

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ
(підпис)

19 травня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-наукової програми «Інформатика»

на тему: «Інформаційна технологія тестування системи електронної комерції»
здобувача групи ІН.м-11н Коропець Максима Олександровича

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

_____ Максим Коропець
(підпис)

Керівник,
в.о. завідувача кафедри,
кандидат технічних наук, доцент

Ігор ШЕЛЕХОВ

_____ (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук, освітньо-наукової програми «Інформатика»
здобувача групи ІН.м-11н Коропець Максима Олександровича

1. Тема роботи: «Інформаційна технологія тестування системи електронної комерції»
затверджую наказом по СумДУ від «08» травня 2023 р. № 0475-VI _____
2. Термін здачі здобувачем кваліфікаційної роботи до 19 травня 2023 року _____
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.
2) Огляд технологій, що використовуються для прогнозування курсу валют. 3) Розробка
інтелектуальної системи з прогнозування курсу валют. 4) Аналіз результатів. _____
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх _____

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____

(підпис)

Керівник _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд технологій, що використовуються для створення системи електронної комерції</i>		
3	<i>Розробка системи електронної комерції</i>		
4	<i>Огляд технологій, що використовуються для тестування системи електронної комерції</i>		
5	<i>Тестування системи електронної комерції</i>		
6	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

АНОТАЦІЯ

Записка: 62 стор., 22 рис., 1 додаток, 20 джерел.

Обґрунтування актуальності теми роботи – Тема кваліфікаційної роботи є актуальною, оскільки присвячена електронній комерції, яка значно зросла за останні роки і стала важливою частиною бізнесу багатьох компаній.

Об’єкт дослідження — процес створення та тестування системи електронної комерції.

Мета роботи — розробка електронної системи комерції та її тестування.

Методи дослідження — алгоритми створення та тестування системи електронної комерції.

Результати — розроблено та протестовано електронну систему комерції за допомогою якої можна купляти та продавати різноманітні товари, будуючи особистий комерційний бізнес.

**СИСТЕМА ЕЛЕКТРОННОЇ КОМЕРЦІЇ, ТЕСТУВАННЯ СИСТЕМИ,
PYTHON, DJANGO, JIRA.**

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Дослідження актуальності проблеми	8
1.2 Аналіз алгоритмів створення електронної системи комерції	9
1.3 Постановка задачі	9
2 ВИБІР МЕТОДУ РІШЕННЯ.....	11
2.1 Переваги та недоліки розробки на Django.....	11
2.2 Вибір середовища розробки.....	14
2.3 Вибір бази даних	16
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	17
3.1 Проектування.....	17
3.2 Програмна реалізація.....	19
4 ТЕСТУВАННЯ	26
4.1 Створення задач в Jira.....	26
4.2 Створення чеклістів	27
4.3 Створення тест кейсів.....	28
4.4 Проходження тест кейсів	30
ВИСНОВКИ.....	31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	33
ДОДАТОК.....	35

ВСТУП

Актуальність. Системи електронної комерції все ще дуже важливі в сучасному світі. Популярність електронної комерції значно зросла за останні роки і стала важливою частиною бізнесу багатьох компаній. Вони забезпечують зручну платформу для продажу товарів і послуг через Інтернет, що дозволяє компаніям залучати нових клієнтів і збільшувати дохід.

Системи електронної комерції також дозволяють споживачам робити покупки в будь-який час і в будь-якому зручному для них місці. Вони можуть шукати та порівнювати продукти, читати відгуки та робити покупки безпосередньо зі своїх пристроїв.

Крім того, система електронної комерції відкриває нові можливості для малого бізнесу та підприємців. Вони дозволяють керувати онлайн-магазином за нижчою ціною, ніж традиційний звичайний магазин.

Загалом, система електронної комерції є актуальним для бізнесу та ефективним інструментом в епоху цифрової комерції. Вони допомагають збільшити продажі, сприяти споживачам і створювати нові можливості для бізнесу.

Об'єкт дослідження. Процес створення та тестування системи електронної комерції.

Предмет дослідження. Аналіз та вивчення процесів, методів та технологій, які використовуються в електронній комерції для здійснення торгівельних операцій через Інтернет

Гіпотеза. Персоналізований підхід може покращити взаємодію з користувачем і зробити процес пошуку та покупки продукту більш зручним і ефективним

Наукова новизна. Впровадження нових та інноваційних методів, технологій та концепцій для покращення торгових процесів в Інтернеті

Структура. Дане робота складається зі вступу, аналізу публікацій, постановки задачі дослідження, вибір методики та інструментів для рішення

поставленої проблеми, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Дослідження актуальності проблеми

Проблеми з системами електронної комерції залишаються через ряд важливих факторів:

Розвиток електронної комерції: використання онлайн-комерції підприємствами та споживачами значно зросло за останні роки. Це створює потребу в розвитку та вдосконаленні систем електронної комерції.

Конкуренція: посилення конкуренції в електронній комерції вимагає нових стратегій і рішень, щоб залишатися конкурентоспроможними. Ефективна та інноваційна система електронної комерції може стати ключовим фактором успіху бізнесу.

Безпека та довіра: питання безпеки в електронній комерції залишаються актуальними. Потрібні постійні вдосконалення захисту персональних даних споживачів, запобігання шахрайству та забезпечення надійних платіжних систем.

Технологічний розвиток: швидкі зміни в таких технологіях, як штучний інтелект, доповнена реальність, блокчейн та Інтернет речей, відкрили нові можливості для розвитку систем електронної комерції, що вимагає постійного дослідження та інтеграції цих технологій. Зміна моделей поведінки споживачів: передбачення та адаптація до змін споживчих пристрастей і вподобань є важливим аспектом електронної комерції.

Дослідження та розуміння нових тенденцій у поведінці споживачів, таких як зростаючий інтерес до екологічної стійкості, персоналізації та соціальної відповідальності, дозволяє компаніям адаптуватися до змін і використовувати їх у своєму бізнесі.

Міжнародні аспекти: електронна комерція відкриває можливості для глобальної торгівлі, але також створює певні проблеми, такі як міжнародна логістика, культурні відмінності, мовні бар'єри та юридичні нюанси. Розвиток

систем електронної комерції вимагає розуміння та вирішення цих складних міжнародних питань.

Загалом питання системи електронної комерції стають все більш актуальними через зміни в поведінці споживачів, технологічний розвиток і зростання конкуренції. Наукові дослідження та інновації в цій галузі сприяють розробці нових рішень, які відповідають сучасним вимогам ринку та підвищують ефективність електронної комерції.

1.2 Аналіз алгоритмів створення електронної системи комерції

Алгоритмічний аналіз для створення систем електронної комерції включає в себе розгляд різних аспектів, які впливають на розвиток і роботу системи. Важливі елементи для аналізу включають архітектуру системи, функціональні вимоги, взаємодію з користувачем, безпеку, інтеграцію та масштабованість, аналітику та звітність, мобільну доступність, технічну інфраструктуру, зручність використання та досвід користувача, тестування та перевірку, керування вмістом, підтримку та обслуговування, оптимізацію SEO, маркетинг інструменти, відповідність вимогам і нормам, аналіз конкурентів. Аналіз цих аспектів допомагає зрозуміти потреби та вимоги ринку, визначити найкраще рішення для розробки систем електронної комерції, підвищити зручність використання та задоволеність користувачів, забезпечити безпеку та надійність, покращити можливості маркетингу та аналізу та забезпечити інтеграцію з іншою системною інтеграцію, а також врахуйте законодавчі вимоги.

1.3 Постановка задачі

Завдання даної роботи складається з наступних етапів:

1. Дослідити популярні фреймворки розробки магазинів і вибрати найбільш підходящий фреймворк для розробки проекту.

2. Огляд популярних інтернет-магазинів і створення прототипу проекту, що характеризується відсутністю зайвої інформації про товар і забезпечує високу швидкість завантаження.

3. Розробка інтерфейсної частини та компонування компонентів проекту. Головні сторінки, які повинен містити магазин, це домашня сторінка, сторінки продуктів, сторінки реєстрації та входу, кошик для покупок та інформаційні сторінки.

4. Розробка back-end частини проекту, включаючи реалізацію логіки та функцій, необхідних для роботи магазину.

5. Тестування магазину, щоб виявити можливі помилки та виправити їх.

2 ВИБІР МЕТОДУ РІШЕННЯ

2.1 Переваги та недоліки розробки на Django



Рисунок 2.1 – Фреймворк Django

Коли у вас є ідея, ви можете використовувати Django, щоб швидко перетворити її на справжній веб-проект. Django — це безкоштовний інструмент, який спрощує процес веб-розробки, дозволяючи розробникам зосередитися на дизайні та функціональності програми. Django з'явився в 2005 році і став одним із найкращих фреймворків, які допомагають розробникам виконувати роботу за лічені хвилини. Спочатку розроблений як інструмент для мови програмування Python, цей фреймворк значно спрощує розробку веб-додатків, забезпечуючи простіший спосіб роботи. Завдяки своїй простоті, дизайну та гнучкості Python є однією з найпопулярніших мов програмування.

1. Продуктивність розробки: Django надає широкий спектр готових компонентів і бібліотек, які спрощують розробку веб-додатків. Він надає потужні інструменти для роботи з базами даних, обробки форм, автентифікації користувачів, керування інтерфейсами адміністратора та іншими частинами веб-додатків. Це дозволяє розробникам швидко створювати функціональні веб-проекти.

2. Масштабованість: Django підтримує масштабованість веб-додатків, дозволяючи їм рости та розширюватися зі збільшенням даних і робочого навантаження. Він забезпечує ефективне кешування, оптимізовану роботу з

базою даних і масштабування сервера, дозволяючи вам забезпечити високу продуктивність ваших веб-додатків навіть при великому навантаженні.

3. Безпека: Django має вбудовані засоби безпеки, які можуть допомогти уникнути багатьох загроз безпеці, таких як ін'єкція SQL, ін'єкція шкідливого коду та злом. Він забезпечує захист від викрадення сесії, CSRF-атак, проблем автентифікації та інших загроз безпеці.

4. Розширюваність: Django може легко розширити функціональність веб-додатків за допомогою сторонніх модулів і розширень. Багато готових пакетів Django є загальнодоступними, що полегшує додавання нових функцій до вашого проекту. Крім того, Django добре інтегрується з іншими технологіями та може використовуватися з іншими фреймворками та бібліотеками, такими як React, Angular, Flask тощо.

5. Швидкість розробки: Django пропонує концепцію «батареї включені», що означає наявність вбудованих інструментів і шаблонів, які дозволяють швидко розпочати розробку веб-проектів. Замість того, щоб писати основні функціональні компоненти з нуля, ви можете зосередитися на важливих аспектах вашого проекту, таких як дизайн і бізнес-логіка.

6. Спільнота та документація: Django має велике та активне співтовариство розробників, що означає, що ви знайдете багато ресурсів, посібників, документації та підтримки від інших розробників. Це спрощує дослідження та вирішення проблем, з якими ви можете зіткнутися під час розробки проекту.

7. Гнучкість бази даних і масштабованість: Django надає вдосконалену ORM (Object Relational Mapping), яка дозволяє легко використовувати різні системи керування базами даних, такі як MySQL, PostgreSQL, SQLite тощо. Ви можете легко взаємодіяти з базою даних, створювати складні запити та легко змінювати модель даних.[5]

Загалом, Django є потужним інструментом для швидкого і ефективного створення веб-проектів. Він пропонує багато переваг, що роблять його ідеальним вибором для розробки веб-додатків будь-якого розміру та складності.

Для початку розробки сайту з Django, потрібно встановити фреймворк. Виконайте наступну команду у командному рядку:

```
pip install Django==1.9.1
```

Щоб перевірити правильність встановлення Django, запустіть команду `django-admin --version`

Вона відобразить поточну версію фреймворку.

Далі створіть новий проект за допомогою команди:

```
django-admin startproject django_example
```

Після завершення створення проекту, структура директорії буде такою:

`django_example/__/__init__.py` - порожній файл, який показує, що ця директорія є пакетом Python.

