Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Южно-Уральский государственный университет» (НИУ)

Высшая школа экономики и управления

Кафедра «Информационные технологии в экономике»

Разработка интернет-магазина на фреймворке Django

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА ККУРСОВОЙ

по дисциплине «Технологии разработки Web систем»

Руководитель работы, старший преподаватель кафедры

\_\_\_\_\_\_\_\_\_\_\_ (Костерин В.В)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Автор работы

Студент группы ЭУ-342

Александровский А.П.

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Работа защищена

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

с оценкой \_\_\_\_\_\_\_\_\_\_\_

Челябинск 2021

**АННОТАЦИЯ**

Александровский А.П. Разработка интернет-магазина

на фреймворке Django

– Челябинск: ЮУрГУ,

ЭУ-342,2020. 24с., 11 рисунков,

Библиографический

список – 1 наим.

Данная работа посвящена разработке интернет-магазина техники на фреймворке Django. Данный магазин включает в себя главную страницу с основной информацией, витриной товаров, страницей товаров с возможностью добавления в корзину, сама корзина с возможностью удаления оттуда товаров, страница оформления заказа, регистрация и авторизация пользователей.

**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ** 4](#_Toc59621686)

[**ГЛАВА 1 ЛУЧШИЕ ФРЕЙМВОРКИ PYTHON ДЛЯ ВЕБ-РАЗРАБОТКИ** 5](#_Toc59621687)

[**ГЛАВА 2 РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА С ПОМОЩЬЮ DJANGO** 13](#_Toc59621688)

[**ВЫВОД** 23](#_Toc59621689)

[**БИБЛИОГРАФИЧЕСКИЙ СПИСОК** 24](#_Toc59621690)

# **ВВЕДЕНИЕ**

В рамках курсовой работы необходимо было создать интернет – магазин, с использованием фреймворков. Для начала, необходимо разобраться что же такое ФРЕЙМВОРК. Фреймворк - программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Иногда это просто необходимо, ведь иногда для реализации того или иного проекта нужно потратить очень большое количество времени, но Фреймворки гораздо упрощают нам жизнь.

Существует множество различных фреймворков, они подразделяются на различные категории и каждый из них предназначен для решения определенной задачи. Хотелось бы отметить самые важные по моему мнению для программиста на сегодняшний день.

# **ГЛАВА 1 ЛУЧШИЕ ФРЕЙМВОРКИ PYTHON ДЛЯ ВЕБ-РАЗРАБОТКИ**

Python — один из самых популярных и простых для изучения языков программирования и применяется почти везде, в том числе в веб-разработке. Для него есть много фреймворков, часть которых не просто облегчает разработку, но и предоставляет инструменты, позволяющие буквально за пару дней поднять готовый сайт.

Такие фреймворки относятся к категории full stack. Они крутые, в них куча инструментов и всё включено, но это может сделать их тяжёлыми, медленными и негибкими. Кроме фулстек-фреймворков есть и другие, например микро- или асинхронные фреймворки.

Главный плюс фулстек-фреймворков в том, что всё, нужное для полноценного

приложения, в них уже есть. Не нужно искать отдельные библиотеки для каждой мелкой задачи и думать о совместимости, поэтому даже начинающие смогут быстро собрать готовое работающее приложение. Немного расскажем о фреймворках

**1. Django**

**2. Pyramid**

**3. Flask**

**4. Tornado**

**5. CherryPy**

**1 Django**

****

Высокоуровневый фреймворк, предназначенный для быстрой и эффективной разработки. Его архитектура выстроена так, чтобы максимально экономить ваше время и силы, а в будущем — время и деньги заказчиков. Основные принципы философии Django — DRY и rapid development — стимулируют пере-использование кода и уменьшают избыточность.

Возможности:

* Контроль версий для баз данных (миграции).
* Собственный движок шаблонов.
* Объектно-реляционные отображения ([ORM](https://ru.wikipedia.org/wiki/ORM)).
* Маршрутизация URL.
* Поддержка веб-серверов.
* Поддержка аутентификации.
* Поддержка интернационализации.

