность.

Целью данного курсового проекта является создание сайта продуктового онлайн-магазина, с помощью которого пользователи могли бы приобретать продукты, не выходя из дома.

## 1. Постановка задачи

Цель данной курсовой работы — реализовать сайт, отвечающий следующим требованиям:

### 1. Безопасность:

1.1. Сайт не должен давать доступ к основному функционалу неавторизованным пользователям.

### 2. Численность и квалификация персонала:

2.1. Команда состоит из двух человек, между которыми распределены задачи разработчика и техрайтера.

### 3. Надежность системы:

3.1. При некорректном вводе информации, либо при возникновении ошибки сайт должен выводить соответствующее сообщение.

3.2. Ошибки и сбои не должны приводить к повреждению данных и работоспособности сайта.

### 4. Удобный пользовательский интерфейс:

4.1. Связность всех html-страниц и возможность удобного перехода между ними.

4.2. Оформление пользовательского интерфейса в едином стиле.

### 5. Функционал:

5.1. Регистрация и авторизация пользователей.

5.2. Разделение пользователей по ролям:

5.2.1. Незарегистрированные пользователи, не имеющие доступа к функционалу сайта.

5.2.2. Зарегистрированные пользователи, прошедшие авторизацию и имеющие полный доступ к функционалу сайта.

5.2.3. Администраторы, имеющие расширенный доступ (возможность просматривать и удалять пользователей, менять статус заказов, а также просматривать, удалять и добавлять товары, менять их цену и описание) к функционалу системы.

5.3. Поиск товаров (через поисковую строку или через сортировку по цене).

5.4. Добавление товаров в корзину.

5.5. Заказ товаров из корзины.

- 5.6. Оформление обычного заказа.
- 5.7. Оформление серийного заказа.
- 5.8. Просмотр информации об оформленных заказах.

## **2. Анализ предметной области**

### **2.1. Глоссарий**

БД — база данных.

Неавторизованный пользователь — пользователь, имеющий учетную запись в БД сайта, но не прошедший авторизацию, либо пользователь, не имеющий учетной записи в БД сайта.

Авторизованный пользователь — пользователь, имеющий учетную запись в БД сайта и успешно прошедший авторизацию.

Техрайтер — лицо, выполняющие написание содержательной части документации.

### **2.2. Анализ существующих решений**

#### **1) «Перекресток»**

Достоинства:

- Широкий ассортимент
- Возможность бесконтактной доставки через сервис «Перекресток.Быстро»

Недостатки:

- Минимальная сумма заказа – 2000 руб.

#### **2) Пятерочка**

Достоинства:

- Возможность заказа через мобильное приложение «Доставка Пятерочка»
- Отсутствие минимальной суммы заказа

Недостатки:

