**Веб-приложение для продажи товаров**

Описание

Готовое приложение будет представлять собой платформу для продажи товаров. У пользователей будет доступен следующий функционал: просмотр товаров по категориям, просмотр детальной информации о товаре, регистрация, авторизация, добавление товаров в корзину, оформление заказа, просмотр списка оформленных заказов, написание отзывов. Администратор будет иметь доступ к админ-панели и сможет создавать и редактировать товары. Также у администратора будет возможность просматривать и редактировать заказы всех пользователей.

Наименование

Web Retail Platform

Данные

В приложении будут использоваться следующие модели данных:

**Category** – модель, используемая для хранения данных о категориях товаров и их сокращенных наименований для url.

class **Category**(models.Model):

name = models.CharField(max\_length=200, db\_index=True)

slug = models.SlugField(max\_length=200, db\_index=True, unique=True)

**SubCategory** – модель, используемая для хранения данных о подкатегориях товаров и их сокращенных наименований для url.

class **SubCategory**(models.Model):

name = models.CharField(max\_length=200, db\_index=True)

slug = models.SlugField(max\_length=200, db\_index=True, unique=True)

category\_id = models.ForeignKey(Category, on\_delete=models.CASCADE)

**Manufacturer** – модель, используемая для хранения данных о производителях товаров.

class **Manufacturer**(models.Model):

name = models.CharField(max\_length=45, help\_text="Enter name of manufacturer")

description = models.TextField(help\_text="Enter description of manufacturer")

email = models.EmailField(help\_text="Enter email of user", blank=True)

website = models.CharField(max\_length=128, help\_text="It is address of website", blank=True)

**Product** – модель, используемая для хранения данных о товарах.

class **Product**(models.Model):

sub\_category\_id = models.ForeignKey(SubCategory, on\_delete=models.CASCADE)

manufacturer\_id = models.ForeignKey(Manufacturer, on\_delete=models.CASCADE)

name = models.CharField(max\_length=200, db\_index=True)

slug = models.SlugField(max\_length=200, db\_index=True)

main\_image = models.ImageField(upload\_to='products/%Y/%m/%d', blank=True)

detail\_image = models.ImageField(upload\_to='products/%Y/%m/%d', blank=True)

description = models.TextField(blank=True)

price = models.DecimalField(max\_digits=10, decimal\_places=2)

stock = models.PositiveIntegerField()

available = models.BooleanField(default=True)

created = models.DateTimeField(auto\_now\_add=True)

updated = models.DateTimeField(auto\_now=True)

**Order** – модель, используемая для хранения данных о заказах и реквизитов покупателя.

class **Order**(models.Model):

first\_name = models.CharField(max\_length=50)

last\_name = models.CharField(max\_length=50)

email = models.EmailField()

address = models.CharField(max\_length=250)

postal\_code = models.CharField(max\_length=20)

city = models.CharField(max\_length=100)

created = models.DateTimeField(auto\_now\_add=True)

updated = models.DateTimeField(auto\_now=True)

paid = models.BooleanField(default=False)

**OrderItem** – модель, используемая для связи заказа и отдельных товаров заказа (их количества, цены покупки).

class **OrderItem**(models.Model):

order = models.ForeignKey(Order, related\_name='items', on\_delete=models.CASCADE)

product = models.ForeignKey(Product, related\_name='order\_items', on\_delete=models.CASCADE)

price = models.DecimalField(max\_digits=10, decimal\_places=2)

quantity = models.PositiveIntegerField(default=1)

**User** – модель, используемая для хранения данных о пользователях.

class **User**(models.Model):

password = models.CharField(max\_length=250)

last\_login = models.DateTimeField(auto\_now=True)

is\_superuser = models.BooleanField(default=False)

user\_name = models.CharField(max\_length=50)

first\_name = models.CharField(max\_length=50)

email = models.EmailField()

is\_staff = models.BooleanField(default=False)

is\_active = models.BooleanField(default=False)

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

**Review** – модель, используемая для хранения данных об отзывах.

class **Review**(models.Model):

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

text = models.CharField(max\_length=1000)

grade = models.DecimalField(max\_digits=1, decimal\_places=0)

product = models.ForeignKey(Product, on\_delete=models.CASCADE)

Пользовательские роли

**Гость** (неавторизованный пользователь)

Возможности:

* Просматривать информацию о товарах
* Зарегистрироваться
* Авторизоваться

**Авторизованный пользователь**

Авторизованный пользователь может все, что и гость +

* Добавлять и убирать товары из корзины
* Просматривать список товаров в корзине
* Оформлять заказы
* Просматривать список своих заказов
* Выходить из аккаунта
* Оставлять отзывы о товарах

**Администратор**

Администратор может все, что и авторизованный пользователь +

* Добавлять новые товары на платформу и редактировать информацию о существующих товарах.
* Добавлять информацию о новых производителях и редактировать существующих
* Создавать новые категории товаров и редактировать текущие
* Просматривать информацию об оформленных заказах всех пользователей и редактировать при необходимости
* Удалять отзывы пользователей при необходимости

UI

**Стартовая страница**

На стартовой странице содержится общая информация о платформе. В верхней части страницы расположены категории товаров, по которым можно перейти, а также иконки корзины, списка заказов, профиля для авторизованных пользователей и иконка для авторизации для неавторизованных пользователей.

**Страница категории товаров**

На странице определенной категории товаров расположен список всех товаров данной категории с миниатюрными изображениями, ценой и названиями. При нажатии на определенный товар осуществляется переход на страницу товара.

**Страница товара**

На странице товара содержится описание, большое изображение, наименование, цена, меню выбора количества и кнопка добавления в корзину. Также имеется возможность посмотреть отзывы и оставить свой.

**Страница корзины**

На странице корзины содержится список товаров, добавленных в корзину. Имеется возможность удалять товары из корзины. Присутствует кнопка перехода к оформлению заказа.

**Страница оформления заказа**

На странице оформления заказа имеются поля для реквизитов покупателя и кнопка оформления заказа.

**Страница заказов**

На странице заказов содержится информация о всех заказах, оформленных пользователем, и их статусы.

**Страница аккаунта пользователя**

На странице аккаунта пользователя имеется кнопка для выхода из аккаунта.

**Страница авторизации**

На странице авторизации имеются поля для логина и пароля пользователя и кнопка для входа в аккаунт.

**Страница регистрации**

На странице регистрации содержатся поля для ввода необходимой информации о пользователе и кнопка зарегистрироваться.

Языки программирования и технологии

Python, Django, Docker.

СУБД

Postgresql.