**Плюсы**

* **Масса библиотек**. Базовую функциональность не нужно писать самому — многое уже написано, остаётся только импортировать соответствующие библиотеки. На качество библиотек Django обычно можно положиться.
* **Сообщество и документация**. У Django подробная документация и дружелюбное сообщество, поэтому всегда можно разобраться, что как работает, или спросить у специалистов.
* **Масштабируемость**. Если вы понятия не имеете, насколько сильно ваш проект вырастет и вырастет ли вообще (как это обычно и бывает со стартапами), Django позволяет начать с малого и масштабироваться по мере необходимости.

**Минусы**

* Django из коробки не поддерживает WebSockets, поэтому он плохо подходит для работы в реальном времени.
* Готовые библиотеки — это в целом хорошо, но часто они снижают гибкость.

**Где используется?**

* [Instagram](https://www.instagram.com/) — социальная сеть для обмена фотографиями и видео.
* [Spotify](https://www.spotify.com/) — музыкальная платформа, рекомендующая пользователям музыку в зависимости от их предпочтений.

**2 Pyramid**

****

Если хочется что-то более минималистичное, то вам может прийтись по вкусу Pyramid. Этот фреймворк подходит для проектов любого размера. В нём есть полезные фичи для создания сложных приложений или масштабирования изначально небольших сайтов под возросшую нагрузку.

**Возможности:**

* Удобные инструменты для работы со [статичными ассетами](https://docs.pylonsproject.org/projects/pyramid/en/latest/narr/assets.html#static-assets-section).
* Предикаты и рендереры.
* Генерация URL.

**Плюсы**

* **Гибкость и удобство кастомизации**. Любой компонент фреймворка, будь то база данных или движок шаблонов, может быть заменён. Можно даже использовать несколько разных компонентов одновременно (например, подключить две разные базы данных).
* **Удобные Ajax-запросы**. Благодаря системе декораторов и представлений можно отправлять XHR-запросы без дополнительных усилий со стороны разработчика.
* **Поддержка SQLAlchemy**. SQLAlchemy обеспечивает удобную работу с базами данных даже для сложных запросов.

**Минусы**

* Требует времени на развёртывание и подготовку к разработке.
* Чтобы заниматься кастомизацией, надо хорошо разбираться в Pyramid.
* Для простых приложений SQLAlchemy может быть чересчур громоздкой.

**Где используется?**

* [Charte.ca](https://www.charte.ca/) — онлайновый редактор графиков для неспециалистов.
* [Easy Blog Networks](https://www.easyblognetworks.com/) — хостинг-провайдер.
* [Substance-D](http://www.substanced.net/) — среда разработки веб-приложений.

3 **Flask**

Flask — это WSGI-фреймворк, который подходит и для простых сайтов, и для сложных платформ. Он совместим со сторонними библиотеками и имеет множество расширений.

**Возможности:**

* Быстрый встроенный дебаггер.
* Встроенный dev-сервер.
* Шаблоны Jinja2.
* Поддержка REST-запросов.
* Совместим с любой ORM.
* Безопасные куки для управления сессией на клиенте.
* Поддержка юнит-тестов.
* Соответствует WSGI 1.0.

**Плюсы**

* Понятный минималистичный фреймворк. Довольно легко разобраться, что именно происходит и какой процесс за что отвечает. Логика работы не меняется от версии к версии.
* Быстрое прототипирование. Все инструменты, нужные для создания прототипа, работают прямо из коробки.

**Минусы**

* Фреймворк относительно низкоуровневый, поэтому в нём надо сперва разобраться, да и требования к уровню программиста выше, чем у остальных. Если хочется не выяснять, что и как работает под капотом, а сразу делать сайты — лучше выбрать что-нибудь другое.

**Где используется?**

* [Rainist](https://rainist.com/) —онлайн-платформа для управления личными финансами.
* [Netflix](https://www.netflix.com/) — одна из крупнейших стриминговых платформ в мире.
* [Lyft](https://www.lyft.com/) — платформа для заказа такси и аренды транспорта.