`django_example/settings.py` - містить конфігурацію проекту.

`django_example/urls.py` - вказує URL-шаблони.

`django_example/wsgi.py` - дозволяє взаємодіяти з веб-сервером через WSGI.

`manage.py` - інтерфейс командного рядка для взаємодії з проектом.

Тепер можна запустити додаток, виконавши команду:

```
python manage.py runserver
```

Потім у браузері перейдіть за адресою `http://127.0.0.1:8000/`. Якщо все працює коректно, ви побачите підтвердження успішного запуску.

Далі створимо найпростіший додаток. Виконайте наступну команду:

```
python manage.py startapp riddles
```

Після створення додатку, можна написати простий вигляд (view) у файлі `riddles/views.py`, відповідно до правил Django.

```
1 | from django.http import HttpResponse
2 |
3 |
4 | def index(request):
5 |     return HttpResponse("Hello, World!")
```

Рисунок 2.2 – Файл `views.py`

Тепер, щоб прив'язати наш вид до URL, створимо файл `urls.py`

```
1 | from django.conf.urls import url
2 |
3 | from . import views
4 |
5 | app_name = 'riddles'
6 |
7 | urlpatterns = [
8 |     url(r'^$', views.index, name='index'),
9 | ]
```

Рисунок 2.3 – Файл `urls.py`

У `urls.py` ми повинні написати наступне:

```
1 | from django.conf.urls import include, url
2 | from django.contrib import admin
3 |
4 | urlpatterns = [
5 |     url(r'^riddles/', include('riddles.urls')),
6 |     url(r'^admin/', admin.site.urls),
7 | ]
```

Рисунок 2.4 – Файл `urls.py`

Тепер, якщо ми запусимо наш додаток `http://127.0.0.1:8000/riddles/` ми побачимо «Hello, World!».

2.2 Вибір середовища розробки

PyCharm - це інтегроване середовище розробки (IDE) для мови програмування Python, розроблене компанією JetBrains. Воно надає розширені можливості розробки, забезпечуючи розробникам зручне та продуктивне середовище для писання, відлагодження і впровадження програм на Python.[6]

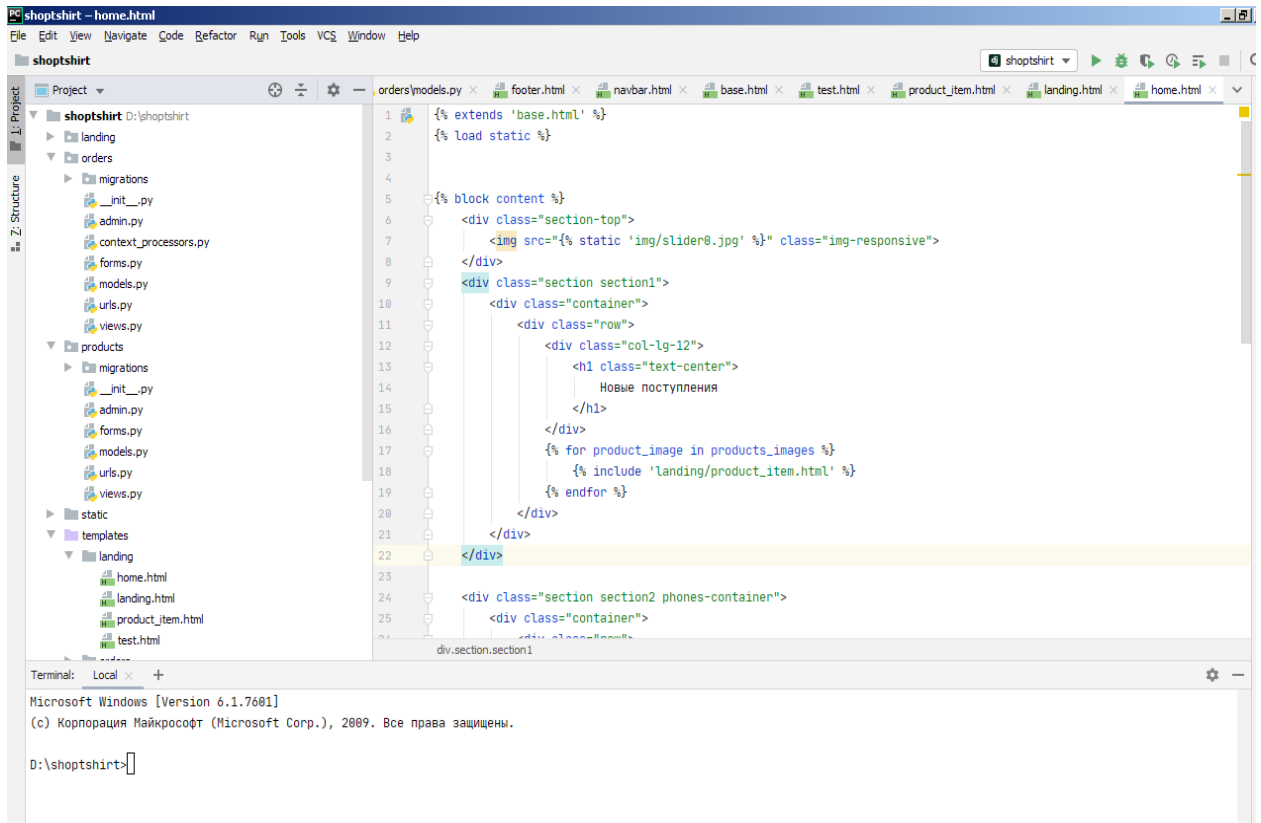


Рисунок 2.5 – Інтерфейс PyCharm

Основні переваги PyCharm:

1. Розширені можливості: PyCharm надає багатий набір інструментів, що полегшують процес розробки. Його функціональність включає автодоповнення коду, підказки, рефакторинг, відлагодження, керування залежностями, інтеграцію з системами контролю версій та багато іншого. Це допомагає збільшити продуктивність розробників і покращити якість коду.

2. Підтримка віртуальних середовищ: PyCharm має вбудовану підтримку віртуальних середовищ, таких як Virtualenv, Pyenv та Conda. Це дозволяє розробникам легко створювати та керувати окремими середовищами для своїх проектів, що сприяє ізолюваності та керуваності залежностей.

3. Інтегрована система відлагодження: PyCharm надає потужну систему відлагодження, яка дозволяє розробникам крокувати по коду, встановлювати точки зупинки, переглядати значення змінних та виконувати інші дії для аналізу та виправлення проблем у програмі.

4. Підтримка рефакторингу: PyCharm допомагає розробникам проводити

рефакторинг коду, тобто вносити зміни до структури та організації програми без зміни її зовнішньої поведінки. Це зроблено для полегшення підтримки та розширення проєктів.

2.3 Вибір бази даних



Рисунок 2.6 – SQLite

SQL (Structured Query Language) - це стандартна мова запитів, яка [8]використовується для взаємодії з реляційними базами даних. Вона дозволяє створювати, змінювати та вибирати дані з баз даних за допомогою запитів. SQL забезпечує можливість створення таблиць, вставки, оновлення та видалення даних, виконання операцій об'єднання, сортування та фільтрації, а також використання функцій та групування даних. Вона широко використовується в розробці баз даних, аналітиці даних та багатьох інших сферах, де потрібно працювати зі структурованими даними.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Проєктування

Проєктування системи електронної комерції (e-commerce) включає такі кроки:

- 1.Визначення бізнес-потреб: Визначення бізнес-цілей, стратегій і основних потреб. Це може включати вибір цільової аудиторії, тип пропонованих товарів або послуг і бізнес-модель, яку використовуватиме система електронної комерції.
- 2.Виберіть технологічний стек: виберіть платформу або структуру, яка відповідає потребам проєкту. Це можуть бути готові рішення, такі як Magento, Shopify або WooCommerce, або самостійно розроблені з використанням різних технологій, таких як веб-інтерфейс (HTML/CSS/JavaScript), серверні мови програмування, такі як PHP, Python, Java система) і дані бази даних
- 3.Функціональний розвиток: визначте основні характеристики та функції системи, такі як реєстрація користувачів, каталог продуктів, кошик для покупок, оформлення замовлення, оплата, система доставки, обробка повернень тощо. Розробка повинна враховувати зручність використання для користувачів і забезпечувати безпеку і захист персональних даних.
4. Дизайн інтерфейсу користувача: розробіть зручний і привабливий дизайн, який відповідає бренду компанії та забезпечує легку навігацію для користувачів. Важливо врахувати мобільну сумісність, щоб забезпечити комфортну роботу на різних пристроях і роздільній здатності екрана.
5. Розробка бази даних: створіть ефективну базу даних, яка зберігає інформацію про користувачів, продукти, замовлення, платежі та інші необхідні дані. Дуже важливо правильно спроектувати структуру бази

даних, визначити зв'язок між таблицями, забезпечити безпеку і надійність зберігання даних.

6. Інтеграція зовнішніх послуг: за потреби системи електронної комерції можна інтегрувати з різними зовнішніми службами, такими як платіжні шлюзи (наприклад, PayPal, Stripe), системи доставки (наприклад, FedEx, UPS) або системи управління запасами. Це допоможе автоматизувати процес і забезпечить більш зручну роботу для користувачів.

7. Тестування та перевірка: Перевірка правильності функціонування системи, виявлення та виправлення помилок, а також перевірка відповідності вимогам і очікуванням користувачів. Важливо виконати різноманітні тести, включаючи перевірку безпеки, навантаження та тестування на сумісність.

Розробка системи електронної комерції вимагає ретельного аналізу, розробки та тестування для створення функціональних можливостей та інтерфейсів, які відповідають потребам бізнесу та користувачів.

Діаграма сайту DFD (схема потоку даних) — це графічне представлення, яке показує потік даних і процеси, що відбуваються на веб-сайті. Це допомагає зрозуміти, як дані перетікають між різними компонентами системи та як вони обробляються. Діаграма DFD веб-сайту може краще зрозуміти взаємозв'язок між різними компонентами системи та потоками даних, що корисно для розробки та аналізу систем електронної комерції.

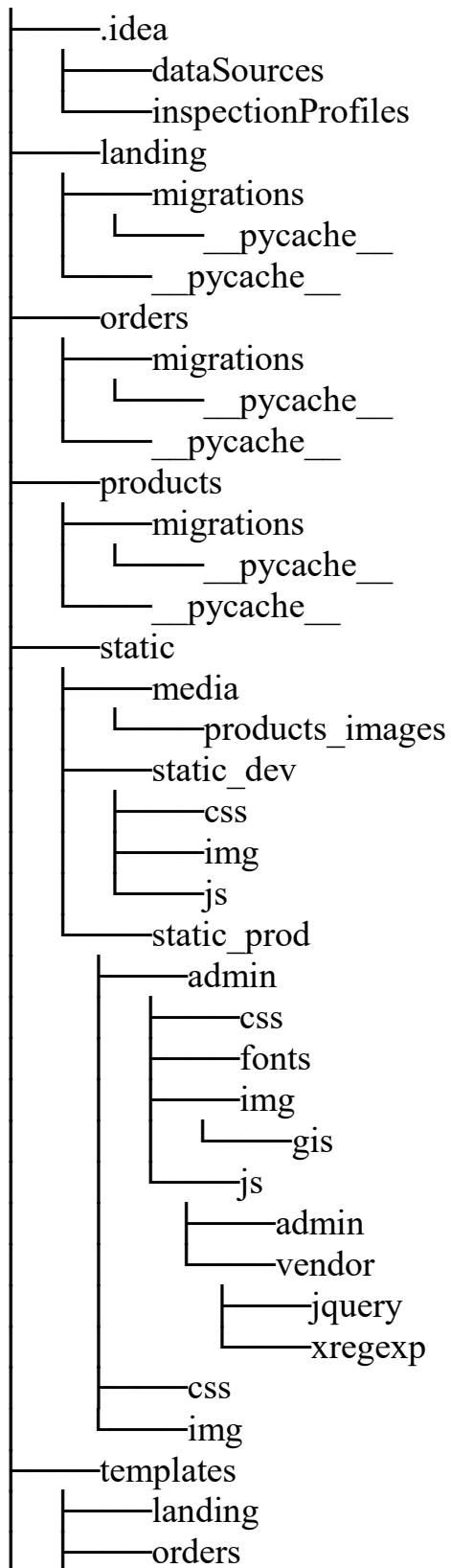


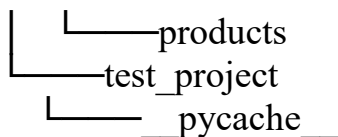
Рисунок 3.1 –DFD діаграма

На DFD діаграмі, рисунок 3.1, можна бачити процес замовлення товару

3.2 Програмна реалізація

Створюємо структуру файлів інтернет магазину





Проект складається з:

- папки landing
- папки orders,
- папки products ,
- папки static ,
- папки templates ,
- папки test_project
- файл бази даних db.sqlite3
- файлу manage.py

Вся файлова структура проекту зображена на рис.3.2

Имя ^	Дата изменения	Тип	Размер
.idea	03.06.2020 11:38	Папка с файлами	
landing	02.06.2020 2:37	Папка с файлами	
orders	02.06.2020 2:38	Папка с файлами	
products	02.06.2020 2:37	Папка с файлами	
static	30.05.2017 16:14	Папка с файлами	
templates	02.06.2020 23:40	Папка с файлами	
test_project	02.06.2020 2:37	Папка с файлами	
.gitignore	30.05.2017 16:14	Файл "GITIGNORE"	2 КБ
db.sqlite3	03.06.2020 10:39	Файл "SQLITE3"	88 КБ
diagram.jpg	30.05.2017 16:14	Рисунок JPEG	74 КБ
manage.py	30.05.2017 16:14	JetBrains PyCharm	1 КБ

Рисунок 3.2 – Структура файлів проекту

Опис структури деяких файлів:

- settings.py містить в собі всі настройки проекту. Тут ми реєструємо додатки, задаємо розміщення статичних файлів, налаштування бази даних і так далі.
- urls.py задає асоціації url адрес з уявленнями.
- wsgi.py використовується для налагодження зв'язку між вашим Django додатком і веб-сервером.

- Файл manage.py використовується для створення програм, роботи з базами даних.

В папці orders знаходяться файли для конфігурації вікна заказів та доставки

В папці products знаходяться файли для конфігурації вікна товарів. А в products-images зображення всіх товарів.

В static розташовуються файли стилів сайту, скрипти, зображення і т.д.

В templates розташовуються макети сторінок сайту

Всі таблиці в базі даних db.sqlite3

Таблиця	SQL
auth_group	CREATE TABLE "auth_group" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(80) NOT NULL UNIQUE)
auth_group_permissions	CREATE TABLE "auth_group_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "group_id" integer NOT NULL REFERENCES "auth_group" ("id"), "permis
auth_permission	CREATE TABLE "auth_permission" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "content_type_id" integer NOT NULL REFERENCES "django_content_type" ("id")
auth_user	CREATE TABLE "auth_user" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "password" varchar(128) NOT NULL, "last_login" datetime NULL, "is_superuser" bool N
auth_user_groups	CREATE TABLE "auth_user_groups" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id"), "group_id" integ
auth_user_user_permissions	CREATE TABLE "auth_user_user_permissions" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "user_id" integer NOT NULL REFERENCES "auth_user" ("id"), "permi
django_admin_log	CREATE TABLE "django_admin_log" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "object_id" text NULL, "object_repr" varchar(200) NOT NULL, "action_flag" sr
django_content_type	CREATE TABLE "django_content_type" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app_label" varchar(100) NOT NULL, "model" varchar(100) NOT NULL)
django_migrations	CREATE TABLE "django_migrations" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "app" varchar(255) NOT NULL, "name" varchar(255) NOT NULL, "applied" dat
django_session	CREATE TABLE "django_session" ("session_key" varchar(40) NOT NULL PRIMARY KEY, "session_data" text NOT NULL, "expire_date" datetime NOT NULL)
landing_subscriber	CREATE TABLE "landing_subscriber" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "email" varchar(254) NOT NULL, "name" varchar(128) NOT NULL)
orders_order	CREATE TABLE "orders_order" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "customer_name" varchar(64) NULL, "customer_email" varchar(254) NULL, "custon
orders_productinbasket	CREATE TABLE "orders_productinbasket" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "hmb" integer NOT NULL, "price_per_item" decimal NOT NULL, "total_pri
orders_productinorder	CREATE TABLE "orders_productinorder" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "is_active" bool NOT NULL, "created" datetime NOT NULL, "updated" date
orders_status	CREATE TABLE "orders_status" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(24) NULL, "is_active" bool NOT NULL, "created" datetime NOT NUL
products_product	CREATE TABLE "products_product" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(64) NULL, "description" text NULL, "is_active" bool NOT NULL,
products_productcategory	CREATE TABLE "products_productcategory" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "name" varchar(64) NULL, "is_active" bool NOT NULL)
products_productimage	CREATE TABLE "products_productimage" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "image" varchar(100) NOT NULL, "is_active" bool NOT NULL, "created" d
sqlite_sequence	CREATE TABLE sqlite_sequence(name,seq)
auth_group_permissions_0e939a4f	CREATE INDEX "auth_group_permissions_0e939a4f" ON "auth_group_permissions" ("group_id")
auth_group_permissions_8373b171	CREATE INDEX "auth_group_permissions_8373b171" ON "auth_group_permissions" ("permission_id")
auth_group_permissions_group_id_0cd325b0...	CREATE UNIQUE INDEX "auth_group_permissions_group_id_0cd325b0_uniq" ON "auth_group_permissions" ("group_id", "permission_id")
auth_permission_417f1b1c	CREATE INDEX "auth_permission_417f1b1c" ON "auth_permission" ("content_type_id")
auth_permission_content_type_id_01ab375a...	CREATE UNIQUE INDEX "auth_permission_content_type_id_01ab375a_uniq" ON "auth_permission" ("content_type_id", "codename")
auth_user_groups_0e939a4f	CREATE INDEX "auth_user_groups_0e939a4f" ON "auth_user_groups" ("group_id")
auth_user_groups_e8701ad4	CREATE INDEX "auth_user_groups_e8701ad4" ON "auth_user_groups" ("user_id")
auth_user_groups_user_id_94350c0c_uniq	CREATE UNIQUE INDEX "auth_user_groups_user_id_94350c0c_uniq" ON "auth_user_groups" ("user_id", "group_id")
auth_user_user_permissions_8373b171	CREATE INDEX "auth_user_user_permissions_8373b171" ON "auth_user_user_permissions" ("permission_id")
auth_user_user_permissions_e8701ad4	CREATE INDEX "auth_user_user_permissions_e8701ad4" ON "auth_user_user_permissions" ("user_id")
auth_user_user_permissions_user_id_14a6b6...	CREATE UNIQUE INDEX "auth_user_user_permissions_user_id_14a6b632_uniq" ON "auth_user_user_permissions" ("user_id", "permission_id")
django_admin_log_417f1b1c	CREATE INDEX "django_admin_log_417f1b1c" ON "django_admin_log" ("content_type_id")
django_admin_log_e8701ad4	CREATE INDEX "django_admin_log_e8701ad4" ON "django_admin_log" ("user_id")
django_content_type_app_label_76bd3d3b_u...	CREATE UNIQUE INDEX "django_content_type_app_label_76bd3d3b_uniq" ON "django_content_type" ("app_label", "model")
django_session_de54fa62	CREATE INDEX "django_session_de54fa62" ON "django_session" ("expire_date")
orders_order_dc91ed4b	CREATE INDEX "orders_order_dc91ed4b" ON "orders_order" ("status_id")
orders_order_e8701ad4	CREATE INDEX "orders_order_e8701ad4" ON "orders_order" ("user_id")
orders_productinbasket_69dfcb07	CREATE INDEX "orders_productinbasket_69dfcb07" ON "orders_productinbasket" ("order_id")
orders_productinbasket_9bea82de	CREATE INDEX "orders_productinbasket_9bea82de" ON "orders_productinbasket" ("product_id")
orders_productinorder_69dfcb07	CREATE INDEX "orders_productinorder_69dfcb07" ON "orders_productinorder" ("order_id")
orders_productinorder_9bea82de	CREATE INDEX "orders_productinorder_9bea82de" ON "orders_productinorder" ("product_id")
products_product_b583a629	CREATE INDEX "products_product_b583a629" ON "products_product" ("category_id")
products_productimage_9bea82de	CREATE INDEX "products_productimage_9bea82de" ON "products_productimage" ("product_id")

Рисунок 3.3 – Таблиці бази даних sqlite3

Django administration WELCOME, ADMIN. V

Site administration

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add Change
Users	+ Add Change

LANDING	
A lot of Subscribers	+ Add Change

ORDERS	
Заказы	+ Add Change
Статусы заказа	+ Add Change
Товары в заказе	+ Add Change
Товары в корзине	+ Add Change

PRODUCTS	
Категория товаров	+ Add Change
Товары	+ Add Change
Фотографии	+ Add Change

Recent actions

My actions

- [320.00, Футболка Nirvana](#)
Товар
- [400.00, Футболка Metallica](#)
Товар
- [320.00, Футболка Nirvana](#)
Товар
- [320, Apple MacBook 12" Retina](#)
Товар
- [280.00, Футболка Iron Maiden](#)
Товар
- [280.00, Футболка Iron Maiden](#)
Товар
- [400.00, Футболка](#)
Товар
- [200.00, Футболка AC DC](#)
Товар
- [Пользуются популярностью](#)
Категория товара
- [Новая коллекция](#)
Категория товара

Рисунок 3.4 – Зовнішній вигляд адмін-панелі Django

TSHOP Корзина (0)

Домой Доставка Контакты Регистрация -

TSHOP

RESPONSIVE ECOMMERCE TEMPLATE

Новые поступления










Рисунок 3.5 – Зовнішній вигляд верхньої частини головної сторінки

Контентом головної сторінки є баннер, який зображений на рисунку 3.5, з усіма товарами. Під яким знаходяться різні категорії товарів. На товарі є можливість або відкрити повну сторінку товару та кнопка додати в корзину, також під фото є короткий опис товару. Зовнішній вигляд картки відображений на рисунку 3.6.