- Платная доставка

## 2.3. Анализ задачи

### 2.3.1. Варианты использования приложения

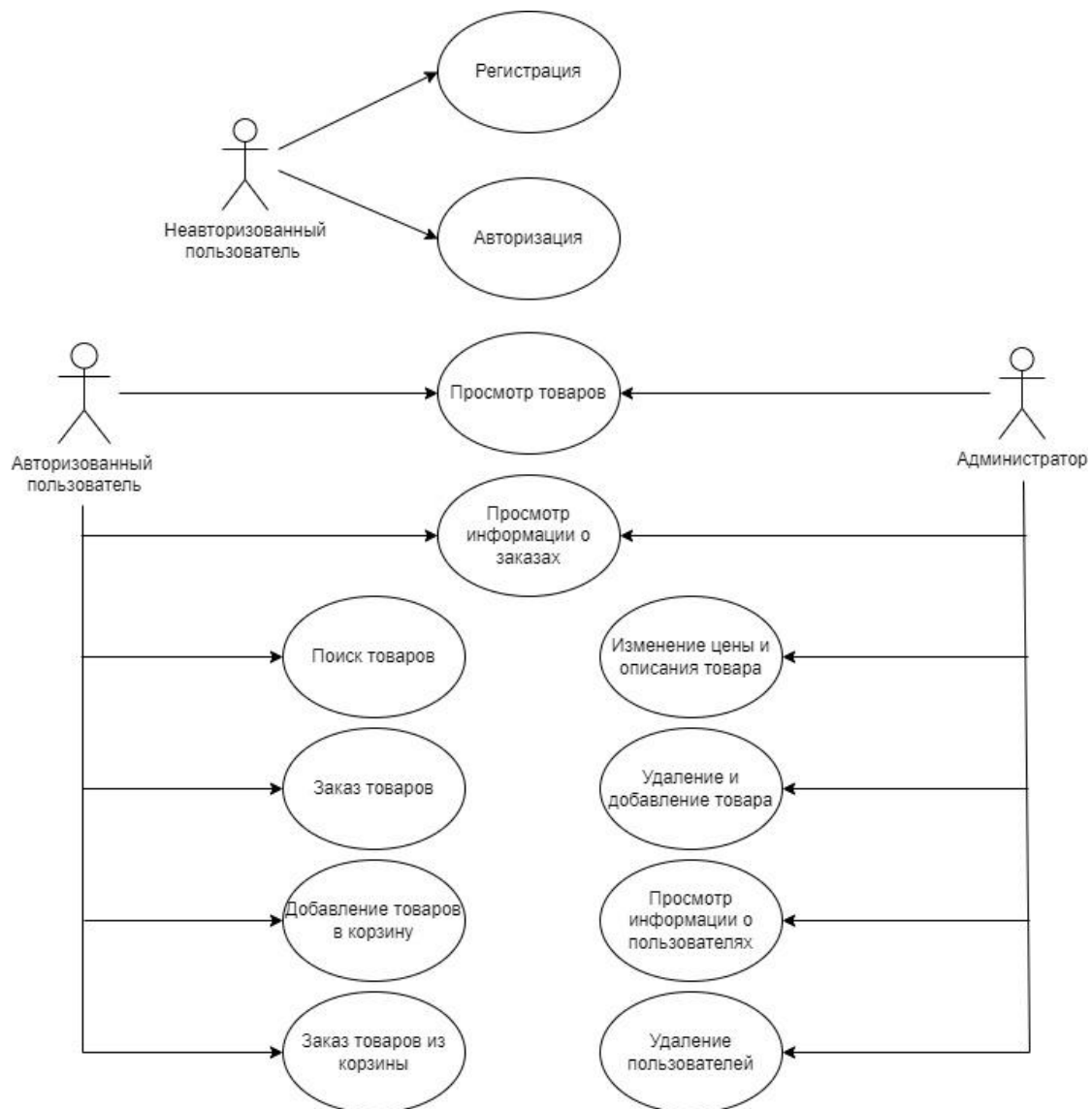


Рисунок 1 – Диаграмма вариантов использования

При взаимодействии с сайтом неавторизованный пользователь имеет следующие возможности (рис. 1):

- Регистрация
- Авторизация

Список возможностей авторизованного пользователя:

- Просмотр товаров
- Просмотр информации о своих заказах

- Поиск товаров
- Заказ товаров
- Добавление товаров в корзину
- Заказ товаров из корзины

Список возможностей администратора:

- Просмотр товаров
- Просмотр информации обо всех заказах
- Изменение цены и описания товара
- Удаление и добавление товара
- Просмотр информации о пользователях
- Удаление пользователей

