Дипломная работа

Реализация проекта

“Космический магазин - блог”

на django

с интеграцией Telegram-бота

Автор:

Галышева Екатерина Олеговна

Python-разработчик 90-й поток | 23.12.2024

Оглавление

1. Введение. 6

Обоснование выбора темы.

Определение цели и задач исследования.

2. Основные понятия и определения. 8

3. Методы и подходы к разработке. 9

Архитектура веб-приложения.

Обеспечение безопасности.

4. Обзор инструментов для разработки проекта 10

Django.

Aiogram.

5. Проектирование приложения. 12

Планирование и анализ требований.

Основные требования.

Технические требования.

6. Разработка в соответствии с созданной документацией. 13

Планирование разработки.

Разработка.

7. Анализ и интерпретация результатов. 13

Сравнение моделей.

Интерпретация результатов.

Рекомендации по выбору модели в зависимости от проекта.

8. Заключение. 14

Обзор выполненной работы.

Дальнейшие планы.

Введение

Основание выбора темы:

**Актуальность:** в условиях растущей конкуренции в онлайн-торговле важно не только продавать товары, но и создавать ценность для клиентов через контент. Блог позволяет делиться знаниями, опытом и рекомендациями, что способствует формированию доверия к бренду.

**Технологические возможности**: Django — это мощный фреймворк для разработки веб-приложений, который предоставляет множество инструментов для быстрого создания и масштабирования проектов. Его возможности по работе с базами данных, аутентификацией пользователей и администрированием делают его идеальным выбором для реализации проекта.

**Интеграция с Telegram**: Telegram-боты становятся всё более популярными среди пользователей, так как они позволяют получать информацию в удобном формате и в реальном времени. Интеграция Telegram-бота с магазином и блогом позволит пользователям получать уведомления о новых товарах, акциях и обновлениях блога, а также взаимодействовать с магазином напрямую через мессенджер.

**Увеличение вовлеченности пользователей**: Комбинирование магазина и блога в одном проекте способствует увеличению времени, проведенного пользователями на сайте, а также повышает вероятность совершения покупок. Блог может служить источником информации о товарах, их использовании и преимуществах, что способствует принятию решения о покупке.

**Перспективы развития**: Проект "Магазин-Блог" можно легко масштабировать, добавляя новые функции, такие как система рекомендаций, пользовательские отзывы, интеграция с социальными сетями и многое другое. Это открывает широкие горизонты для дальнейшего развития и улучшения пользовательского опыта.

Определение цели и задач исследования

Основной целью проекта является создание функционального и удобного веб-приложения, которое объединяет возможности интернет-магазина и блога. Для достижения этой цели необходимо решить следующие задачи:

* Разработка структуры базы данных для хранения информации о товарах, статьях блога и пользователях.
* Реализация интерфейса для управления товарами и контентом блога с помощью административной панели Django.
* Создание пользовательского интерфейса для отображения товаров и статей, а также для взаимодействия с пользователями.
* Интеграция Telegram-бота, который будет уведомлять пользователей о новинках и акциях, а также предоставлять доступ к контенту блога.
* Проведение тестирования и оптимизация приложения для обеспечения высокой производительности и удобства использования.
* Написание дипломной работы: составить дипломную работу, включающую в себя введение, обзор литературы, методологию и результаты исследования, анализ результатов, выводы и рекомендации.

2. Основные понятия и определения

Обзор основных понятий

**Фреймворк** (Framework): Программная платформа, которая предоставляет готовые компоненты и инструменты для разработки приложений. В контексте веб-приложений часто используются Django, FastAPI или Flask.

**Веб-приложение** (Web Application): Программное приложение, которое работает на веб-сервере и доступно через браузер.

**Бэкенд** — это серверная часть приложения, которая отвечает за обработку данных, взаимодействие с базой данных и бизнес-логику.

**Фронтенд** — это клиентская часть приложения, которая отвечает за взаимодействие с пользователем.

**Django** — это высокоуровневый веб-фреймворк на языке Python, который позволяет быстро разрабатывать веб-приложения. Он следует принципу "не повторяйся" (DRY) и включает множество встроенных функций, таких как аутентификация пользователей, работа с базами данных и администрирование.

**Telegram-бот** — это автоматизированный аккаунт в мессенджере Telegram, который может взаимодействовать с пользователями, отправлять уведомления, обрабатывать команды и выполнять различные задачи. Боты могут быть использованы для улучшения пользовательского опыта и взаимодействия с клиентами.