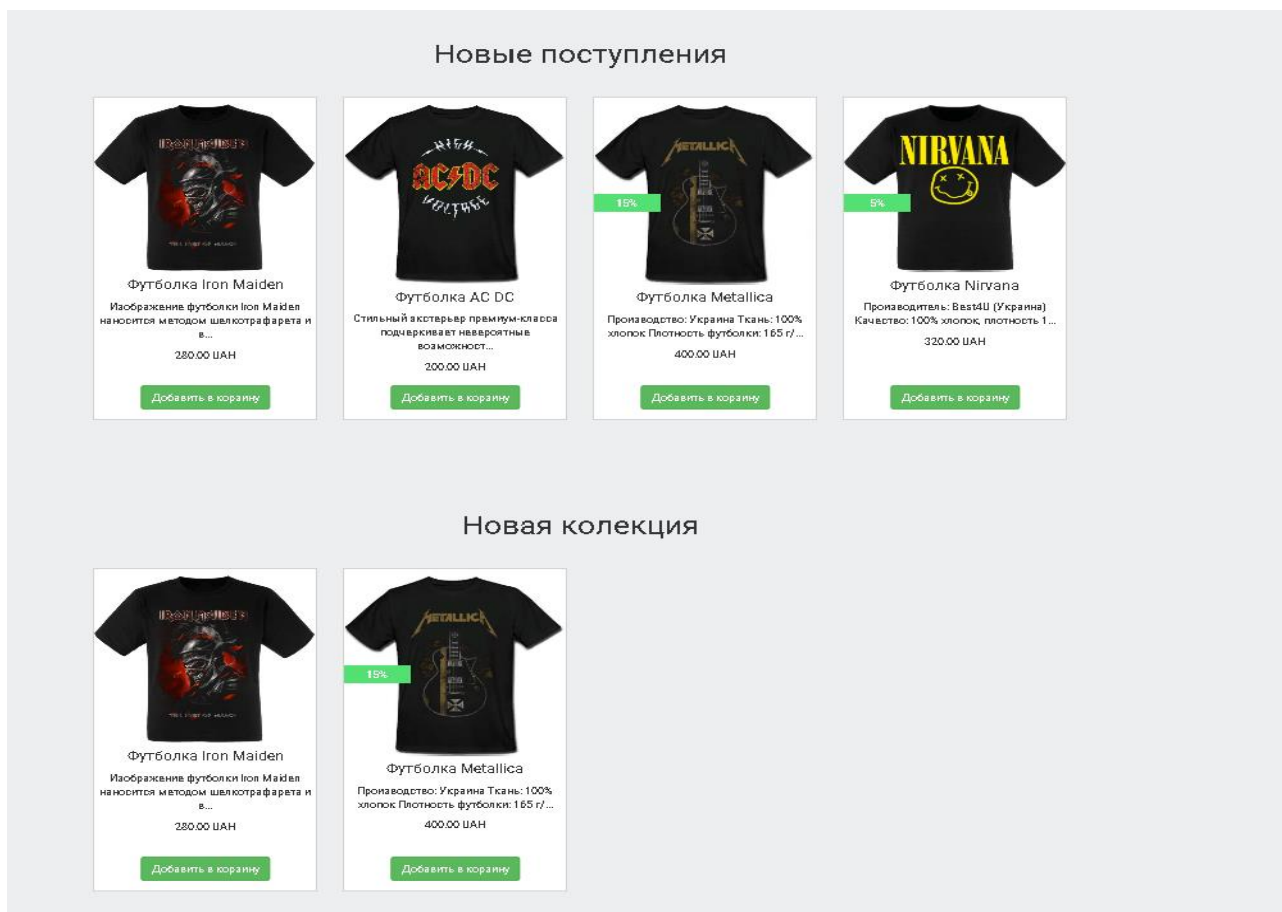


Рисунок 3.6 – Зовнішній вигляд нижньої частини головної сторінки



Футболка Iron Maiden

Изображение футболки Iron Maiden наносится методом шелкотрафарета и В...

280.00 UAH

Добавить в корзину

Рисунок 3.7 – Зовнішній вигляд картки товару

Також на картці товару використовується кнопка для того щоб було можна відразу додати товар в корзину з головної сторінки.

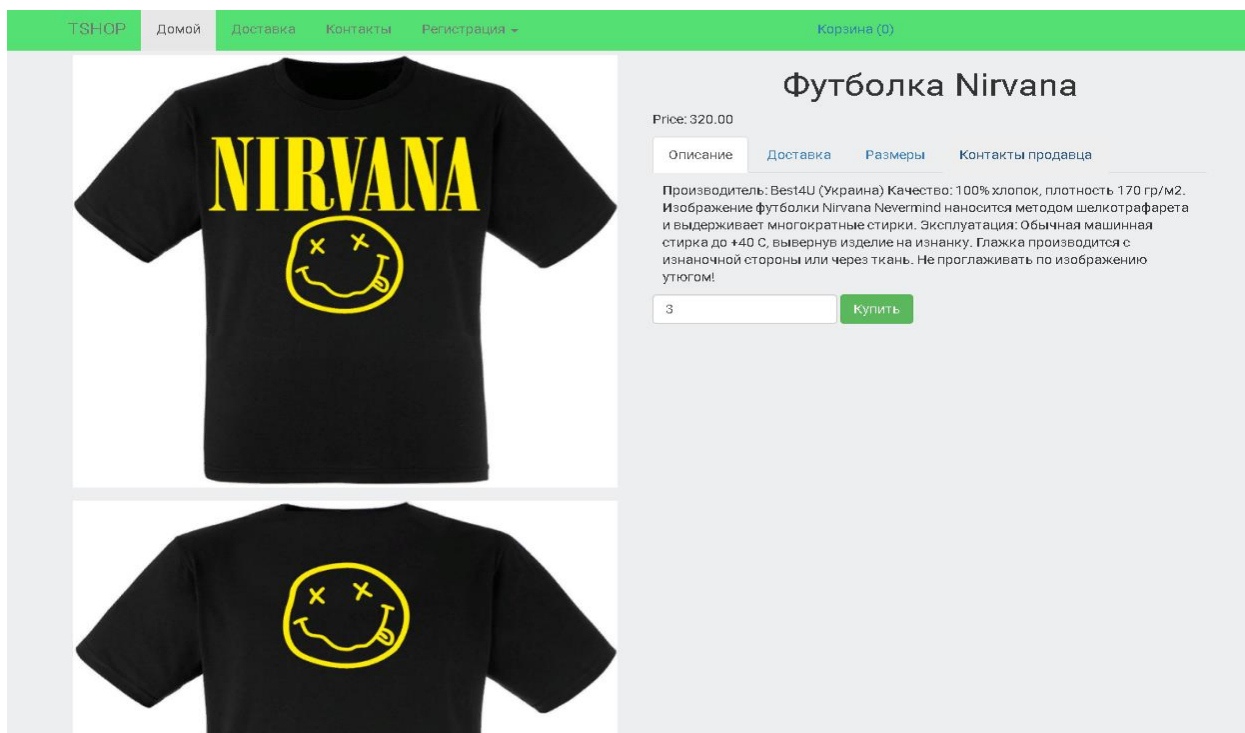


Рисунок 3.8 – Зовнішній вигляд сторінки товару

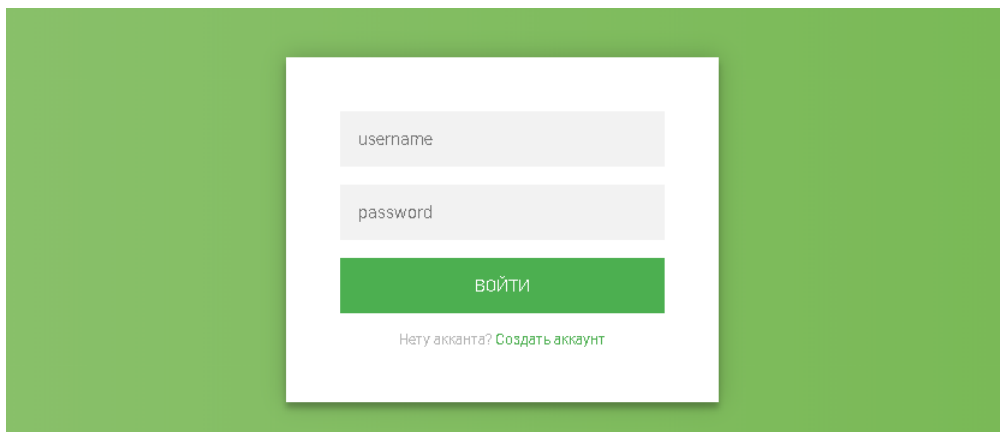


Рисунок 3.8 – Зовнішній вигляд сторінки входу

На сторінці Реєстрації в порівнянні з сторінкою Входу є додаткове поле email яке відповідає за додаткову перевірку паролю, для того щоб успішно завершити процедуру створення акаунту необхідно в нього ввести свій email.

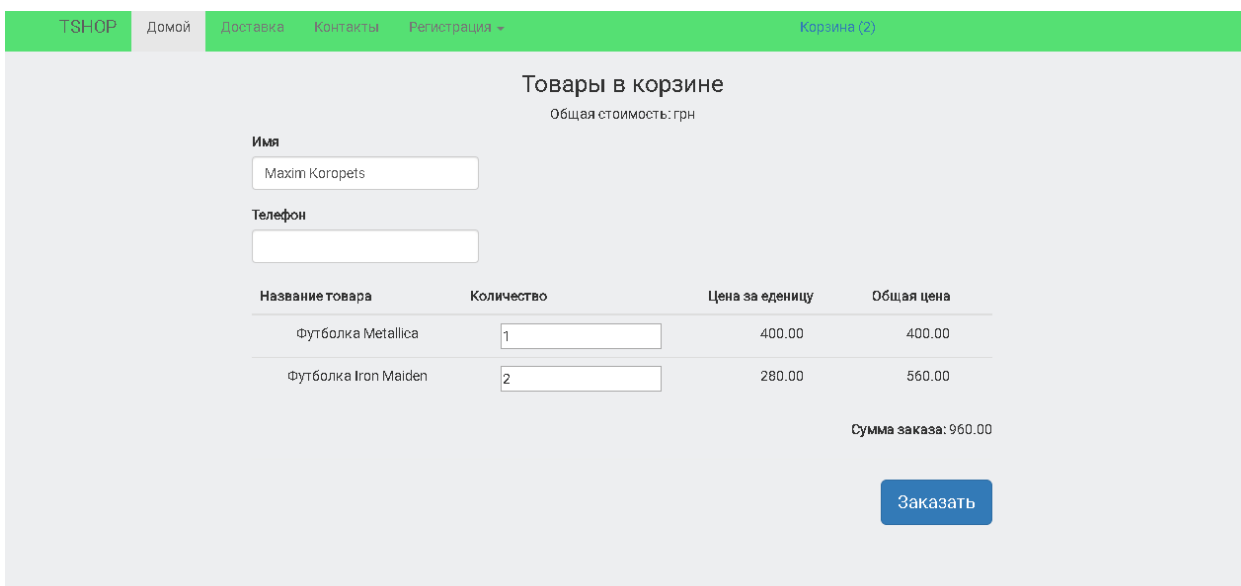


Рисунок 3.8 – Зовнішній вигляд сторінки корзини

Тут відображаються товари які були раніше добавлені сюди через картку товару з головної сторінки або через сторінку товару

4 ТЕСТУВАННЯ

4.1 Створення задач в Jira



Рисунок 4.1 – Jira

Jira - це популярна платформа для керування проектами і відстеження задач, [19]розроблена компанією Atlassian. Jira надає широкий набір інструментів для планування, відстеження та керування проектами, співпраці команди і виконання завдань. Вона дозволяє створювати проекти, задачі, тикети або баги, встановлювати пріоритети, вказувати терміни виконання, відстежувати прогрес та комунікувати з командою через коментарі. Jira також має розширені можливості звітності і аналітики, що допомагають керівникам проектів отримувати уявлення про стан проектів і ресурсне планування. Вона широко використовується в розробці програмного забезпечення, агільних методологіях, IT-проектах та багатьох інших галузях для ефективного управління робочим процесом і забезпечення спільної роботи команди.