### 2.3.2. Взаимодействие компонентов системы

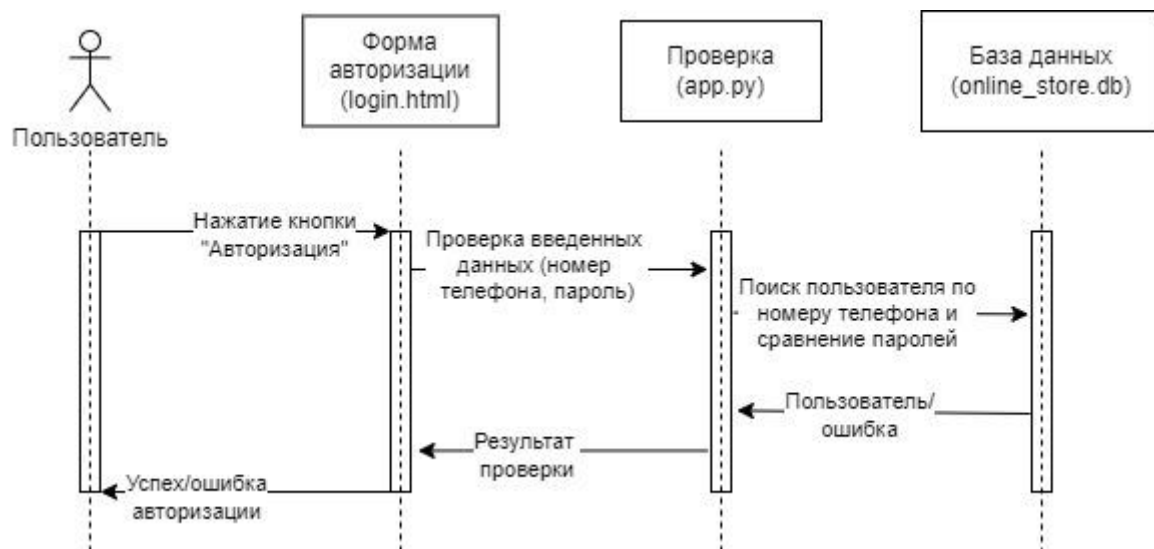


Рисунок 2 – Диаграмма последовательности

На рисунке 2 изображена последовательность взаимодействий пользователя и сайта при авторизации. Сначала пользователь заходит на необходимую страницу и заполняет поля. После нажатия кнопки «Отправить» на сервере происходит проверка корректности введенных данных. После успешной проверки в базе данных ищется пользователь с

указанным номером телефона. Если он находится, и пароли совпадают, авторизация завершается успешно.

### 2.3.3. Варианты состояния системы



Рисунок 3 – Диаграмма состояний

На Рисунке 3 показаны возможные состояния системы. В их число входит:

- Регистрация (если пользователь не имеет учетной записи)
- Авторизация (если пользователь был зарегистрирован ранее)
- Оформление заказа
- Доставка

### 2.3.4. Развёртывание приложения



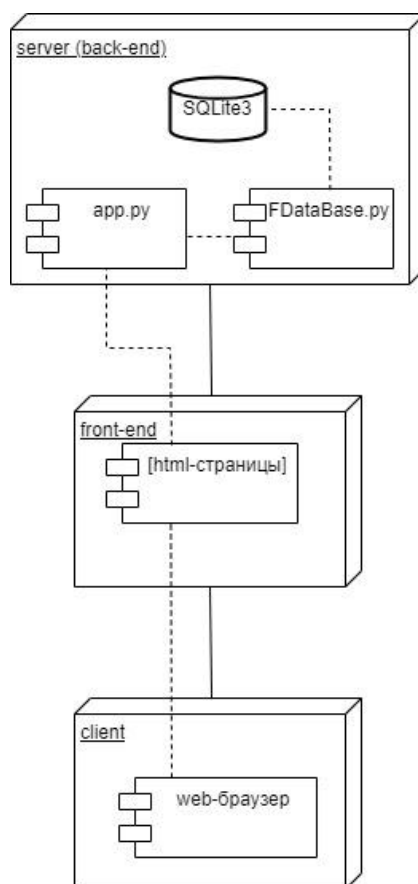


Рисунок 4 – Диаграмма развертывания

На Рисунке 4 представлена диаграмма развертывания. Обращение пользователя к сайту происходит через web-браузер, который в свою очередь отображает html-страницы приложения. Они представляют собой front-end.

Главным файлом приложения является файл `app.py`. Он обрабатывает всю информацию, поступающую с html-страниц.

Также есть класс `FDataBase`. В нем содержатся все необходимые методы, которые использует главный файл для работы с БД (выборка, добавление, удаление, изменение данных).

### 3. Анализ средств реализации

В качестве средств реализации использованы следующие технологии:

1. Язык программирования Python. Он имеет высокую скорость разработки, большое разнообразие библиотек, логичный и простой синтаксис.

2. Фреймворк Flask для создания web-приложений. Его главным плюсом является понятность и простота в использовании. Кроме того, его освоение не занимает много времени.

3. Язык разметки html. Он был использован для создания front-end части приложения, т.к. он имеет небольшой вес и экономичен в расходе ресурсов сервера.

4. СУБД SQLite3. Данная БД имеет высокую скорость и малый размер. Помимо того она технически хорошо согласована с языком программирования Python и Фреймворком Flask.

