**Руководство программиста**

**1) Назначение и условия применения программы:**

Сайт должен представлять проект «MemoryMerch» в Интернете и служить как онлайн-площадка для продажи продукции.

Программное обеспечение клиентской части должно удовлетворять следующим требованиям:

- Веб-браузер: Internet Explorer 8.0 и выше, или Firefox 3.5 и выше, или Opera 9.5 и выше, или Safari 3.2.1 и выше, или Chrome 2 и выше;

- Операционная система Windows 10 и выше;

- Устройства с разрешением экрана не менее 1280x720 пикселей

- Клавиатура и мышь

**2) Характеристика программы:**

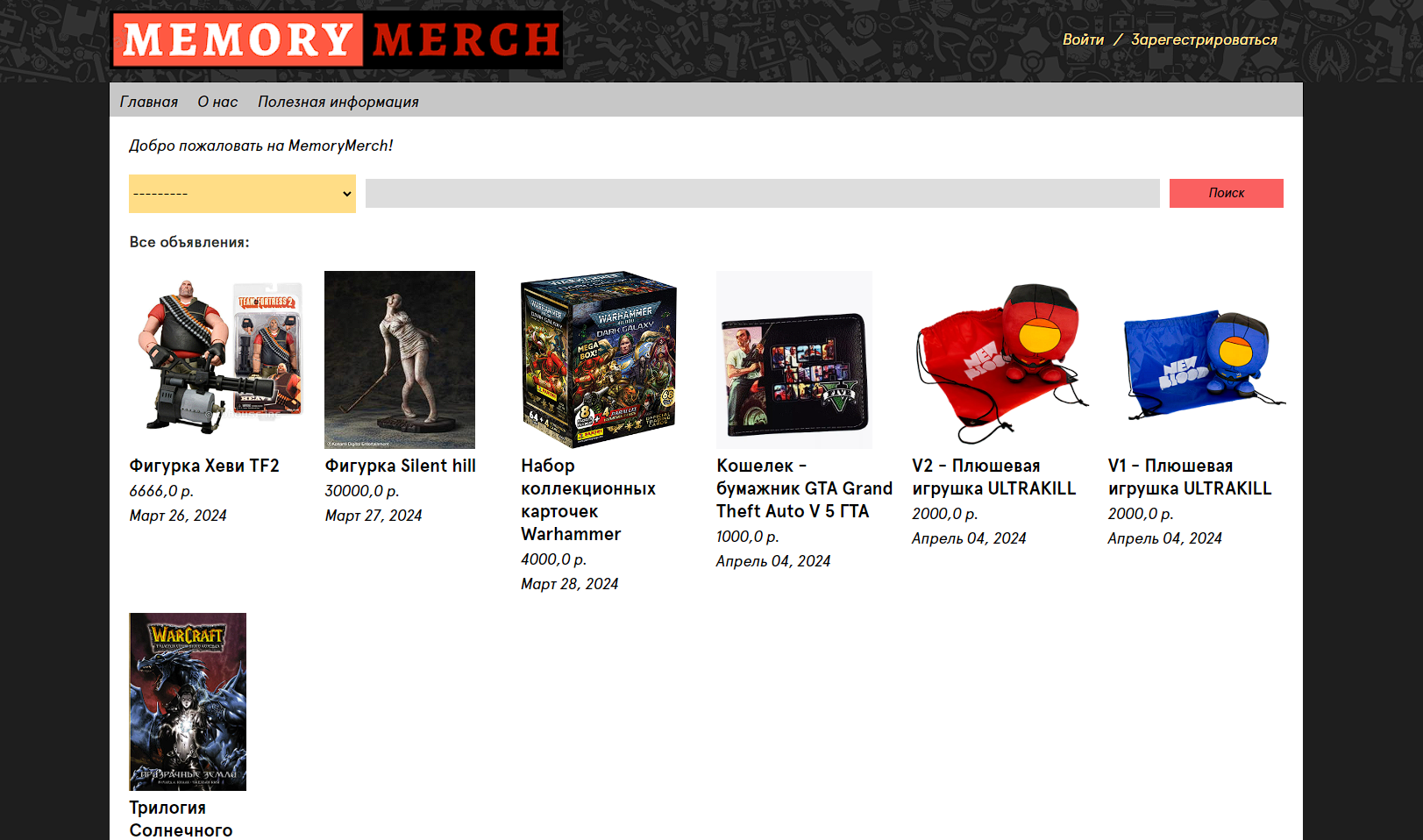
Сайт предназначен для привлечения новых клиентов на стоматологические услуги и предоставление информации об организации.

