Министерство образования Московской области

Государственное образовательное учреждение высшего образования Московской области

«Государственный гуманитарно-технологический университет»

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**КУРСОВОЙ ПРОЕКТ**

Тема: «Разработка Интернет-магазина игровой продукции»

МДК.08.01 «Проектирование и разработка интерфейсов пользователя»

**Выполнила:**

Шипкова Софья Сергеевна

Студентка 3 курса группы ИСП.21.2А

09.02.07 Информационные системы и программирование)

очной формы обучения

**Руководитель:**

Гжегожевский Сергей Владимирович

Оценка

\_\_\_\_\_\_\_\_\_/\_\_\_­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Подпись руководителя

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ликино-Дулево

2024 год

**Введение**

MemoryMerch представляет из себя интернет магазин одноименного проекта MemoryMerch. Сайт предназначен для продажи редкой видеоигровой продукции(«мерча»). Данный сайт будет интересен в первую очередь коллекционерам, геймерам а так же просто фанатам тех или иных медиа. На сайте можно создать свой аккаунт а также приобрести выставленные в каталог товары. Товары распределяются по категориям что упрощает их поиск.

# 1. Разработка системного проекта.

## 1.1. Назначение разработки

Целью разработки является создать веб-сайт для проекта MemoryMerch, с возможностью просмотра каталога товаров, добавление товара в корзину и оформление заказа; а также ознакомления с информацией об проекте, контактами данных.

## 1.2. Требования к программному продукту

### 1.2.1. Требования к функциональным характеристикам

Так как это веб-сайт пользователю потребуется доступ к интернету и любой из интернет-браузеров.

### 1.2.2. Требования к надежности и безопасности

Чтобы добавлять товары в корзину пользователю необходимо зарегистрироваться на сайте.

Для регистрации пользователь вводит придуманное имя, email(не обязательно), пароль и подтверждает пароль. Для безопасности, к созданию пароля применяются следующие условия:

* Пароль не должен быть слишком похож на другую личную информацию.
* Пароль должен содержать как минимум 8 символов.
* Пароль не должен быть слишком простым и распространенным.
* Пароль не может состоять только из цифр.

Попадая в базу данных, пароль пользователя хешируется.

В Django для шифрования паролей используется механизм хеширования. Хеш - это строка уникальных символов, которая получается в результате применения математической функции к исходным данным (в данном случае - к паролю). Хеш-функции обеспечивают одностороннее преобразование данных - из хеша невозможно получить исходные данные.

### 1.2.3. Требования к технологиям и инструментам для вёрстки

Для верстки данного проекта использовалась программа VS Code, так как она является наиболее удобной для работы с HTML и CSS языками.

### 1.2.4. Требования к составу и параметрам технических средств

Описать характеристики клиентских устройств, где может эффективно работать и отображаться ваш сайт. Плюсом будет описание требований к серверной части приложения (размещение на хостинге).

|  |  |
| --- | --- |
| **Технические средства** | **Требования** |
| Браузеры | Последняя версия Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, Opera и др. |
| Устройства | Устройства с разрешением экрана не менее 1280x720 пикселей |
| Устройства ввода | Клавиатура и мышь |
| Операционные системы | Windows 7, 8, 10, macOS 10.12 или новее, iOS 11 или новее, Android 5 или новее и другие ОС |

# 2. Разработка технического проекта

## 2.1. Обоснование выбора CASE – средств

## 2.2. Проектирование модели данных

### 2.2.1. Диаграмма прецедентов

Опишите действующих лиц и процессы, которые возникают между ними с учетом разрабатываемой системы (предмета исследования). Например, пользователь, администратор сайта, менеджер отвечающий за формирование и отправку заказа и др.

Диаграмма вариантов использования или диаграмма прецедентов (use case diagram) - это тип графического представления, который иллюстрирует отношения между актерами и прецедентами. Она помогает в определении требований к системе, анализе поведения системы и выявлении проблем и противоречий в проектировании.



### 2.2.2. Моделирование бизнес-процессов

На основе п.2.2.1 выписать бизнес-процессы

### 2.2.3. Словесный алгоритм бизнес-процессов

Описать 1 из бизнес-процессов, описанных в п.2.2.2.

### 2.2.4. Построение диаграммы действий

Для любого из бизнес-процессов.

# 3. Реализация приложения

## 3.1. Обоснование выбора средств разработки

Для разработки изначально использовалась программа PyCharm однако большая часть позже была написана именно в VS Code. VS Code представляет удобные средства работы с HTML CSS а также с языком Python при установке бесплатного расширения из каталога. Таким образом в одной программе было удобно работать со всеми элементами проекта. Бекендом на Python и фронтендом на HTML и CSS.

## 3.2. Руководство программиста

Структура приложения в обозревателе решений:

### 3.2.1. Построение моделей и структура БД.

Дать описать моделей и составить схему БД.

* Приложение cart (Корзина):

class CartItem(models.Model):

    tovar = models.ForeignKey(Tovar, on\_delete=models.CASCADE)

    quantity = models.PositiveIntegerField(default=0)

    user = models.ForeignKey(User, on\_delete=models.CASCADE)

    date\_added = models.DateTimeField(auto\_now\_add=True)

    def \_\_str\_\_(self):

        return f'{self.quantity} x {self.tovar.tovarname}'

* Приложение main:

class Categories(models.Model):

    catgname = models.CharField(max\_length=60, blank=False, default="")

    def \_\_str\_\_(self):

        return self.catgname

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}"

class Post(models.Model):

    title = models.CharField(max\_length=100, blank=False)

    postbody = models.TextField(max\_length=1000)

    def \_\_str\_\_(self):

        return self.title

### 3.2.2. Представления

* Представление добавления товара:

def createtovar(request):

    error = ""

    if request.method == "POST":

        form = TovarForm(request.POST, request.FILES)

        if form.is\_valid():

            form.save()

            return redirect("index")

        else:

            error = "Форма заполнена неправильно или не до конца"

    form = TovarForm()

    context = {"form": form}

    return render(request, "main/create-tovar.html", context)

* Представление Главной страницы

def index(request):

    tovars\_all = (Tovar.objects.all(),)

    posts = Post.objects.all()

    filter\_tovar = TovarFilter(request.GET, queryset=Tovar.objects.all())

    return render(request, "main/index.html", {"tovars": filter\_tovar, "posts": posts})

### 3.2.3. Шаблоны

Описать какие шаблоны есть. Можно рассказать про структуру шаблона на примере одного из.