## 4. Реализация

### 4.1. Сущности

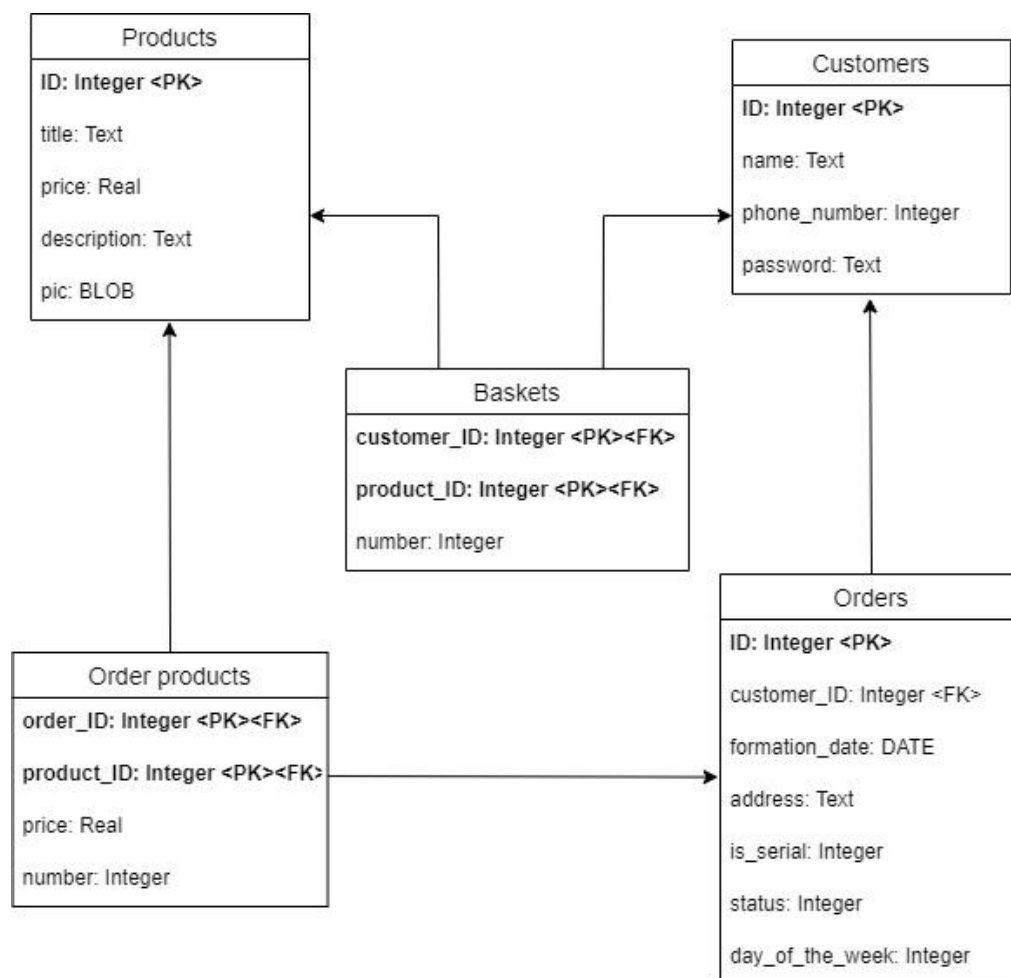


Рисунок 4 – Диаграмма классов

На Рисунке 4 изображены таблицы (классы) БД сайта и их связи.

1. Класс «Products». В нем представлены продукты, доступные пользователям для заказа. Класс имеет следующие свойства:

- 1.1. ID – уникальный идентификатор;
- 1.2. title – название продукта;
- 1.3. price – цена продукта;
- 1.4. description – описание продукта;
- 1.5. pic – картинка (в бинарном виде);

2. Класс «Customers». Он содержит информацию о пользователях сайта:

- 2.1. ID – уникальный идентификатор;
- 2.2. name – ФИО пользователя;
- 2.3. phone\_number – номер телефона;
- 2.4. password – пароль;

3. Класс «Baskets». Он имеет составной первичный ключ: ID пользователя и ID продукта. Таким образом, каждая запись данной таблицы содержит определенный товар в корзине у определенного пользователя.