Сайт должен соответствовать функциональным требованиям:

* Пользователь должен иметь возможность войти в систему, используя свое имя пользователя и пароль;
* Система запоминает зарегистрированного пользователя;
* Пользователь должен иметь возможность переходить по страницам и оставлять добавлять товары в корзину, заказывать на сайте;
* Администратор может добавлять новые товары, состоящие из названия, фото, цены и описания о товаре.
* Администратор может добавлять новые посты, состоящие из заголовка и текста.

**3) Обращение к программе:**

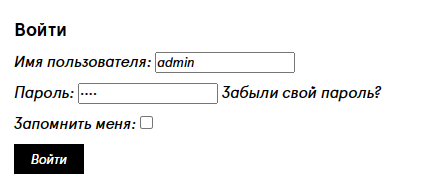
Для открытия сайта необходимо запустить локальный Django-сервер командой *python manage.py runserver*



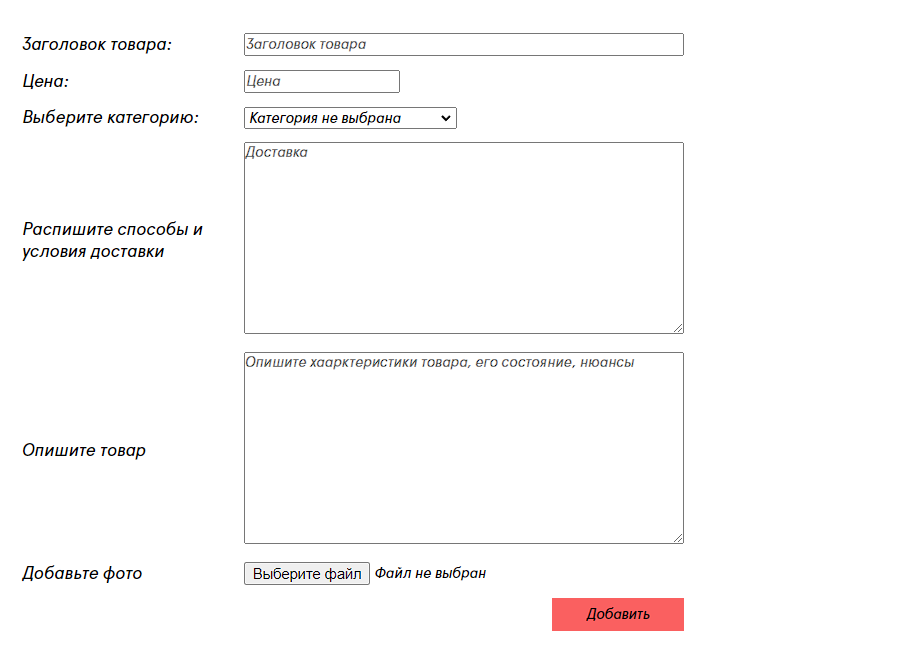
Главная страница

* Добавление Товара в каталог

Товар добавляет только Администратор.



Для добавления необходимо нажать по кнопке с надписью «Добавить товар» в шапке сайта и заполнить следующие поля: Заголовок товара, цена, выбор категории, способы доставки, описание товара, фото товара.



Форма добавления товара

Код модели Товара:

class Tovar(models.Model):

tovarname = models.CharField(max\_length=100, blank=False, default="")

tovarprice = models.FloatField()

category = models.ForeignKey(

"Categories",

related\_name="tovars\_categories",

on\_delete=models.CASCADE,

null=True,

)

shipping = models.TextField(max\_length=300, blank=False, default="")

tovardescrpt = models.TextField(max\_length=500, blank=False, default="")

date\_posted = models.DateTimeField(default=timezone.now, null=True)

tovarimage = models.ImageField(

upload\_to="img/tovarimages",

verbose\_name="Фото товара",

default=None,

blank=False,

)

def \_\_str\_\_(self):

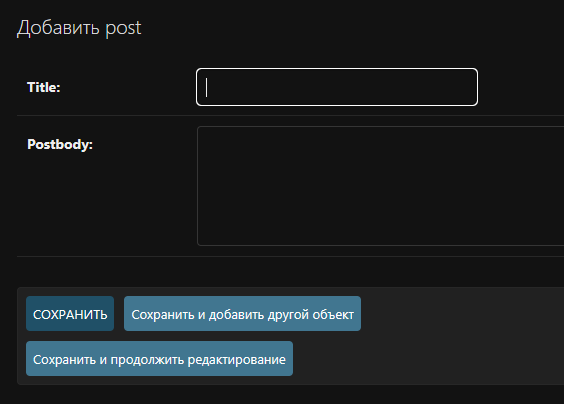
return f"{self.tovarname} {self.category}"

* Работа с админ-панелью

В Django есть встроенная панель администратора, позволяющая работать с базой данных сайта. От аккаунта администратора можно зайти в админ панель по адресу http://127.0.0.1:8000/admin/ . В данной панели админ может добавлять, удалять и редактировать товары, посты.