Створюємо задачі для подальшого тестування за допомогою Jira

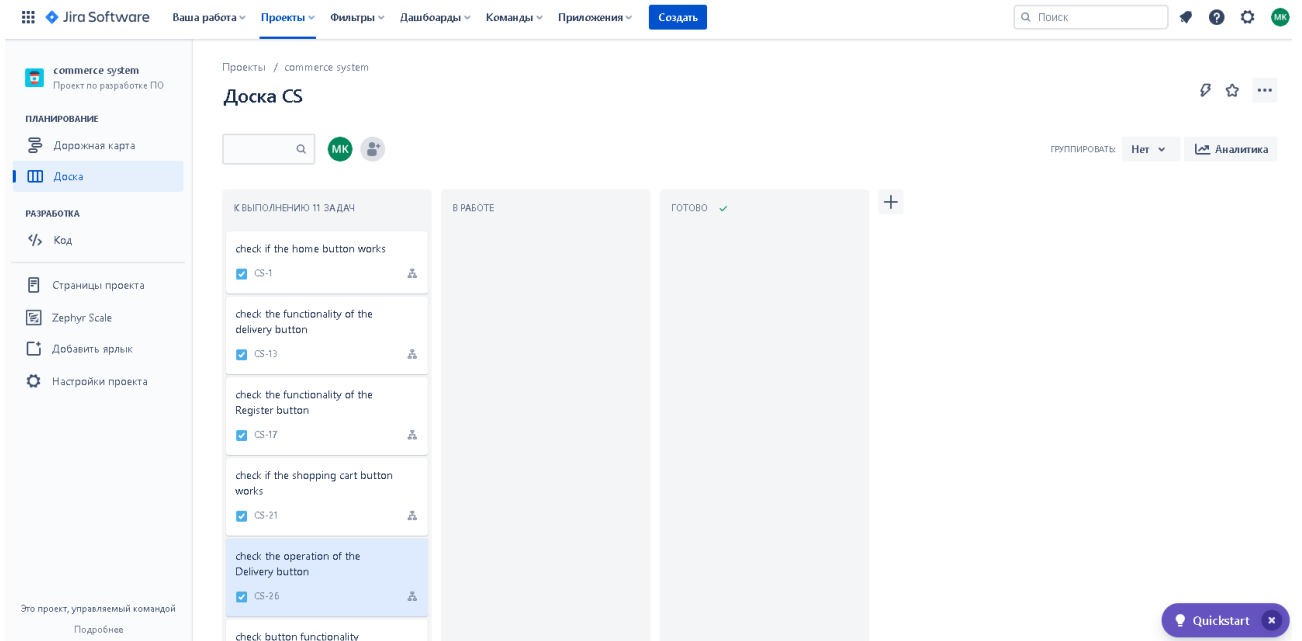


Рисунок 4.2 – Створення taskів

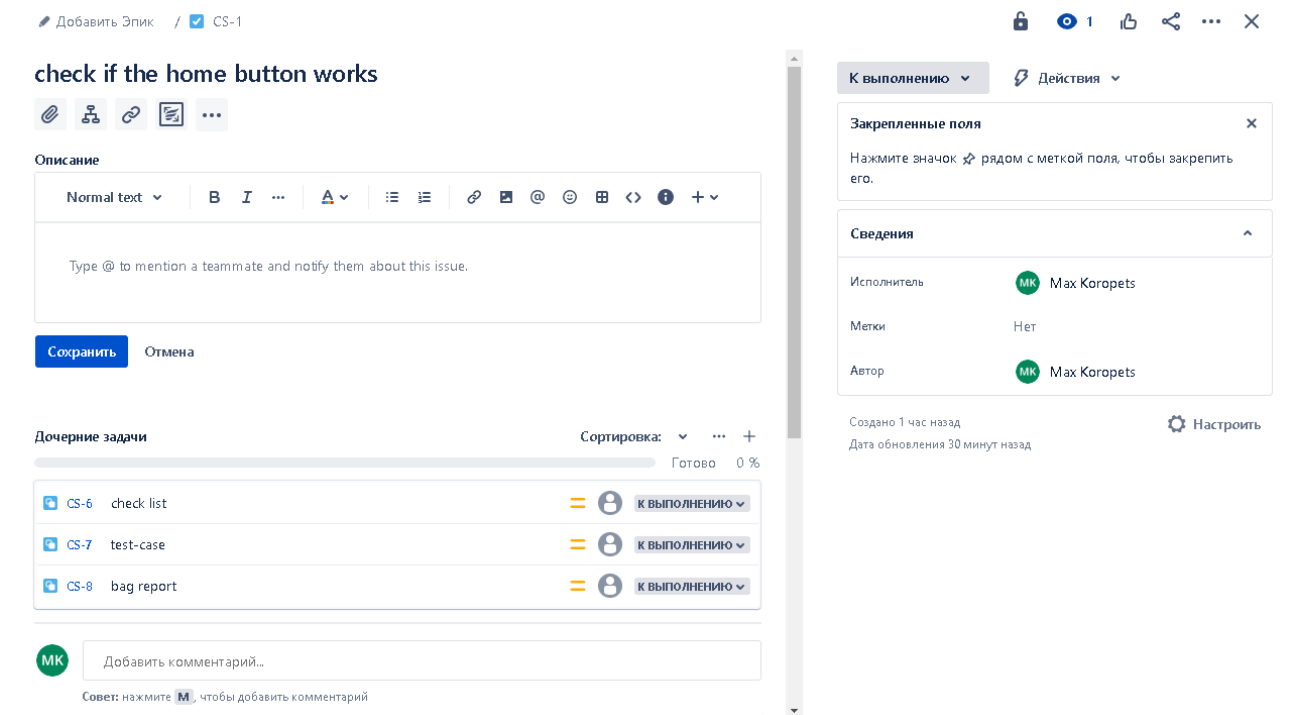


Рисунок 4.3 – Створення taskів

4.2 Створення чеклістів

Чеклісти в тестуванні використовуються для організації та систематизації процесу тестування програмного забезпечення. Вони допомагають тестувальникам перевірити, чи виконані всі необхідні тестові сценарії, перевірки

і критерії якості перед випуском продукту.

Існує багато альтернатив та програм для створення чеклістів ,але для новачків та тих користувачам яким зручний Microsoft Excel створюємо чек лісти тут.

TSHOP test										
Description										
№	Checking	Inputs	Expected result	Status	Author	Code & Link	Comments			
4.1	check if the home button works	press home button	button works		Max Koropets	https://sincoo.atlassian.net/jira/software/projects/CS/boards/1?selectedIssue=CS-1				
4.2	check the functionality of the delivery button	press delivery button	button works		Max Koropets	https://sincoo.atlassian.net/jira/software/projects/CS/boards/1?selectedIssue=CS-2				
4.3	check the functionality of the Register button	press register button	button works		Max Koropets	https://sincoo.atlassian.net/jira/software/projects/CS/boards/1?selectedIssue=CS-3				
4.4	check if the shopping cart button works	press shopping cart button	button works		Max Koropets	https://sincoo.atlassian.net/jira/software/projects/CS/boards/1?selectedIssue=CS-4				
4.5	check the operation of the Delivery button	press Delivery button	button works		Max Koropets	https://sincoo.atlassian.net/jira/software/projects/CS/boards/1?selectedIssue=CS-5				
4.6	check button functionality Description	press Description button	button works		Max Koropets	https://sincoo.atlassian.net/jira/software/projects/CS/boards/1?selectedIssue=CS-6				
4.7	check the functionality of the Contact Seller button	press Contact Seller button	button works		Max Koropets	https://sincoo.atlassian.net/jira/software/projects/CS/boards/1?selectedIssue=CS-7				

Рисунок 4.4 – Створення чеклістів

4.3 Створення тест кейсів

Zephyr - це комерційний інструмент для управління тестовою документацією та виконання тестових кейсів. Він надає можливість створювати, організувати, виконувати та відстежувати тестові кейси, що дозволяє забезпечити якість програмного забезпечення.[20]

За допомогою Zephyr ви можете:

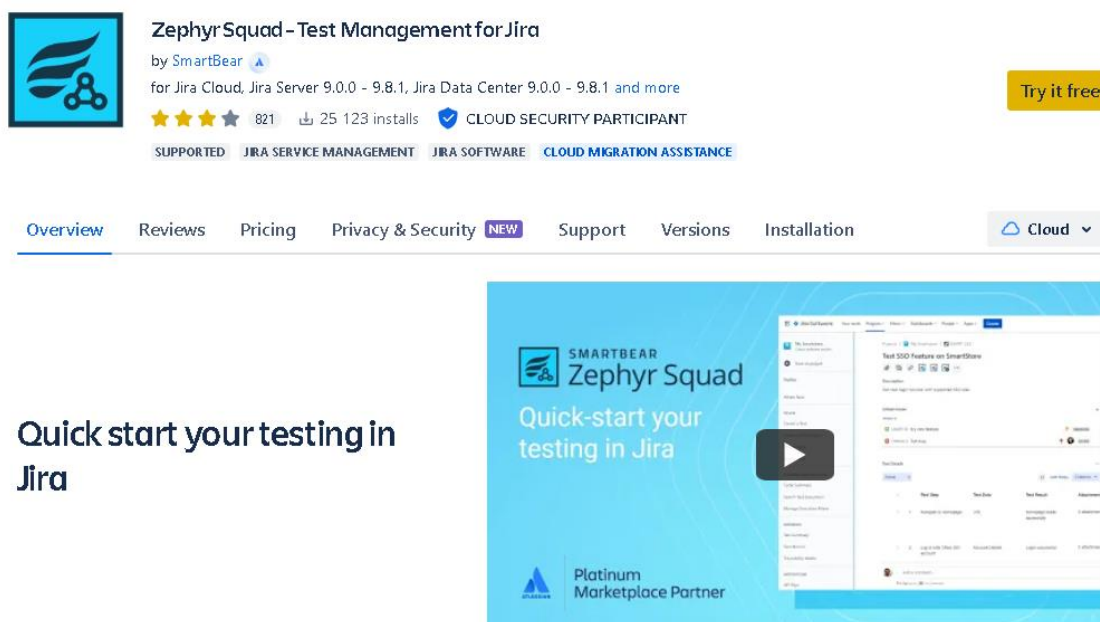


Рисунок 4.5 – Zephyr

Створювати тестові кейси: Ви можете створювати детальні тестові кейси, які описують кроки, очікувані результати, вхідні дані та передумови для кожного тестувального сценарію.

Організовувати тестову документацію: Zephyr дозволяє організовувати тестові кейси у логічні групи або набори, що спрощує навігацію та керування тестовими документами.

Виконувати тестові кейси: Zephyr надає інтерфейс для виконання тестових кейсів, в якому тестувальник може позначати кожний крок виконання, реєструвати фактичні результати та оцінювати пройдені тести.

Відстежувати прогрес тестування: Zephyr зберігає історію виконання тестових кейсів, дозволяючи відстежувати прогрес тестування, виявляти проблеми та забезпечувати повноту тестування.

Генерувати звіти: Інструмент надає можливість генерувати різноманітні звіти про виконані тести, прогрес тестування, статус дефектів та іншу важливу інформацію для стейкхолдерів проекту.

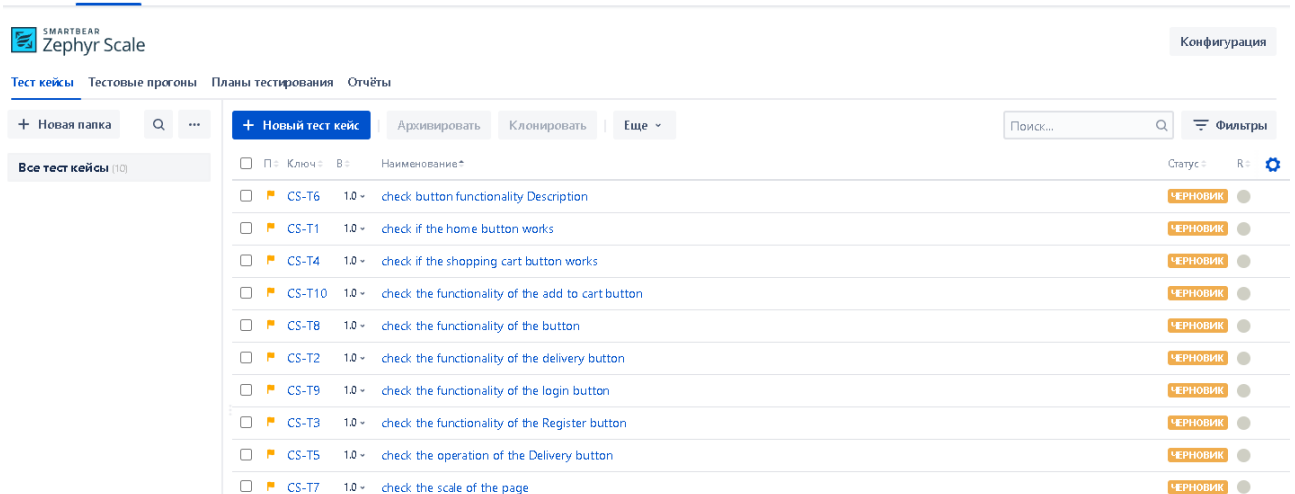


Рисунок 4.6 – Створення тест кейсів

4.4 Проходження тест кейсів

Проходимо всі тест кейси на сайті та виконуємо тести в Jira

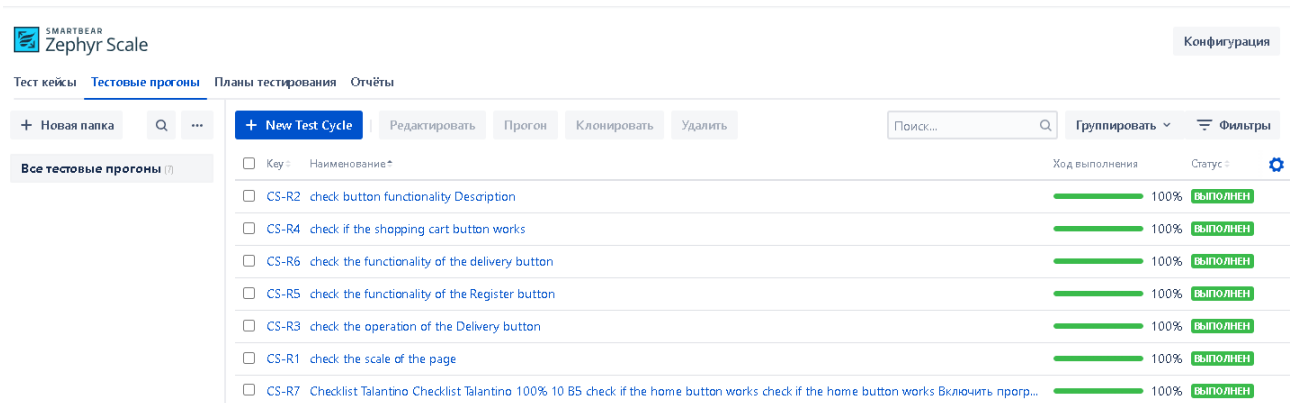


Рисунок 4.7 – Проходження тест кейсів

Створюємо тест репорт, виносимо всі імпрувменти та баги в документацію.