**Интернет-магазин** — это веб-платформа, на которой пользователи могут просматривать товары, добавлять их в корзину и оформлять заказы. Он предоставляет функционал для управления каталогом товаров, обработки платежей и управления заказами.

**Блог** — это веб-сайт, на котором публикуются статьи, записи или посты на определённые темы. Блоги могут содержать текст, изображения, видео и другие медиаформаты, и служат для информирования, обучения и взаимодействия с аудиторией.

**База данных** — это организованная структура для хранения, управления и извлечения данных. В контексте проекта "Магазин-Блог" база данных будет использоваться для хранения информации о пользователях, товарах, статьях блога и заказах.

**Административная панель** — это интерфейс, предоставляемый Django, который позволяет администраторам управлять содержимым веб-приложения, включая добавление, редактирование и удаление товаров и статей блога.

**Пользовательский интерфейс** — это часть веб-приложения, с которой взаимодействуют пользователи. Он включает в себя элементы дизайна, такие как кнопки, меню, формы и другие компоненты, которые обеспечивают удобство использования.

**Оптимизация** — это процесс улучшения производительности веб-приложения, который может включать в себя ускорение загрузки страниц, уменьшение времени отклика сервера и улучшение пользовательского опыта.

**Вовлеченность пользователя** — это степень активности и заинтересованности пользователей в контенте и функционале веб-приложения. Высокая вовлеченность может привести к увеличению времени, проведенного на сайте, и повышению вероятности совершения покупок.

**Контентный маркетинг** — это стратегия, направленная на создание и распространение ценного контента для привлечения и удержания целевой аудитории. В контексте проекта "Магазин-Блог" блог будет служить инструментом контентного маркетинга, позволяя делиться знаниями и опытом с пользователями.

Эти понятия и определения помогут лучше понять ключевые аспекты проекта и его функциональность.

Методы и подходы к разработке

**Выбор фреймворка** для разработки проекта зависит от нескольких факторов, включая требования проекта, опыт команды и предпочтения в технологиях. В данном проекте используется Django.

Django — это высокоуровневый фреймворк на Python, который следует принципу "бatteries-included". Он предлагает множество встроенных функций, таких как система аутентификации, админ-панель, ORM (Object-Relational Mapping) и поддержка работы с формами.

Преимущества:

**б**ыстрая разработка благодаря множеству встроенных функций.

Хорошая документация и поддержка сообщества.

Подходит для создания сложных приложений с высокой степенью безопасности.

**Интеграция Django в проект** включает в себя установку необходимых библиотек, создание проекта и приложений, настройку базы данных, создание моделей для таблиц, создание панели администрирования, настройку URL-адресов, создание представлений и шаблонов.

**Архитектура веб-приложения.** Веб-приложение обычно состоит из двух основных компонентов: фронтенд и бэкенд.

Бэкенд. Django принимает HTTP-запросы от клиента и обрабатывает их. Используя ORM (Object-Relational Mapping), Django взаимодействует с базой данных для сохранения и извлечения данных. Django предоставляет встроенные механизмы для работы с пользователями, их регистрацией и управлением доступом. **С** помощью Django REST Framework можно создать RESTful API для взаимодействия с фронтендом или мобильными приложениями. Бэкенд реализует бизнес-логику, например, обработку заказов в магазине или управление постами в блоге.

Фронтенд включает в себя HTML структуру веб-страниц, CSS стилизацию и оформление веб-страниц, JavaScript логику взаимодействия и динамическое обновление контента на страницах.

Взаимодействие фронтенда и бэкенда:

HTTP-запросы: Фронтенд отправляет запросы к бэкенду через REST API или через стандартные формы.

Получение данных: Бэкенд обрабатывает запросы и возвращает данные, которые фронтенд отображает пользователю.

Для связи между фронтендом и бэкендом можно использовать REST API. Это позволит фронтенду отправлять изображения или другие данные на сервер для обработки моделью и получать результаты.

**Обеспечение безопасности.**

Для аутентификации и авторизации пользователя используется JWT.

Для отслеживания действий пользователей и выявления подозрительных действий можно использовать логирование и мониторинг с помощью Sentry или New Relic.