- 3.1. customer\_ID – уникальный идентификатор пользователя;
- 3.2. product\_ID – уникальный идентификатор продукта;
- 3.3. number – количество;

4. Класс «Orders». Содержит информацию о заказе.

- 4.1. ID – уникальный идентификатор
- 4.2. customer\_ID – уникальный идентификатор пользователя
- 4.3. formation\_date – дата формирования
- 4.4. address – адрес доставки
- 4.5. is\_serial – является ли заказ серийным (0 – нет, 1 – да)

4.6. status – статус (0 – в процессе отправки (для серийных), 1 – отправлено, 2 – доставлено)

4.7. day\_of\_the\_week – день недели для серийной доставки (0 – понедельник, 6 – воскресенье, -1 – для обычных заказов)

5. Класс «Order products». В нем находятся данные о товарах заказа.

5.1. order\_ID – уникальный идентификатор заказа

5.2. product\_ID – уникальный идентификатор продукта

5.3. price – цена

5.4. number – количество

## 4.2. Графический интерфейс

Дизайн сайта базируется на двух составляющих: это картинка, выступающая в качестве заднего фона (установлена с помощью css-файла), и блок «div», в котором выводится вся информация.

Графический интерфейс сайта в основном состоит из полей для ввода данных и кнопок, отправляющих их на сервер. Также есть различные кнопки навигации между html-страницами. Например, для авторизованного пользователя есть кнопки для выхода из профиля, просмотра заказов и корзины (рис. 5).

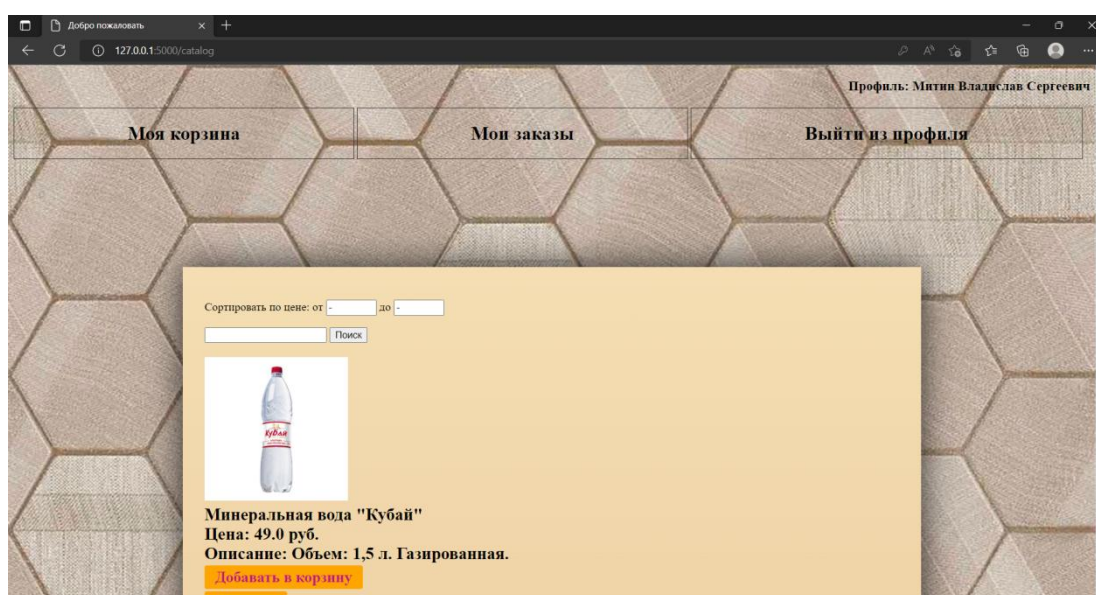


Рисунок 5 – Интерфейс авторизованного пользователя

## **5. Тестирование**

### **5.1. Ограничения целостности**

При работе с БД сайта возникли ошибки, связанные с ограничением целостности данных. Это происходило, когда вызывался метод, удаляющий запись из таблицы «Customers». Информация об удаляемом пользователе оставалась в других таблицах в виде внешних ключей.

Данная проблема была успешно решена путем отладки кода соответствующего метода (Приложение А).

### **5.2. Обработка введенных данных**

При тестировании приложения были выявлены многочисленные погрешности, связанные с некорректной обработкой данных, введенных пользователем.