ВИСНОВКИ

В результаті виконання роботи було створено та протестовано інформаційну технологію електронної комерції.

Інформаційна технологія та тестування системи електронної комерції має важливе значення для забезпечення якості та надійності таких систем. Тестування допомагає виявити помилки, дефекти та функціональні проблеми, які можуть вплинути на роботу користувача та бізнес-результати.

Основні цілі тестування інформаційних технологій системи електронної комерції включають: Функціональне тестування:

1. Тестування системи електронної комерції дозволяє переконатися, що основні функції системи працюють належним чином. Це включає функції тестування для додавання товарів у кошик, оформлення замовлення, оформлення замовлення, доставки, керування запасами тощо.

2.Відповідність: Тестування допомагає переконатися, що система електронної комерції відповідає вимогам клієнта. Це може включати перевірку відповідності функціональним вимогам, стандартам безпеки, шаблонам проектування та правилам інтеграції з платіжними системами, поштовими службами тощо.

3.Тестування безпеки: системи електронної комерції мають багато конфіденційної інформації, такої як особисті дані користувачів і банківські реквізити. Тестування безпеки допомагає виявити можливі вразливості, зловживання та інші проблеми безпеки системи та розробляє заходи для їх усунення.

4. Тестування продуктивності дозволяє перевірити, як ваша система електронної комерції працює під високим навантаженням, забезпечуючи швидку реакцію та високу доступність. Це може включати тестування часу відповіді, завантаження серверів, обробку транзакцій і обробку великих обсягів даних. Перевірка сумісності: системи електронної комерції повинні бути сумісні з різними пристроями, операційними системами та браузерами, якими

користуються користувачі. Тестування на сумісність допомагає переконатися, що система працює в різних конфігураціях і забезпечує однакову взаємодію з користувачем незалежно від використовуваного пристрою та програмного забезпечення.

5. Тестування умов експлуатації: системи електронної комерції тестуються в різних умовах роботи, таких як високе навантаження, обмежена пропускну здатність мережі, перебої в мережі, зміна часового поясу тощо. Це допомагає виявити можливі проблеми та визначити, як система поводить себе в реальних умовах. Висновок. Тестування інформаційних технологій систем електронної комерції включає тестування функціональності, відповідності, безпеки, продуктивності, сумісності та умов роботи. Це допомагає забезпечити якість, надійність і ефективність систем електронної комерції, забезпечуючи задоволеність користувачів і успішну бізнес-операцію.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Коропець М.О. Інтернет магазин на базі фреймворку Django [Текст]:робота на здобуття кваліфікаційного рівня бакалавр; спец.:122-комп'ютерні науки М.О.Коропець; кер.І.Шеліхов – Суми : СумДУ,2020
2. Python-фреймворки: тенденції 2020 року [Електронний ресурс] – Режим доступу: waksoft.susu.ru/2020/01/14/top-5-frejmworkov-python-dlya-web-razrabotki-v-2020-godu
3. Огляд Django [Електронний ресурс] – Режим доступу: <https://djbook.ru/>
4. Документація Django [Електронний ресурс] – Режим доступу: <https://tproger.ru/translations/create-your-first-django-app>
5. Документація Django [Електронний ресурс] – Режим доступу: https://habr.com/ru/company/SECL_GROUP/blog/218921/
6. ТОП 10 кращих фреймворків для Python <https://django.fun/>
7. Документація Django [Електронний ресурс] – Режим доступу: <https://vuetifyjs.com/ru/components/api-explorer/>
8. Документація SQLite [Електронний ресурс] – Режим доступу: <https://proglib.io/p/sqlite-tutorial>
9. Інструментарій для розробки на Django [Електронний ресурс] – Режим доступу: <https://djbook.ru/>
10. Лутц, М. Програмування на Python. Т. 1 / М. Лутц. - М.: Символ, 2016. - 992 с
11. Лутц, М. Програмування на Python. Т. 2 / М. Лутц. - М.: Символ, 2016. - 876 с.
12. Метиз, Е. Вивчаємо PYTHON.Програмування програм, візуалізація даних, веб-додатки / Е. Метиз. - СПб.: Москва, 2017. - 496 с.
13. Володимир Дронов Django 2.1. Практика створення веб-сайтів на Python/В.Дронов-БХВ-Петербург,2016-672с
14. Pinkham A. Django Unleashed / Pinkham A. - Sams, 2012- 700с
15. Джефф форс, Пол Біссекс, Уеслі Чан. Django: Розробка веб-додатків на Python / Джефф форс, Пол Біссекс, Уеслі Чан. Django: Розробка веб-додатків на Python – Символ-плюс,2010 267с

16. Модульне тестування [Електронний ресурс] – Режим доступу до ресурсу: https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%A2%D0%B5%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F/page14.html
17. Тестування [Електронний ресурс] – Режим доступу до ресурсу: <https://qalight.ua/baza-znaniy/qa-qc-i-testuvannya/>
18. QA[Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ua/companies/otus/articles/588923/>
19. QC [Електронний ресурс] – Режим доступу до ресурсу: <https://sqa.lviv.ua/u-chomu-riznytsya-mizh-qa-ta-qc-testuvalnykom>
20. Керівництво по Python [Електронний ресурс] – Режим доступу: <https://python-scripts.com/>

ДОДАТОК

Файл admin.py:

```
from django.contrib import admin
from .models import *

class ProductInOrderInline(admin.TabularInline):
    model = ProductInOrder
    extra = 0

@admin.register(Status)
class StatusAdmin(admin.ModelAdmin):
    list_display = [field.name for field in Status._meta.fields]

@admin.register(Order)
class OrderAdmin(admin.ModelAdmin):
    list_display = [field.name for field in Order._meta.fields]
    inlines = [ProductInOrderInline]
```

Файл forms.py

```
from django import forms
from .models import *

class SubscriberForm(forms.ModelForm):
    class Meta:
        model = Subscriber
        exclude = []

class CheckoutContactForm(forms.Form):
    name = forms.CharField(required=True)
    phone = forms.CharField(required=True)
```

Файл models.py:

```
from django.db import models
from products.models import Product
from django.db.models.signals import post_save
from django.contrib.auth.models import User
```

```

class Status(models.Model):
    name = models.CharField(max_length=24, blank=True, null=True, default=None)
    is_active = models.BooleanField(default=True)
    created = models.DateTimeField(auto_now_add=True, auto_now=False)
    updated = models.DateTimeField(auto_now_add=False, auto_now=True)

    def str(self):
        return "Статус %s" % self.name

    class Meta:
        verbose_name = 'Статус заказа'
        verbose_name_plural = 'Статусы заказа'

class Order(models.Model):
    user = models.ForeignKey(User, blank=True, null=True, default=None,
on_delete=models.DO_NOTHING)
    total_price = models.DecimalField(max_digits=10, decimal_places=2, default=0)
    customer_name = models.CharField(max_length=64, blank=True, null=True,
default=None)
    customer_email = models.EmailField(blank=True, null=True, default=None)
    customer_phone = models.CharField(max_length=48, blank=True, null=True,
default=None)
    customer_address = models.CharField(max_length=128, blank=True, null=True,
default=None)
    comments = models.TextField(blank=True, null=True, default=None)
    status = models.ForeignKey(Status, on_delete=models.DO_NOTHING)
    created = models.DateTimeField(auto_now_add=True, auto_now=False)
    updated = models.DateTimeField(auto_now_add=False, auto_now=True)

    def str(self):
        return "Заказ %s %s" % (self.id, self.status.name)

    class Meta:
        verbose_name = 'Заказ'
        verbose_name_plural = 'Заказы'

    def save(self, *args, **kwargs):
        super().save(*args, **kwargs)

class ProductInOrder(models.Model):
    order = models.ForeignKey(Order, blank=True, null=True, default=None,

```

```

on_delete=models.DO_NOTHING)
    product = models.ForeignKey(Product, blank=True, null=True, default=None,
on_delete=models.DO_NOTHING)
    nmb = models.IntegerField(default=1)
    price_per_item = models.DecimalField(max_digits=10, decimal_places=2,
default=0)
    total_price = models.DecimalField(max_digits=10, decimal_places=2, default=0)
    is_active = models.BooleanField(default=True)
    created = models.DateTimeField(auto_now_add=True, auto_now=False)
    updated = models.DateTimeField(auto_now_add=False, auto_now=True)

    def str(self):
        return "%s" % self.product.name

    class Meta:
        verbose_name = 'Товар в заказе'
        verbose_name_plural = 'Товары в заказе'

    def save(self, *args, **kwargs):
        self.price_per_item = self.product.price
        self.total_price = int(self.nmb) * self.price_per_item
        super().save(*args, **kwargs)

def product_in_order_post_save(sender, instance, created, **kwargs):
    order = instance.order
    all_products_in_order = ProductInOrder.objects.filter(order=order, is_active=True)

    order_total_price = 0
    for item in all_products_in_order:
        order_total_price += item.total_price

    instance.order.total_price = order_total_price
    instance.order.save(force_update=True)

post_save.connect(product_in_order_post_save, sender=ProductInOrder)

```

Файл views.py:

```

from django.shortcuts import render
from .forms import SubscriberForm
from products.models import ProductImage

```

Max, [21.05.2023 02:21]

```
def landing(request):
    name = "CodingMedved"
    current_day = "03.01.2017"
    form = SubscriberForm(request.POST or None)

    if request.method == "POST" and form.is_valid():
        print(request.POST)
        print(form.cleaned_data)
        data = form.cleaned_data
        print(data["name"])

        new_form = form.save()

    return render(request, 'landing/landing.html', locals())
```

```
def home(request):
    products_images = ProductImage.objects.filter(is_active=True, is_main=True,
product__is_active=True)
    products_images_phones = products_images.filter(product__category__id=1)
    products_images_laptops = products_images.filter(product__category__id=2)
    return render(request, 'landing/home.html', locals())
```