4. Обзор инструментов для разработки проекта

Разработка веб-приложений на Python, особенно с использованием Django и библиотеки aiogram для создания ботов Telegram, требует использования ряда популярных инструментов и библиотек. Вот обзор некоторых из них:

1. Django

Описание: Django — это высокоуровневый веб-фреймворк на Python, который способствует быстрой разработке и чистому, практичному дизайну.

Основные функции:

Встроенная система аутентификации и авторизации.

ORM (Object-Relational Mapping) для работы с базами данных.

Поддержка REST API через Django REST Framework.

Мощные средства для работы с формами и валидацией данных.

Шаблонизатор для генерации HTML-страниц.

2. Django REST Framework (DRF)

Описание: DRF — это мощный и гибкий инструмент для создания веб-API на основе Django.

Основные функции:

Поддержка аутентификации (JWT, OAuth2 и др.).

Легкая работа с сериализацией данных.

Встроенные классы представлений и маршрутизации для API.

Поддержка пагинации, фильтрации и поиска.

3. aiogram

Описание: aiogram — это асинхронная библиотека для разработки ботов Telegram на Python, которая использует asyncio.

Основные функции:

Простота в использовании и высокая производительность.

Поддержка всех функций Telegram Bot API.

Возможность работы с вебхуками и долгими опросами.

Удобные механизмы для обработки сообщений и состояний.

4. PostgreSQL

Описание: PostgreSQL — это мощная объектно-реляционная система управления базами данных.

Основные функции:

Поддержка сложных запросов и транзакций.

Высокая производительность и надежность.

Поддержка расширений, таких как PostGIS для работы с геоданными.

5. Docker

Описание: Docker — это платформа для автоматизации развертывания приложений в контейнерах.

Основные функции:

Упрощение развертывания и управления зависимостями.

Создание изолированных сред для разработки и тестирования.

Легкость в масштабировании приложений.

6. Celery

Описание: Celery — это асинхронный распределенный таск-менеджер для выполнения задач в фоновом режиме.

Основные функции:

Поддержка планирования задач.

Возможность обработки долгих операций, таких как отправка электронной почты или выполнение сложных вычислений.

Интеграция с Django через django-celery.

7. Redis

Описание: Redis — это хранилище данных в памяти, которое часто используется в сочетании с Celery для управления очередями задач.

Основные функции:

Высокая скорость обработки данных.

Поддержка различных структур данных (строки, списки, множества и т.д.).

Используется для кэширования и хранения сессий.

8. pytest

Описание: pytest — это мощный инструмент для тестирования на Python.

Основные функции:

Поддержка простого и гибкого написания тестов.

Возможность использования фикстур для подготовки тестовых данных.

Поддержка тестирования как для Django, так и для aiogram.

9. Swagger / OpenAPI

Описание: Инструменты для документирования API, такие как Swagger или OpenAPI, помогают создавать интерактивные документации для вашего API.

Основные функции:

Автоматическая генерация документации на основе кода.

Возможность тестирования API прямо из документации.

Упрощение работы с API для разработчиков и пользователей.

Использование этих инструментов в сочетании с Django и aiogram позволит эффективно разрабатывать веб-приложения и ботов Telegram. Каждый из инструментов имеет свои уникальные функции и возможности, которые помогают оптимизировать процесс разработки и улучшить качество конечного продукта.

5. Проектирование приложения

ПЛАНИРОВАНИЕ И АНАЛИЗ ТРЕБОВАНИЙ

Выбор фреймворка и инструментов:

Фреймворк: используем Django как основной фреймворк для разработки бэкенда приложения. Django предлагает мощные инструменты для работы с базами данных, аутентификацией и маршрутизацией, что делает его идеальным выбором для проектов с пользовательскими данными и API.

Фронтенд: для разработки пользовательского интерфейса используем HTML, CSS и JavaScript. Bootstrap поможет в создании адаптивного и современного дизайна.

Бэкенд: Django для обработки бизнес-логики, управления пользователями и API.

База данных: PostgreSQL для хранения данных о пользователях, загруженных изображениях и результатах распознавания.

Интеграция с Telegram-ботом: используем библиотеку python-telegram-bot для создания и интеграции Telegram-бота, который будет взаимодействовать с пользователями, предоставляя информацию о товарах и новостях блога.