4 Tornado



Асинхронный фреймворк, благодаря неблокирующему обмену данными способный одновременно поддерживать множество пользовательских соединений. Прекрасно подходит для задач, требующих подолгу поддерживать соединение с каждым пользователем.

**Возможности:**

* Есть свой механизм аутентификации, при необходимости можно подключить сторонние.
* Поддержка переводов и локализации.
* Работа в реальном времени.

**Где используется?**

* [Uploadcare](https://uploadcare.com/) — облачный сервис для работы с файлами.

Фреймворк Tornado отличается от большинства фреймворков веб-серверов (и, разумеется, большинства Python-фреимворков), так как он неблокирующий и достаточно быстрый. Tornado неблокирующий и использует [epoll](http://www.kernel.org/doc/man-pages/online/pages/man4/epoll.4.html), он может обрабатывать тысячи одновременных постоянных подключений, что делает его идеальным решением для веб-сервисов реального времени. Этот веб-сервер специально разработан для обслуживания возможностей, требующих работы в реальном времени — каждый активный пользователь FriendFeed поддерживает открытое соединение с серверами FriendFeed.

5 CherryPy



CherryPy — это веб-фреймворк Python, который предоставляет дружественный интерфейс для HTTP-протокола для разработчиков Python. Это также называют библиотекой веб-приложения.

CherryPy использует сильные стороны Python в качестве динамического языка для моделирования и связывания протокола HTTP в API. Это одна из старейших веб-платформ для Python, которая обеспечивает чистый интерфейс и надежную платформу.

Сильные стороны CherryPy

Следующие особенности CherryPy считаются его сильными сторонами —Простота

Разработка проекта в CherryPy — это простая задача с несколькими строками кода, разработанными в соответствии с соглашениями и отступами Python.

CherryPy также очень модульный. Основные компоненты хорошо управляются с правильной логикой, а родительские классы могут быть расширены до дочерних классов.