Файл context_processors.py:

```
from .models import ProductInBasket
```

```
def getting_basket_info(request):
    session_key = request.session.session_key
    if not session_key:
        request.session["session_key"] = 123
        request.session.cycle_key()

    products_in_basket = ProductInBasket.objects.filter(session_key=session_key,
is_active=True, order__isnull=True)
    products_total_nmb = products_in_basket.count()

    return locals()
```

urls.py in the project's main folder:

```
from django.conf.urls import url, include
from django.contrib import admin
```

```
from landing import views
```

```
urlpatterns = [
    url(r'^$', views.home, name='home'),
    url(r'^landing123/$', views.landing, name='landing'),
    url(r'^admin/', admin.site.urls),
    url(r'^', include('landing.urls')),
]
```

urls.py in the "Orders" folder:

```
from django.conf.urls import url
from . import views
```

```
urlpatterns = [
    url(r'^basket_adding/$', views.basket_adding, name='basket_adding'),
    url(r'^checkout/$', views.checkout, name='checkout'),
]
```

Файл views.py :

```
from django.http import JsonResponse, HttpResponseRedirect
from .models import *
from django.shortcuts import render
from .forms import CheckoutContactForm
from django.contrib.auth.models import User
```

```
def basket_adding(request):
    return_dict = {}
    session_key = request.session.session_key
    print(request.POST)
    data = request.POST
    product_id = data.get("product_id")
    nmb = data.get("nmb")
    is_delete = data.get("is_delete")

    if is_delete == 'true':
        ProductInBasket.objects.filter(id=product_id).update(is_active=False)
    else:
        new_product, created =
        ProductInBasket.objects.get_or_create(session_key=session_key,
        product_id=product_id,
        is_active=True, defaults={"nmb": nmb})

        if not created:
```

```

    print("not created")
    new_product.nmb += int(nmb)
    new_product.save(force_update=True)

# Common code for both cases
products_in_basket = ProductInBasket.objects.filter(session_key=session_key,
is_active=True, order__isnull=True)
products_total_nmb = products_in_basket.count()
return_dict["products_total_nmb"] = products_total_nmb

return_dict["products"] = []

for item in products_in_basket:
    product_dict = {}
    product_dict["id"] = item.id
    product_dict["name"] = item.product.name
    product_dict["price_per_item"] = item.price_per_item
    product_dict["nmb"] = item.nmb
    return_dict["products"].append(product_dict)

return JsonResponse(return_dict)

def checkout(request):
    session_key = request.session.session_key
    products_in_basket = ProductInBasket.objects.filter(session_key=session_key,
is_active=True, order__isnull=True)
    print(products_in_basket)
    for item in products_in_basket:
        print(item.order)

form = CheckoutContactForm(request.POST or None)
if request.POST:
    print(request.POST)
    if form.is_valid():
        print("yes")
        data = request.POST
        name = data.get("name", "3423453")
        phone = data["phone"]
        user, created = User.objects.get_or_create(username=phone,
defaults={"first_name": name})

Max, [21.05.2023 02:21]
order = Order.objects.create(user=user, customer_name=name,

```



```

customer_phone=phone, status_id=1)

    for name, value in data.items():
        if name.startswith("product_in_basket_"):
            product_in_basket_id = name.split("product_in_basket_")[1]
            product_in_basket =
ProductInBasket.objects.get(id=product_in_basket_id)
            print(type(value))

            product_in_basket.nmb = value
            product_in_basket.order = order
            product_in_basket.save(force_update=True)

            ProductInOrder.objects.create(product=product_in_basket.product,
nmb=product_in_basket.nmb,
                                         price_per_item=product_in_basket.price_per_item,
                                         total_price=product_in_basket.total_price,
                                         order=order)

        return HttpResponseRedirect(request.META['HTTP_REFERER'])
    else:
        print("no")
    return render(request, 'orders/checkout.html', locals())

```

Папка test_project
Файл settings.py

```

import os

BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(file)))

SECRET_KEY = 'your-secret-key'

DEBUG = True

ALLOWED_HOSTS = []

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',

```

```
'landing',
'products',
'orders',
]

MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'test_project.urls'

TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [os.path.join(BASE_DIR, 'templates')],
'APP_DIRS': True,
'OPTIONS': {
'context_processors': [
'django.template.context_processors.debug',
'django.template.context_processors.request',
'django.contrib.auth.context_processors.auth',
'django.contrib.messages.context_processors.messages',
'orders.context_processors.getting_basket_info',
],
},
},
]

WSGI_APPLICATION = 'test_project.wsgi.application'

DATABASES = {
'default': {
'ENGINE': 'django.db.backends.sqlite3',
'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
}
}

AUTH_PASSWORD_VALIDATORS = [
```

```

    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

```

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

STATIC_URL = '/static/'

```

STATICFILES_DIRS = (
    os.path.join(BASE_DIR, "static", "static_dev"),
)

```

STATIC_ROOT = os.path.join(BASE_DIR, "static", "static_prod")

MEDIA_URL = '/media/'

MEDIA_ROOT = os.path.join(BASE_DIR, "static", "media")

Файл urls.py

```

from django.urls import path, include
from django.contrib import admin
from django.conf import settings
from django.conf.urls.static import static

```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", include('landing.urls')),
    path("", include('products.urls')),
    path("", include('orders.urls')),
] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT) +
static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Файл manage.py:

```
import os
import sys
```

Max, [21.05.2023 02:21]

```
if name == "main":
    os.environ.setdefault("DJANGO_SETTINGS_MODULE", "test_project.settings")
    try:
        from django.core.management import execute_from_command_line
    except ImportError:
        try:
            import django
        except ImportError:
            raise ImportError(
                "Couldn't import Django. Are you sure it's installed and available on your PYTHONPATH environment variable? Did you forget to activate a virtual environment?"
            )
        raise
    execute_from_command_line(sys.argv)
```

Файл home.html:

```
{% extends 'base.html' %}
{% load static %}

{% block content %}
<div class="section-top">
    
</div>
<div class="section section1">
    <div class="container">
        <div class="row">
            <div class="col-lg-12">
                <h1 class="text-center">
```

```

        НОВЫЕ ПОСТУПЛЕНИЯ
    </h1>
</div>
    {% for product_image in products_images %}
        {% include 'landing/product_item.html' %}
    {% endfor %}
</div>
</div>
</div>

<div class="section section2 phones-container">
    <div class="container">
        <div class="row">
            <div class="col-lg-12">
                <h1 class="text-center">
                    Новая колекция
                </h1>
            </div>
            {% for product_image in products_images_phones %}
                {% include 'landing/product_item.html' %}
            {% endfor %}
        </div>
    </div>
</div>

<div class="section section3 laptops-container">
    <div class="container">
        <div class="row">
            <div class="col-lg-12">
                <h1 class="text-center">
                    Пользуются популярностью
                </h1>
            </div>
            {% for product_image in products_images_laptops %}
                {% include 'landing/product_item.html' %}
            {% endfor %}
        </div>
    </div>
</div>
{% endblock %}

```

Файл product-item.html

```

<div class="col-lg-3">
  <div class="product-item">
    <div>
      <a href="{% url 'product' product_image.product.id %}">
        
      </a>
    </div>

    {% if product_image.product.discount %}
      <div class="discount-container">
        <span>{{ product_image.product.discount }}%</span>
      </div>
    {% endif %}

    <h4>{{ product_image.product.name }}</h4>
    <p>
      {{ product_image.product.description|truncatechars_html:70 }}
    </p>
    <div>
      {{ product_image.product.price }} UAH
    </div>
    <div class="add-to-card-btn">
      <button class="btn btn-success">
        Добавить в корзину
      </button>
    </div>
  </div>
</div>

<![endif]-->

<!-- Favicon and touch icons -->
<link rel="shortcut icon" href="assets/ico/favicon.png">
<link rel="apple-touch-icon-precomposed" sizes="144x144"
href="assets/ico/apple-touch-icon-144-precomposed.png">
<link rel="apple-touch-icon-precomposed" sizes="114x114"
href="assets/ico/apple-touch-icon-114-precomposed.png">
<link rel="apple-touch-icon-precomposed" sizes="72x72"
href="assets/ico/apple-touch-icon-72-precomposed.png">
<link rel="apple-touch-icon-precomposed" href="assets/ico/apple-touch-icon-
57-precomposed.png">

</head>

```

```

<body>

  <!-- Loader -->
  <div class="loader">
    <div class="loader-img"></div>
  </div>

  <!-- Top content -->
  <div class="top-content">

    <!-- Top menu -->
    <nav class="navbar navbar-inverse navbar-no-bg" role="navigation">
      <div class="container">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#top-navbar-1">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
          <a class="navbar-brand" href="index.html">Marco - Bootstrap Landing
Page</a>
        </div>
        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse" id="top-navbar-1">
          <ul class="nav navbar-nav navbar-right">
            <li><a class="scroll-link" href="#features">Features</a></li>
            <li><a class="scroll-link" href="#testimonials">Testimonials</a></li>
            <li><a class="scroll-link" href="#pricing">Price</a></li>
            <li><a class="scroll-link" href="#how-it-works">How it
works</a></li>
            <li><a class="scroll-link" href="#about-us">About</a></li>
            <li><a class="scroll-link" href="#contact">Contact</a></li>
          </ul>
        </div>
      </div>
    </nav>

    <div class="inner-bg">
      <div class="container">

        <div class="row">

```

```

<div class="col-sm-8 col-sm-offset-2 text">
  <h1 class="wow fadeInLeftBig">Begin Using Our Product
Today</h1>
  <div class="description wow fadeInLeftBig">
    <p>
      Subscribe by entering your email address in the form below and
we'll send you a link to start right away.
      Working MailChimp form!
    </p>
  </div>
  <div class="subscribe wow fadeInUp">
    <form class="form-inline" role="form"
action="assets/subscribe.php" method="post">
      <div class="form-group">
        <label class="sr-only" for="subscribe-email">Email
address</label>
        <input type="text" name="email" placeholder="Enter your
email..." class="subscribe-email form-control" id="subscribe-email">
      </div>
      <button type="submit" class="btn">Subscribe</button>
    </form>
    <div class="success-message"></div>
  </div>
</div>

<div class="row">
  <div class="col-sm-8 col-sm-offset-2 top-video-link medium-paragraph
wow fadeInUp">
    <a href="#" class="launch-modal" data-modal-id="modal-how-it-
works">
      <span class="top-video-link-icon"><i class="fa fa-
play"></i></span>
      <span class="top-video-link-text">See how it works</span>
    </a>
  </div>
</div>

<div class="row">
  <div class="col-sm-12 top-social wow fadeInUp">
    <a href="#"><i class="fa fa-facebook"></i></a> <span
class="divider-2"></span>
    <a href="#"><i class="fa fa-twitter"></i></a> <span class="divider-
2"></span>

```



```

        <a href="#"><i class="fa fa-google-plus"></i></a> <span
class="divider-2"></span>
        <a href="#"><i class="fa fa-instagram"></i></a> <span class="divider-
2"></span>
        <a href="#"><i class="fa fa-pinterest"></i></a>
        </div>
    </div>

</div>
</div>

</div>

<!-- Features -->
<div class="features-container section-container">
    <div class="container">
        <div class="row">
            <div class="col-sm-12 features section-description wow fadeIn">
                <h2>Features</h2>
                <div class="divider-1 wow fadeInUp"><span></span></div>
                <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et.
                Ut wisi enim ad minim veniam, quis nostrud tempor incididunt ut
labore et.</p>
            </div>
        </div>
        <div class="row">
            <div class="col-sm-4 features-box wow fadeInUp">
                <div class="features-box-icon"><i class="fa fa-eye"></i></div>
                <h3>Easy To Use</h3>
                <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et.</p>
            </div>
            <div class="col-sm-4 features-box wow fadeInDown">
                <div class="features-box-icon"><i class="fa fa-thumbs-up"></i></div>
                <h3>Responsive Design</h3>
                <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et.</p>
            </div>
            <div class="col-sm-4 features-box wow fadeInUp">
                <div class="features-box-icon"><i class="fa fa-cog"></i></div>
                <h3>Bootstrap Engine</h3>
                <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et.</p>
            </div>
        </div>
    </div>
</div>

```

```

    </div>
  </div>
  <div class="row">
    <div class="col-sm-4 features-box wow fadeInUp">
      <div class="features-box-icon"><i class="fa fa-play"></i></div>
      <h3>Lots Of Videos</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et.</p>
    </div>
    <div class="col-sm-4 features-box wow fadeInDown">
      <div class="features-box-icon"><i class="fa fa-phone"></i></div>
      <h3>Mobile App</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et.</p>
    </div>
    <div class="col-sm-4 features-box wow fadeInUp">
      <div class="features-box-icon"><i class="fa fa-
commenting"></i></div>
      <h3>Big Community</h3>
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et.</p>
    </div>
  </div>
  <div class="row">
    <div class="col-sm-12 section-bottom-button wow fadeInUp">
      <a class="btn btn-link-1 scroll-link" href="#pricing">Plans &
Pricing</a>
    </div>
  </div>
</div>
</div>
<!-- More features 2 -->
<div class="more-features-2-container section-container section-container-gray-
bg">
  <div class="container">
    <div class="row">
      <div class="col-sm-7 more-features-2-box wow fadeInLeft">
        
      </div>
      <div class="col-sm-5 more-features-2-box more-features-2-box-right
wow fadeInUp">
        <h3>We have more features for you:</h3>
        <ul>

```

- <i class="fa fa-check-circle"></i> 15+ Layouts
- <i class="fa fa-check-circle"></i> 30+ Forms (registration, login, contact & subscription forms)
- <i class="fa fa-check-circle"></i> Responsive Layout
- <i class="fa fa-check-circle"></i> Bootstrap Framework
- <i class="fa fa-check-circle"></i> MailChimp Subscription

Form

See Pricing

</div>

</div>

</div>

</div>

<!-- Many options -->

<div class="many-options-container section-container">

<div class="container">

<div class="row">

<div class="col-sm-5 many-options-box wow fadeInLeft">

<h3>Many options to choose from:</h3>

<p>

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et.