РАЗРАБОТКА ПРОТОТИПА

Создание базовой версии приложения, которая включает:

Страницу магазина: отображение списка товаров с возможностью фильтрации и поиска. Каждая карточка товара будет содержать изображение, название, цену и кнопку для добавления в корзину.

Страницу блога: отображение списка статей, с возможностью сортировки по категориям и тегам. Каждая статья будет содержать заголовок, краткое описание и ссылку на полную версию.

Функционал корзины: пользователи смогут добавлять товары в корзину, просматривать ее содержимое и оформлять заказы.

Регистрация и аутентификация пользователей: реализация системы регистрации и входа для пользователей с возможностью восстановления пароля.

РАЗРАБОТКА КЛЮЧЕВОГО ФУНКЦИОНАЛА

Магазин:

Разработка моделей для товаров, категорий и заказов.

Реализация CRUD (создание, чтение, обновление, удаление) операций для управления товарами через панель администратора.

Интеграция платежной системы для обработки заказов (например, через Stripe или PayPal).

Блог:

Создание моделей для статей, категорий и комментариев.

Реализация функционала для добавления, редактирования и удаления статей и комментариев.

Возможность комментирования статей зарегистрированными пользователями.

Telegram-бот:

Разработка бота, который будет предоставлять пользователям информацию о новых товарах и статьях блога.

Реализация команд для получения информации о текущих акциях, статусе заказов и новых записях в блоге.

Настройка уведомлений о новых товарах и статьях, которые будут отправляться подписчикам.

ИНТЕГРАЦИЯ, ПОДКЛЮЧЕНИЕ И НАСТРОЙКА

Интеграция с Telegram-ботом:

Настройка веб хуков для получения обновлений от Telegram.

Реализация логики обработки команд и сообщений от пользователей.

Настройка базы данных для хранения информации о подписчиках и их предпочтениях.

Обработка изображений и медиа файлов:

Использование библиотеки Pillow для обработки изображений товаров и статей блога.

Проектирование приложения включает в себя создание функционального и удобного интерфейса для пользователей, а также интеграцию с Telegram-ботом для улучшения взаимодействия с клиентами. Следуя данному плану, можно создать качественное и эффективное приложение, которое будет сочетать в себе возможности онлайн-магазина и блога.

.

6. Разработка в соответствии с созданной документацией

Планирование разработки

Разработка была разделена на несколько основных этапов: проектирование интерфейса, реализация серверной логики и интеграция модели машинного обучения. Планирование задач и отслеживание прогресса осуществлялось в программе Obsidian.  
  
Разработка  
  
Фронтенд-разработка:  
  
Создан пользовательский интерфейс с использованием Jinja2 для шаблонов и Bootstrap для стилизации.  
  
Разработаны формы загрузки изображений и страницы отображения результатов распознавания. Создана страница авторизации.