Мощность

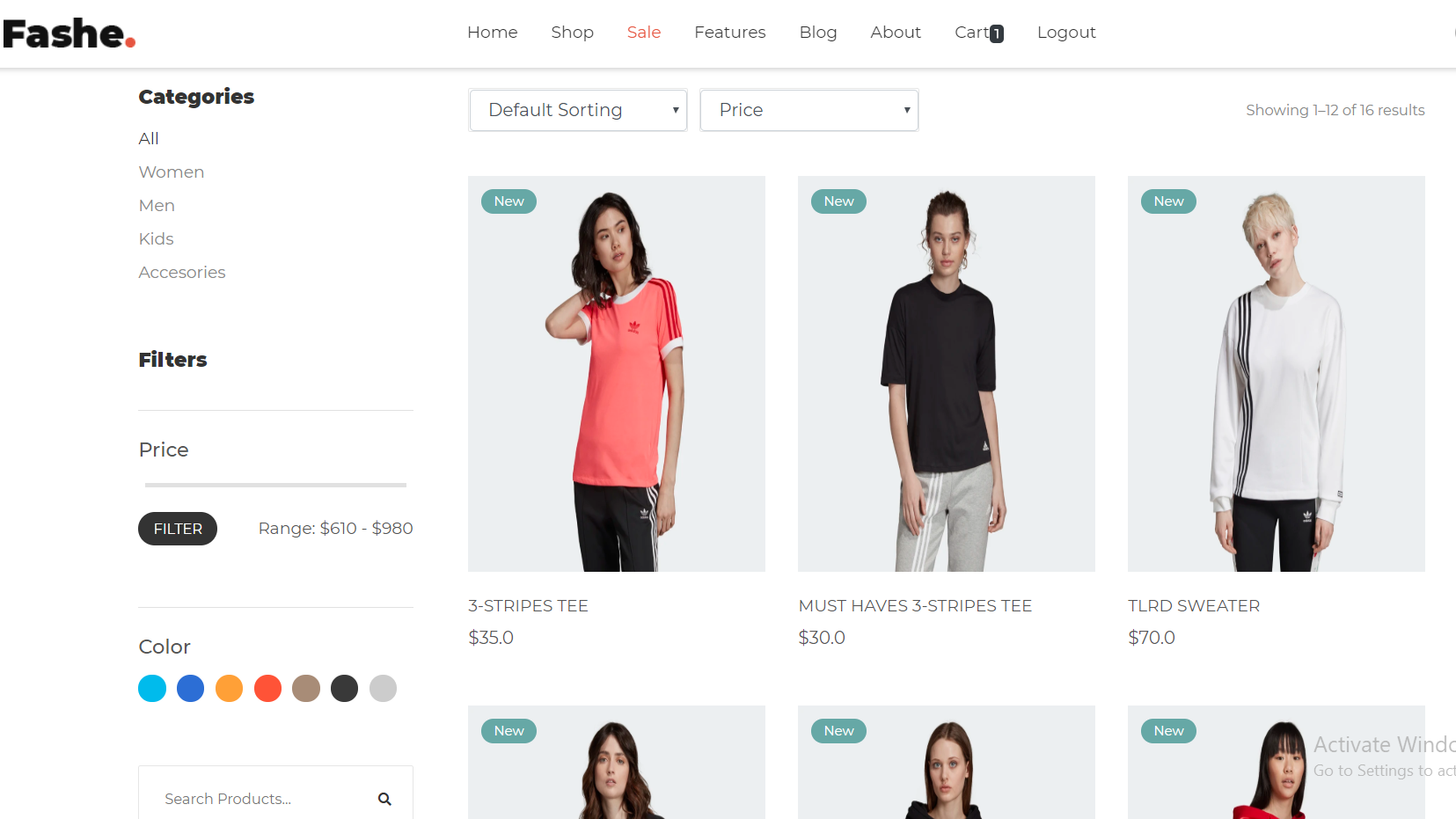
CherryPy использует всю мощь Python. Он также предоставляет инструменты и плагины, которые являются мощными точками расширения, необходимыми для разработки приложений мирового уровня.