Ut wisi enim ad minim veniam, quis nostrud.

</p>

<p>

Exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl.

</p>

<p>

Learn More

</p>

</div>

<div class="col-sm-7 many-options-box many-options-box-right wow fadeInUp">

</div>

</div>

</div>

</div>

<!-- Testimonials -->

```

<div class="testimonials-container section-container section-container-image-
bg">
  <div class="container">
    <div class="row">
      <div class="col-sm-12 testimonials section-description wow fadeIn">
        <h2>Testimonials</h2>
        <div class="divider-1 wow fadeInUp"><span></span></div>
        <p>Take a look below to read what our clients have said about us:</p>
      </div>
    </div>
    <div class="row">
      <div class="col-sm-10 col-sm-offset-1 testimonial-list wow fadeInUp">
        <div role="tabpanel">
          <!-- Tab panes -->
          <div class="tab-content">
            <div role="tabpanel" class="tab-pane fade in active" id="tab1">
              <div class="testimonial-image">
                
              </div>
              <div class="testimonial-text">
                <h3>5 / 5 Stars</h3>
                <p>
                  "Lorem ipsum dolor sit amet, consectetur adipisicing elit,
sed do eiusmod tempor incididunt ut labore et.
                  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
do eiusmod tempor incididunt ut labore et.
                  Lorem ipsum dolor sit amet, consectetur..."<br>
                  <a href="#">Lorem Ipsum, dolor.co.uk</a>
                </p>
              </div>
            </div>
            <div role="tabpanel" class="tab-pane fade" id="tab2">
              <div class="testimonial-image">
                
              </div>
              <div class="testimonial-text">
                <h3>4 / 5 Stars</h3>
                <p>
                  "Ut wisi enim ad minim veniam, quis nostrud exerci tation
ullamcorper suscipit lobortis nisl ut aliquip
                  ex ea commodo consequat. Ut wisi enim ad minim veniam,
quis nostrud exerci tation ullamcorper suscipit

```

```

        lobortis nisl ut aliquip ex ea commodo consequat..."<br>
        <a href="#">Minim Veniam, nostrud.com</a>
    </p>
</div>
</div>
<div role="tabpanel" class="tab-pane fade" id="tab3">
    <div class="testimonial-image">
        
    </div>
    <div class="testimonial-text">
        <h3>5 / 5 Stars</h3>

</body>

</html>

```

Папка static

Файл landing.css

```

html, body{
    height: 100%;
    max-height: 100%;
    font-family: 'Roboto', sans-serif;
}

.top-container{
    padding-top: 10%;
}

.general-container{
    background: url('./img/bg_image.jpg');
    height: 100%;
}

.general-container form{
    margin-top: 30px;
}

.general-container form label{
    font-weight: 100;
    color: white;
}

.btn-orange{

```

```
background-color: #e8643e;
border: none;
}
```

```
.btn-orange:hover{
background-color: #dc5129;
border: none;
}
```

```
.title{
text-align: center;
color: white;
font-weight: 100;
line-height: 50px;
}
```

Файл style.css

```
height: 100%;
max-height: 100%;
font-family: 'Roboto', sans-serif;
background-color: #edeef0;
}
```

```
.navbar{
margin-bottom: 0;
}
```

```
.wrapper{
min-height: 100%;
}
```

```
.wrapper-content{
overflow: auto;
padding-bottom: 180px; /* must be same height as the footer */
}
```

```
.footer{
position: relative;
margin-top: -180px; /* negative value of footer height */
height: 180px;
clear: both;
background-color: white;
}
```

```
.section-top{
  margin-bottom: 20px
}

.product-description-tabs{
  padding: 10px;
}

.section{
  padding: 50px 0;
}

.section h1{
  margin-bottom: 40px;
}

.product-item{
  height: 460px;
  background-color: white;
  border: 1px solid lightgrey;
  position: relative;
  padding: 10px 0 10px 0;
  text-align: center;
  margin-bottom: 10px;
}

.add-to-card-btn{
  position: absolute;
  bottom: 15px;
  left: 50%;
  transform: translateX(-50%);
}

.discount-container{
  position: absolute;
  top: 30%;
  background: #55e073;
  width: 30%;
  color: white;
  font-weight: 700;
  padding: 3px;
}
```

```
.section-delivery{  
  height: 300px;  
  background-color: darkseagreen;  
  text-align: center;  
  color: white;  
}
```

```
.product-image-item{  
  padding: 5px;  
  margin-bottom: 5px;  
}
```

```
.navbar-top{  
  min-height: 10px;  
  height: 20px;  
  background-color: green;  
  border: none;  
  border-radius: 0;  
}
```

```
.navbar-main{  
  background-color: #55e073;  
  border: none;  
  border-radius: 0;  
}
```

```
.basket-container{  
  position: relative;  
  width: 400px;  
  padding: 15px 10px;  
}
```

```
.basket-items{  
  position: absolute;  
  top: 50px;  
  width: 100%;  
  background-color: #55e093;  
  z-index: 10;  
  padding: 10px;  
}
```

```
.form-error{  
  color: red;  
}
```



```
html, body{
  height: 100%;
  max-height: 100%;
  font-family: 'Roboto', sans-serif;
  background-color: #edeef0;
}

.navbar{
  margin-bottom: 0;
}

.wrapper{
  min-height: 100%;
}

.wrapper-content{
  overflow: auto;
  padding-bottom: 180px; /* must be same height as the footer */
}

.footer{
  position: relative;
  margin-top: -180px; /* negative value of footer height */
  height: 180px;
  clear: both;
  background-color: white;
}

.section-top{
  margin-bottom: 20px
}

.product-description-tabs{
  padding: 10px;
}

.section{
  padding: 50px 0;
}

.section h1{
  margin-bottom: 40px;
}
```

```
.product-item{
  height: 460px;
  background-color: white;
  border: 1px solid lightgrey;
  position: relative;
  padding: 10px 0 10px 0;
  text-align: center;
  margin-bottom: 10px;
}

.add-to-card-btn{
  position: absolute;
  bottom: 15px;
  left: 50%;
  transform: translateX(-50%);
}

.discount-container{
  position: absolute;
  top: 30%;
  background: #55e073;
  width: 30%;
  color: white;
  font-weight: 700;
  padding: 3px;
}

.section-delivery{
  height: 300px;
  background-color: darkseagreen;
  text-align: center;
  color: white;
}

.product-image-item{
  padding: 5px;
  margin-bottom: 5px;
}

.navbar-top{
  min-height: 10px;
  height: 20px;
  background-color: green;
}
```

```

border: none;
border-radius: 0;
}

.navbar-main{
background-color: #55e073;
border: none;
border-radius: 0;
}

```

```

.basket-container{
position: relative;
width: 400px;
padding: 15px 10px;
}

```

```

.basket-items{
position: absolute;
top: 50px;
width: 100%;
background-color: #55e093;
z-index: 10;
padding: 10px;
}

```

```

.form-error{
color: red;
}

```

Папка product

Файл navbar.html

```
<nav class="navbar navbar-default navbar-top">
```

```
</nav>
```

```
<nav class="navbar navbar-default navbar-main">
```

```
<div class="container">
```

```
<div class="navbar-header">
```

```
<button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
data-target="#navbar" aria-expanded="false" aria-controls="navbar">
```

```
<span class="sr-only">Toggle navigation</span>
```

```
<span class="icon-bar"></span>
```

```
<span class="icon-bar"></span>
```

```
<span class="icon-bar"></span>
```

```
</button>
```

```
<a class="navbar-brand" href="#">ROCK SHOP</a>
```

```

</div>
<div id="navbar" class="navbar-collapse collapse">
  <ul class="nav navbar-nav">
    <li class="active"><a href="/">Домой</a></li>
    <li><a href="#about">Доставка</a></li>
    <li><a href="#contact">Контакты</a></li>
    <li class="dropdown">
      <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">Регистрация <span
class="caret"></span></a>
      <ul class="dropdown-menu">
        <li><a href="#">Войти</a></li>
        <li><a href="#">Регистрация</a></li>
        <li><a href="#">Забыл пароль</a></li>
        <li role="separator" class="divider"></li>
        <li class="dropdown-header">Дополнительно</li>
        <li><a href="#">Восстановить номер</a></li>
        <li><a href="#">Восстановить Email</a></li>
      </ul>
    </li>
  </ul>
  <ul class="nav navbar-nav navbar-right">
    <li>
      <div class="basket-container">
        <a href="">Корзина <span id="basket_total_amount"></span><span
id="basket_total_nmb">
          {% if products_total_nmb %}
            ({{ products_total_nmb }})
          {% else %}
            (0)
          {% endif %}
        </span></a>
        <div class="basket-items hidden">
          <ul class="">
            {% for product_in_basket in products_in_basket %}
              <li>
                {{ product_in_basket.product.name }} {{
product_in_basket.nmb }} шт.
                по {{ product_in_basket.price_per_item }} грн.
                <a class="delete-item" href="" data-product_id="{{
product_in_basket.id }}">x</a>
              </li>
            {% endfor %}
          </ul>
        </div>
      </div>
    </li>
  </ul>

```

```

        <a href="{% url 'checkout' %}">
            Оформить заказ
        </a>
    </li>
</ul>
</div>
</div>
</li>

</ul>
</div><!--/.nav-collapse -->
</div>
</nav>

```

Файл base.html

```

{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>

    <!-- Bootstrap-->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/3.3.6/css/bootstrap.min.css">

    <link
href="https://fonts.googleapis.com/css?family=Roboto:100,100i,300,300i,400"
rel="stylesheet">

    <!-- Custom styles -->
    <link rel="stylesheet" href="{% static 'css/style.css' %}">

</head>
<body>

    <div class="wrapper">
        <div class="wrapper-content">
            {% include 'navbar.html' %}

```

```
    {% block content %}  
    {% endblock content %}  
</div>
```

```
</div>
```

```
{% include 'footer.html' %}
```

```
<!-- jQuery -->  
<script src="https://code.jquery.com/jquery-2.1.4.min.js"></script>  
<!-- Latest compiled and minified JavaScript -->  
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"  
integrity="sha384-  
0mSbJDEHialfmuBBQP6A4Qrprq5OVfW37PRR3j5ELqxss1yVqOtnepnHVP9aJ7x  
S" crossorigin="anonymous"></script>
```

```
<script src="{% static 'js/scripts.js' %}"></script>
```

```
</body>
```

```
</html>
```