Для упрощения верстки я использовала возможности встроенного шаблонизатора jinja. Был создан «базовый» файл-шаблон верстки, включающий в себя хедер и футер сайта. В данный шаблон я далее подключала различное наполнение и контент с помощью «{% block название блока %}{% endblock %}».

Код базового шаблона сайта можно посмотреть в приложении.

### 3.2.4. Формы

Код форм описан в файле forms.py. Формы создаются с помощью класса ModelForm

Форма добавления товара:

class TovarForm(ModelForm):

    def \_\_init\_\_(self,\*args,\*\*kwargs):

        super().\_\_init\_\_(\*args,\*\*kwargs)

        self.fields['category'].empty\_label = 'Категория не выбрана'

    category = forms.ModelChoiceField(queryset=Categories.objects.all(), empty\_label="Категория", required=True)

    class Meta:

        model = Tovar

        fields = ["tovarname", 'tovarprice', 'category', 'shipping', 'tovardescrpt','tovarimage']

        widgets = {

            "tovarname": TextInput(attrs={'placeholder': 'Заголовок товара', 'class': 'form-input'}),

            "tovarprice": TextInput(attrs={'placeholder': 'Цена', 'class': 'form-input'}),

            "category": forms.Select(attrs={'placeholder': 'Категория', 'class': 'form-input'}),

            "shipping": Textarea(attrs={'placeholder': 'Доставка', 'class': 'form-input'}),

            "tovardescrpt": Textarea(attrs={'placeholder': 'Фото товара', 'class': 'form-input'}),

        }

Формы аутентификации создавалась автоматически библиотекой allauth.

### 3.2.5. Можно добавить любой свой пункт

## 3.3. Руководство пользователя

Можно добавить подпункты, можно не добавлять.

# 4. Тестирование программного продукта

## 4.1. Тестовые сценарии

См. картинку Тестовые-сценарии ниже. Создать не менее 3 тестовых сценариев.

## 4.2. Юнит-тесты в Django

Описать созданные тесты

# 5. Размещение проекта на хостинге.

Заключение

Список литературы

https://icons8.com/icons

Приложения

Код шаблона base.html

{% load static %}

<!DOCTYPE html>

<html lang="ru">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="{% static 'css/styles.css'%}" type="text/css">

    <title>Memorymerch | {% block title%}{% endblock %}</title>

</head>

<body>

    <img src="{% static 'img/kitten.gif' %}" alt="" id="kitten">

    <header>

        <div class="header-wrapper">

            <div class="logo"><a href="{% url 'index' %}"><img src="{% static 'img/logo.png' %}" alt="logo"></a></div>

            {% if user.is\_authenticated %}

            <form action="{% url 'create-tovar' %}" target="\_blank">

                <button id="createOB">Добавить товар</button>

            </form>

            {% else %}

            {% endif %}

            <div id="userlog">

                {% if user.is\_authenticated %}

                {{ user.username }}

                <a class="main-header\_links" href="{% url 'cart:view\_cart' %}" id="avatar"><img

                        src="{% static 'img/cart.png' %}" alt="avatar"></a> |

                <a class="main-header\_links" href="{% url 'account\_logout' %}">Выйти</a>

                {% else %}

                <a class="main-header\_links" href="{% url 'account\_login' %}">Войти</a> / <a

                    href="{% url 'account\_signup' %}" class="main-header\_links">Зарегестрироваться</a>

                {% endif %}

            </div>

        </div>

    </header>

    <div class="wrapper">

        <!-- навигация -->

        <div class="main-nav">

            <li><a href="{% url 'index' %}">Главная</a></li>

            <li><a href="{% url 'about' %}">О нас</a></li>

            <li><a href="{% url 'rules' %}">Полезная информация</a></li>

        </div>

        <!-- контент -->

        <div class="container">

            <div class="content">

                {% if messages %}

                {% for message in messages %}

                <div class="alert alert-{{ message.tags }}">

                    {{ message }}

                </div>

                {% endfor %}

                {% endif %}

                {% block content %}{% endblock %}

            </div>

        </div>

    </div>

    <footer>

        <div class="footer-wrapper">

            <div class="footer-grid">

                <div class="footer\_\_c">

                    <p>(с) MEMORYMERCH 2024 <br>

                        Все права защищены</p>

                </div>

                <div class="footer\_\_socials">

                    <a href=""><img src="{% static 'img/socials\_vk.png' %}" alt="social-1"></a>

                    <a href=""><img src="{% static 'img/socials\_tg.png' %}" alt="social-2"></a>

                    <a href=""><img src="{% static 'img/socials\_ok.png' %}" alt="social-3"></a>

                </div>

                <div class="footer\_contacts">

                    <p><span id="footer\_\_phone">+ 7 999 999 99 99</span> - 11:00 - 18:00 ПН-ПТ</p>

                    <p id="footer\_\_email">shop.memorymerch.support@mm.ru</p>

                </div>

            </div>

        </div>

    </footer>

</body>

</html>

Тестовые-сценарии