Открытый исходный код

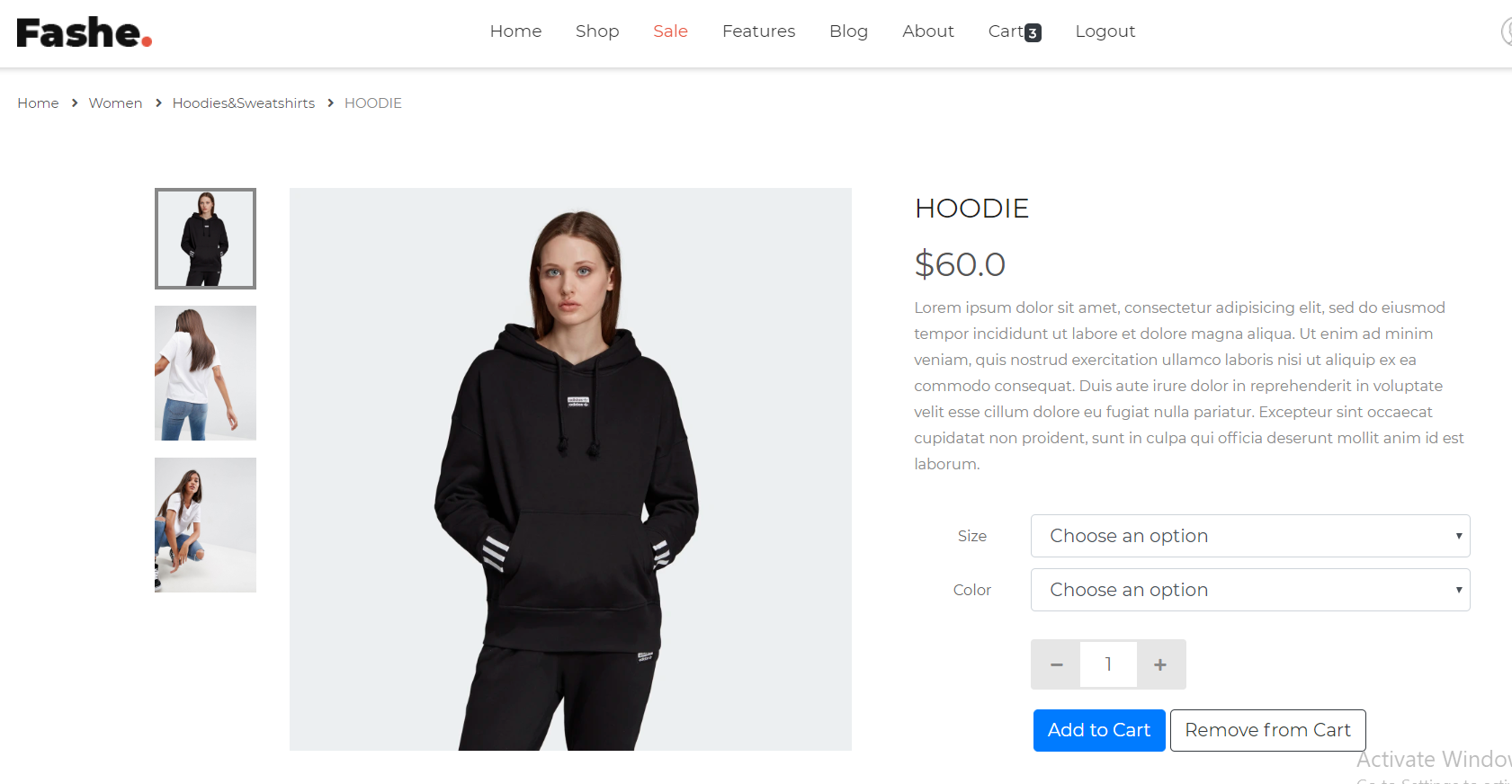
CherryPy — это Python Web Framework с открытым исходным кодом (лицензируется по лицензии BSD с открытым исходным кодом), что означает, что эта платформа может использоваться в коммерческих целях по нулевой цене.