По большей части они были связаны с вызовом неверных исключений (try-except) при вводе неточных данных, вследствие чего пользователь получал сообщения, не относящиеся к произошедшей ошибке.

Проблема также была успешно решена. В Приложении В показан соответствующий пример.

## **Заключение**

В ходе данного курсового проекта было сделано следующее:

- Были рассмотрены существующие сайты продуктовых магазинов, их преимущества и недостатки
- Рассмотрены варианты использования приложения
- Рассмотрено взаимодействие компонентов системы
- Рассмотрены варианты состояния системы
- Было описано развертывание приложения
- Были проанализированы средства реализации, рассмотрены классы БД сайта и графический интерфейс
- Было проведено тестирование

В результате проведенной работы был создан сайт, отвечающий поставленным требованиям.

## Список использованных источников

- 1) Почему покупать товары в интернет-магазине удобно и выгодно? [электронный курс] – Режим доступа: <https://detimd.ru/blog/pochemu-pokupat-tovary-v-internet-magazine-udobno-i-vygodno/> – Заглавие с экрана. – (Дата обращения: 27.08.2022).
- 2) Топ-10 продуктовых онлайн-ритейлеров [электронный курс] – Режим доступа: [https://new-retail.ru/business/e\\_commerce/top\\_10\\_produktovykh\\_onlayn\\_riteylerov1934/](https://new-retail.ru/business/e_commerce/top_10_produktovykh_onlayn_riteylerov1934/) – Заглавие с экрана. – (Дата обращения: 2.09.2022).
- 3) Плюсы и минусы Python [электронный курс] – Режим доступа: <https://brainskills.ru/blog/plyusy-i-minusy-python/> – Заглавие с экрана. – (Дата обращения: 13.09.2022).
- 4) HTML или CMS – что лучше для коммерческого сайта [электронный курс] – Режим доступа: <https://vzh.ru/article/html-ili-cms-cto-luchshe-dlya-kommercheskogo-sajta/> – Заглавие с экрана. – (Дата обращения: 13.09.2022).
- 5) HTML или CMS – что лучше для коммерческого сайта [электронный курс] – Режим доступа: <https://vzh.ru/article/html-ili-cms-cto-luchshe-dlya-kommercheskogo-sajta/> – Заглавие с экрана. – (Дата обращения: 13.09.2022).
- 6) SQLite [электронный курс] – Режим доступа: <https://blog.skillfactory.ru/glossary/sqlite/> – Заглавие с экрана. – (Дата обращения: 13.09.2022).

# Приложения

## Приложение А

```
def delete_customer(self, customer_id):
    try:
        self.__cur.execute("DELETE FROM baskets WHERE customer_id = ?", (customer_id,))
        self.__db.commit()
        self.__cur.execute("SELECT * FROM orders WHERE customer_id = ?", (customer_id,))
        orders_id = self.__cur.fetchall()
        for el in orders_id:
            self.__cur.execute("DELETE FROM order_products WHERE order_id = ?", (el['id'],))
            self.__db.commit()
        self.__cur.execute("DELETE FROM orders WHERE customer_id = ?", (customer_id,))
        self.__db.commit()
        self.__cur.execute("DELETE FROM customers WHERE id = ?", (customer_id,))
        self.__db.commit()
    except sqlite3.Error as e:
        print("При удалении пользователя из БД произошла ошибка")
        print(str(e))
        return False
    return True
```



## Приложение В

```
@app.route('/add_to_basket/<product_id>', methods=['post', 'get'])
@login_required
def add_to_basket(product_id):
    db = get_db()
    dbase = FDataBase(db)

    product = dbase.get_product_by_id(product_id)
    customer_id = current_user.get_id()
    if request.method == 'POST':
        number = request.form['number']
        try:
            number = int(number)
            if number <= 0:
                flash('Введено некорректное количество.', category='fail')
                return render_template('add_to_basket.html', product=product)
        except ValueError:
            flash('Поле заполнено некорректно.', category='fail')
            return render_template('add_to_basket.html', product=product)

    res = dbase.add_product_in_basket(customer_id, product_id, number)
    if res:
        flash('Товар добавлен в вашу корзину!', category='success')
    else:
        flash('Заказываемый товар уже есть в вашей корзине.', category='fail')

    return render_template('add_to_basket.html', product=product)
```