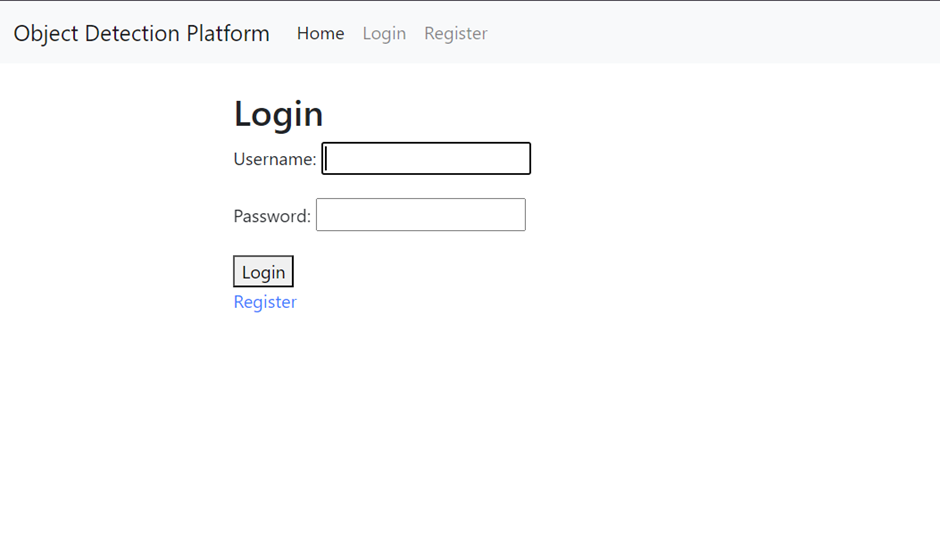


Рис. 1 Страница авторизации

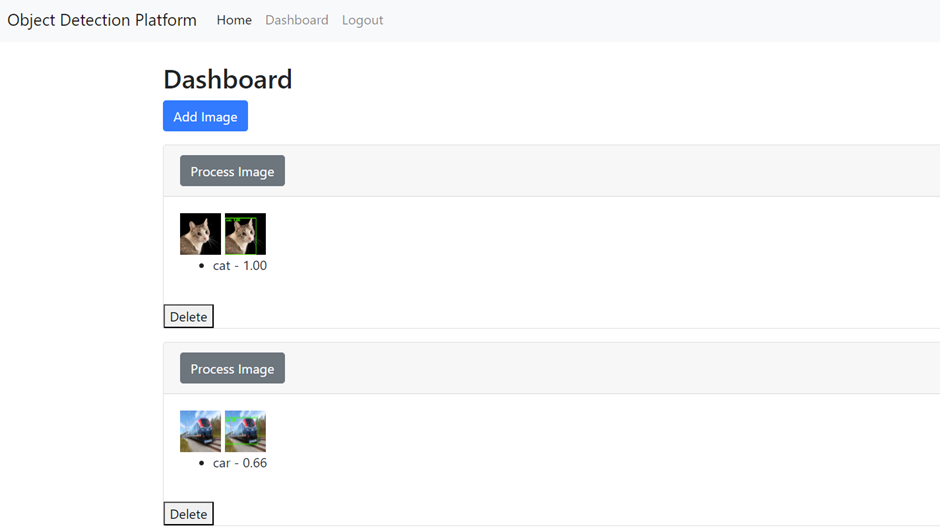


Рис. 2 Изображение и результаты обработки

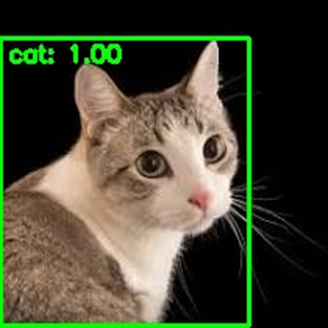


Рис. 3 Пример обработанного изображения крупным планом

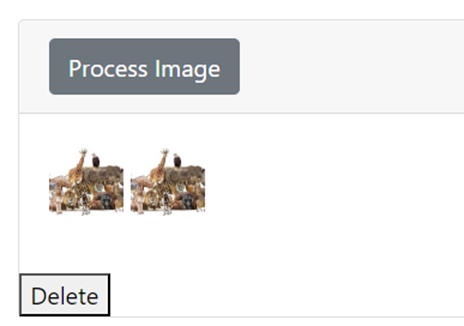


Рис. 4 Пример перегруженного объектами изображения, на нём VGG ничего не смогло обнаружить  
  
Бэкенд-разработка:  
  
Реализована серверная логика с использованием фреймворка Django.  
  
Настроены маршруты для обработки загрузок изображений, взаимодействия с моделью машинного обучения и передача результатов на фронтенд.  
  
Интеграция модели машинного обучения:  
  
Подключена и настроена предобученная модель VGG для выполнения распознавания объектов.  
  
Реализована обработка результатов модели (обнаруженные объекты обводятся рамкой) и их отображение на веб-страницах.

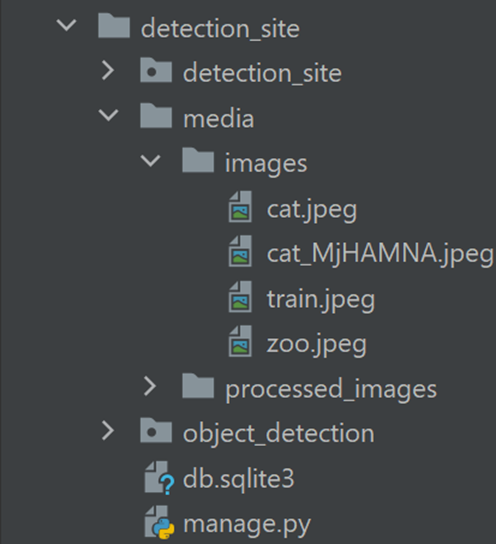


Рис. 5 Файловая структура проекта

7. Анализ и интерпретация результатов

Нужно написать что именно сделано!!!!