* Добавление постов

Администратор заполняет поля заголовка и текста поста после сохраняет данные. Пост добавляется на главную страницу.



Форма постов

**4) Входные и выходные данные:**

Таблица 1 Входные данные Товара

|  |  |
| --- | --- |
| **Поле** | **Тип данных** |
| Название товара | CharField |
| Цена | FloatField |
| Категория | ForeignKey |
| Доставка | TextField |
| Описание товара | TextField |
| Дата добавления товара | DateTimeField |
| Фото товара | ImageField |

**5) Выходная информация:**

Выходные данные организованы в виде:

Списка карточек товаров – Главная страница. Название и дата добавления товара на каждой карточке.

Список карточек постов – Главная страница. Заголовок и текст поста.

Строки товаров - Корзина. Название товара и количество в корзине.

Карточка товара – Страница товара. Должна выводиться вся информация о товаре что есть в базе данных.

**6) Сообщения об ошибках:**

Ошибка, возникающая если не установлен какой-то из модулей.

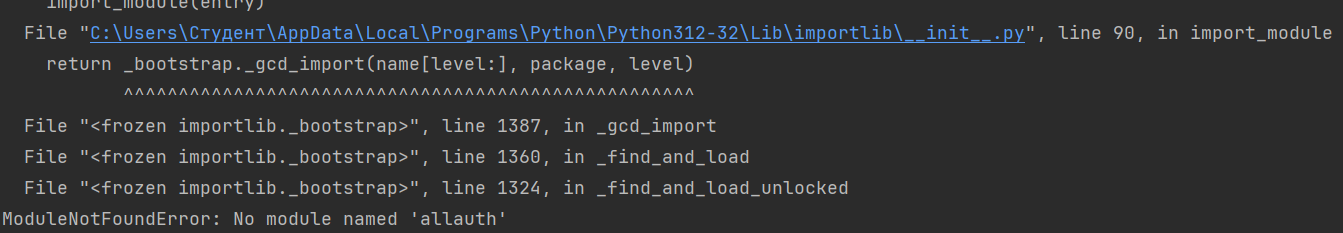


Рисунок 2 Ошибка модуля

Ошибка сайта при попытке добавить товар в корзину не авторизовавшись.

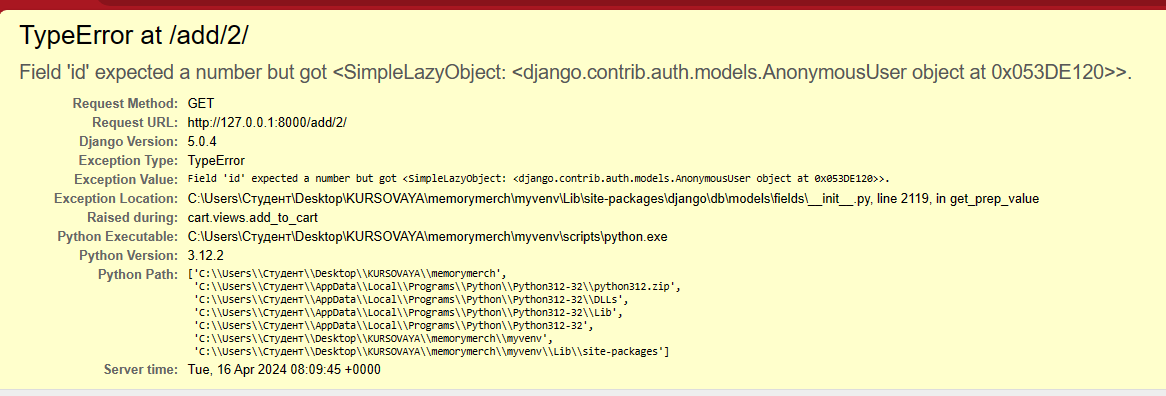


Рисунок 3 Ошибка корзины