**ГЛАВА 2 РАЗРАБОТКА ИНТЕРНЕТ-МАГАЗИНА С ПОМОЩЬЮ DJANGO**

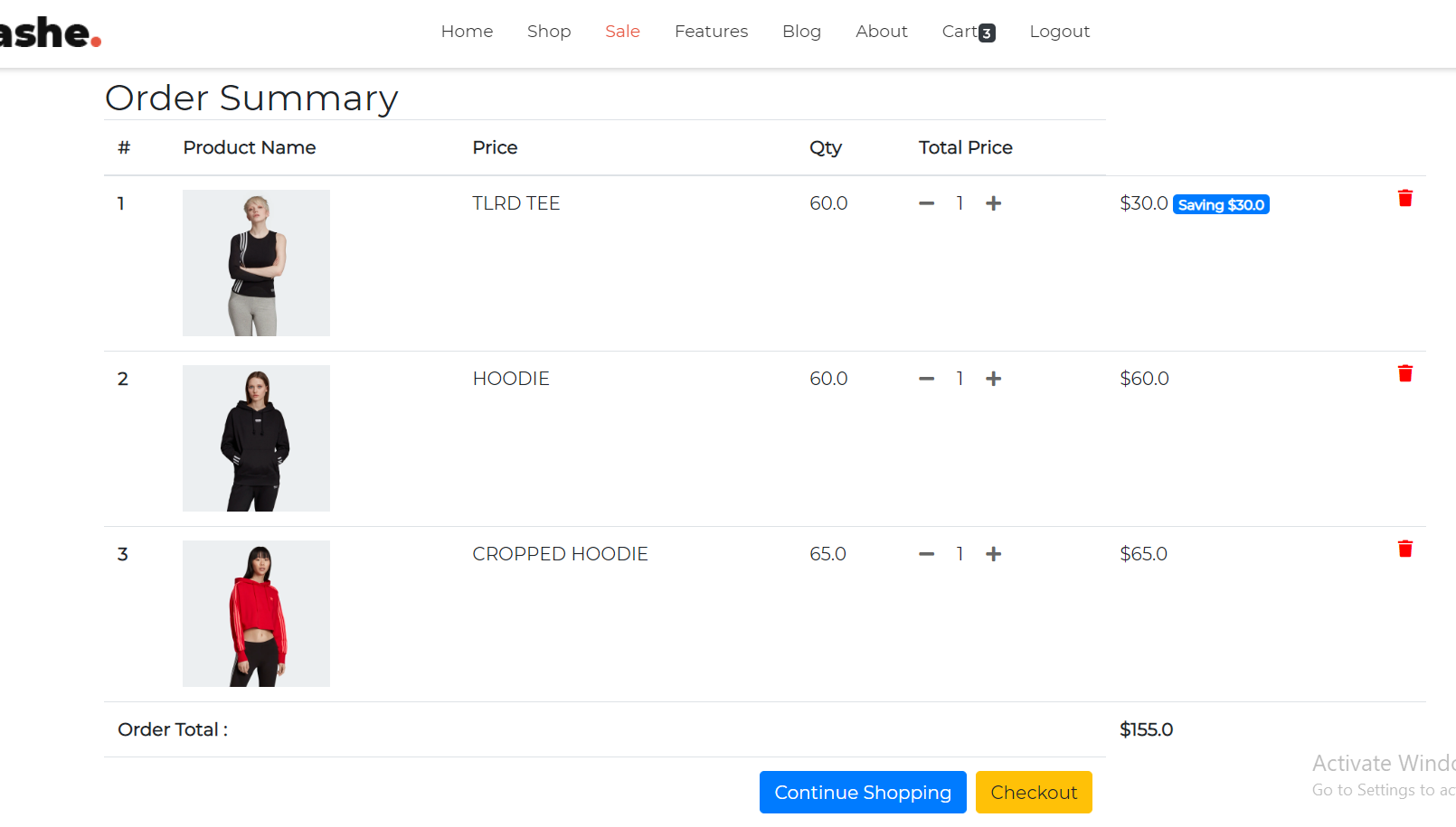
В первую очередь стоит начать с Главной страницы, на ней реализовано несколько блоков, которые отвечают за маркетинг. Также реализован набор пунктов необходимых для продуктивной работы на сайте. Главная страница должна быть яркой и броской, чтобы привлечь внимание покупателей.