**Сравнение моделей**  
  
Были протестированы различные предобученные модели, включая VGG, на различных наборах данных.  
  
Основные метрики, такие как достоверность, полнота, точность (precision) и время отклика, были использованы для оценки производительности каждой модели.  
  
**Интерпретация результатов**  
  
Модель VGG показала высокую точность распознавания объектов, но время отклика было выше по сравнению с более легкими моделями.  
  
Другие модели, такие как YOLO и ResNet, также продемонстрировали хорошие результаты, с различиями в скорости и точности.  
  
**Рекомендации по выбору модели в зависимости от проекта**  
  
Для задач, требующих высокой точности и допускающих большее время отклика, рекомендуется использовать модель VGG.  
  
Для задач, где важна скорость обработки, можно рассмотреть использование моделей YOLO или других легковесных моделей.

После завершения разработки проекта важно провести анализ и интерпретацию полученных результатов. Это поможет понять, насколько успешно приложение решает поставленные задачи и удовлетворяет потребности пользователей.

Оценка функциональности

**Магазин:**

Пользовательский интерфейс: оценка удобства навигации по страницам магазина, легкость поиска и фильтрации товаров.

Корзина: проверка функциональности добавления, удаления товаров и оформления заказов. Анализ процесса оформления заказа на предмет его простоты и интуитивности.

Платежная система: оценка интеграции с платежной системой, включая скорость обработки транзакций и безопасность данных пользователей.

**Блог:**

Качество контента: анализ статей на предмет их актуальности и интересности для целевой аудитории. Оценка вовлеченности пользователей через комментарии и взаимодействие с контентом.

Удобство навигации: проверка легкости поиска статей по категориям и тегам, а также возможность сортировки по дате или популярности.

**Telegram-бот:**

Удобство использования: оценка интерфейса бота, простота команд и доступность информации. Анализ частоты использования бота и его популярности среди пользователей.

Обратная связь: сбор отзывов пользователей о функциональности бота и его полезности для получения информации о товарах и новостях блога.

Производительность приложения

Скорость загрузки: измерение времени загрузки страниц приложения, включая страницы магазина и блога. Определение параметров, которые могут замедлять работу (например, оптимизация изображений и кода).

Нагрузочное тестирование: проведение тестов на нагрузку для определения, как приложение справляется с большим количеством пользователей одновременно. Оценка стабильности работы при увеличении трафика.

Результаты и выводы

Достижение целей: оценка, насколько приложение соответствует изначально поставленным целям и задачам. Выявление успешных аспектов и тех, которые требуют доработки.

Потенциальные улучшения: на основе собранных данных и отзывов пользователей формулирование рекомендаций по улучшению функциональности и пользовательского опыта. Это может включать добавление новых функций, улучшение дизайна или оптимизацию производительности.

Анализ и интерпретация результатов разработки проекта позволят не только оценить его текущее состояние, но и определить направления для дальнейшего развития. Важно продолжать собирать отзывы и данные о пользователях, чтобы адаптировать приложение под их потребности и улучшать взаимодействие с ним.

8. Заключение

Проект "Магазин-Блог" на Django с интеграцией Telegram-бота представляет собой актуальное и многообещающее решение для бизнеса, стремящегося увеличить свою онлайн-присутствие и улучшить взаимодействие с клиентами. Реализация данного проекта позволит не только создать удобную платформу для покупок, но и стать источником полезной информации для пользователей, что в свою очередь будет способствовать росту продаж и укреплению бренда.

****Обзор выполненной работы****

Проектирование и разработка веб-приложения на платформе Django для отображения магазина и блога на космическую тематику с интеграцией телеграмм бота успешно завершены в соответствии с изначально созданной документацией.

Приложение включает функционал небольшого магазина, подключенного к базе данных SQLite, блога с постами по тематике товаров из магазина, а также имеет возможность коммуникации с посетителем через бот.

Реализованное приложение соответствует требованиям и демонстрирует высокую производительностью.

Приложение прошло тестирование и готово к реализации.

****Дальнейшие планы****

Планируется расширение интерфейса, добавление нового функционала, таких как:

оформление заказов из корзины;

обработка платежных систем;

создание системы доставки;

подключение социальных сетей;

доработка отправки постов по email;

сохранение истории загрузок и результатов;

улучшение интерфейса и расширение функциональности.