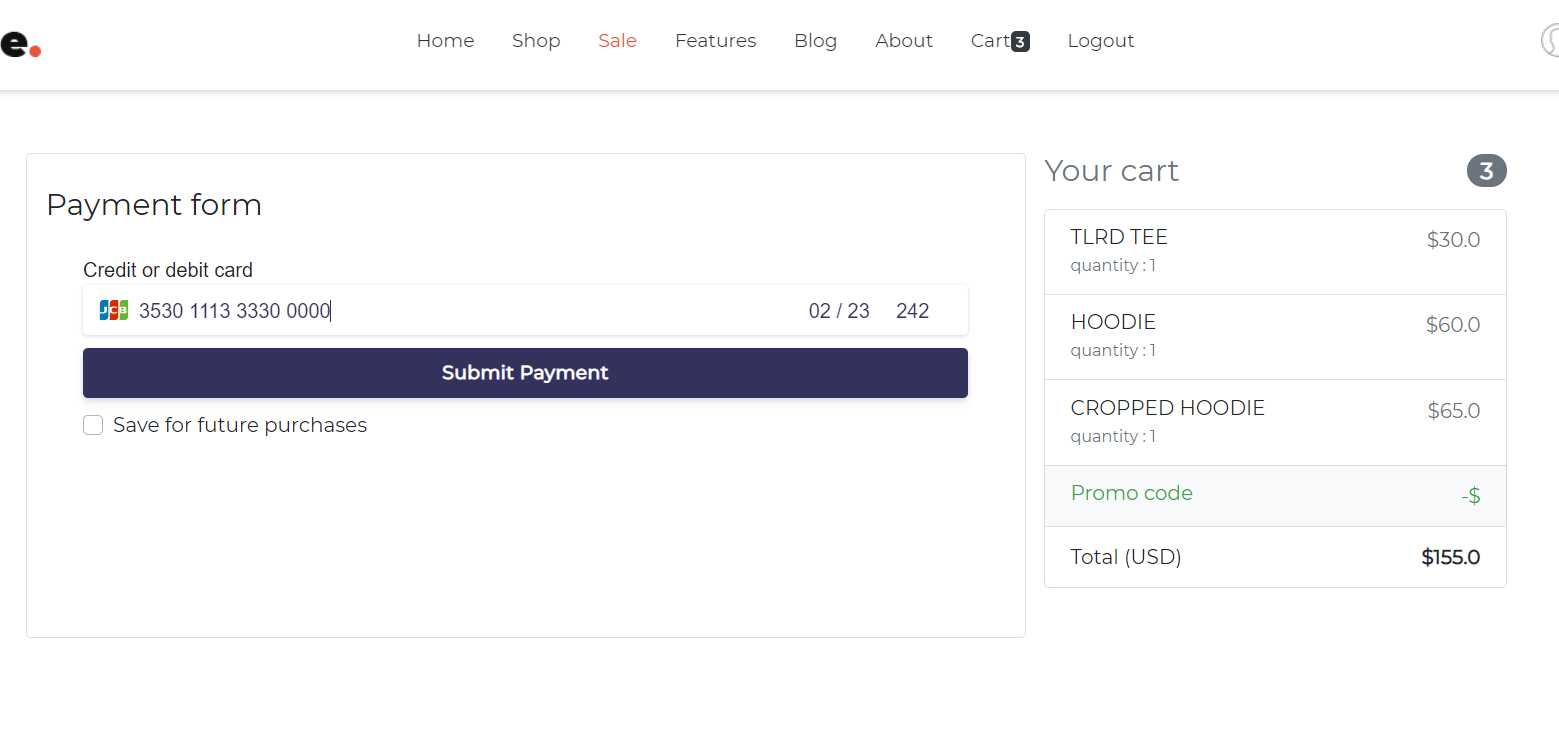
Далее необходимо рассмотреть страницу товара, здесь представлен сам товар, также представлено его описание, реализована возможность выбирать количество товара и добавлять товар в корзину. Также можно оставить комментарий к товару.



Покупатель путешествует по магазину и выбирает товары, все выбранные товары сохраняются в корзине. Товары отображаются вместе с картинкой и названием. Реализована возможность убрать товар из корзины, также увеличить или уменьшить его количество, реализована возможность применить промо-код.



Оформление заказа, происходит следующим образом, пользователь заполняет поля необходимые для доставки. Рядом с ввденными данными дублируется заказ покупателя, для того чтобы он мог его проверить, выводится сумма, чтобы клиент видел стоимость своего заказа. Пользователь может выбрать способ оплаты, либо оплатить заказ при получении наличными или же использовать платежную систему.



Также результат выполненной работы можно посмотреть на GitHub. Там подробно изложены все файл, а также архитектура Магазина.

HTML шаблон: <https://colorlib.com/etc/fashe/index.html>

GitHub : <https://github.com/art01alex/Django/>

# **ВЫВОД**

Исходя из проделанной мною работой, я могу смело сказать, что фреймворкDjango не зря является одним из самых популярных в мире. Потому что у него очень развитая экосистема, он постоянно изменяется и усовершенствуется, в случае возникновения трудностей найти ответ на конкретный вопрос не составляет труда, ведь тысячи специалистов уже решали такие же проблемы до нас и делились своим опытом в интернете. Помимо этого, Django поддерживает использование различных библиотек при разработке веб-приложений.

# **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Документация Django

