Kazakhstan Market Organization(K.M.O.)

Автор: Ким Анастасия

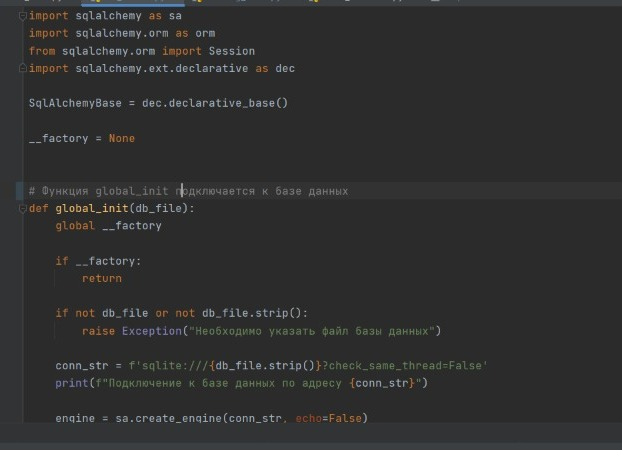
Описание идеи

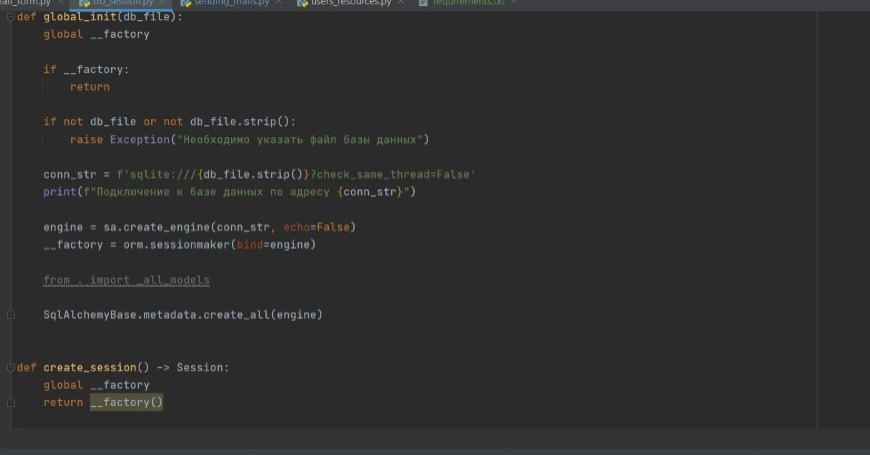
Сайт предназначен для выставки объявлений о продаже и покупке какого-либо товара. На нем покупатели могут добавлять в корзину или получать необходимый им контакт продавца, чтобы купить товар. А продавцы могут публиковать объявление, ставя цену и описание товара.

Описание реализации

Из классов были созданы ORM модели. User (класс для пользователей, атрибутами которого являются имя, фамилия, возраст, почта и контакты) и Goods (класс для товаров, атрибутами которого являются название товара, цена, описание, контакты продавца и т. д.). Также при создании API с помощью flask\_restful я создала классы GoodsResource(для одного объекта), UserResource(тоже для одного объекта), UsersListResource(для нескольких объектов), GoodsListResource(тоже для нескольких объектов)

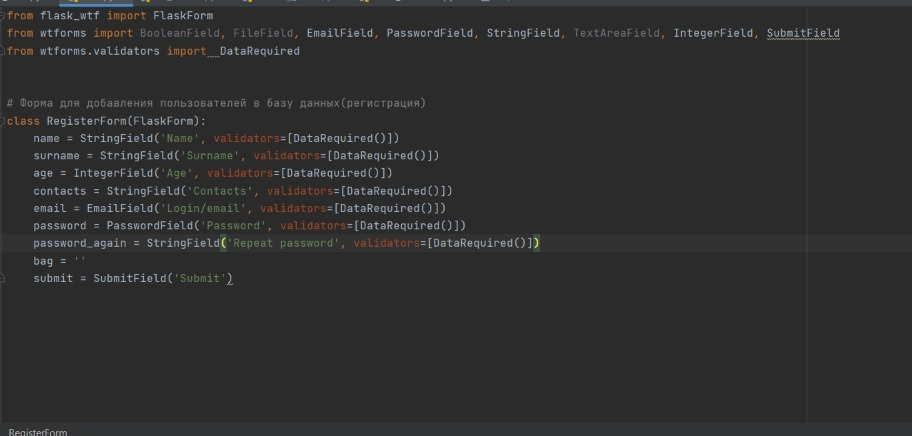
В файле db\_session устанавливается подключение к базе данных через sqlalchemy.



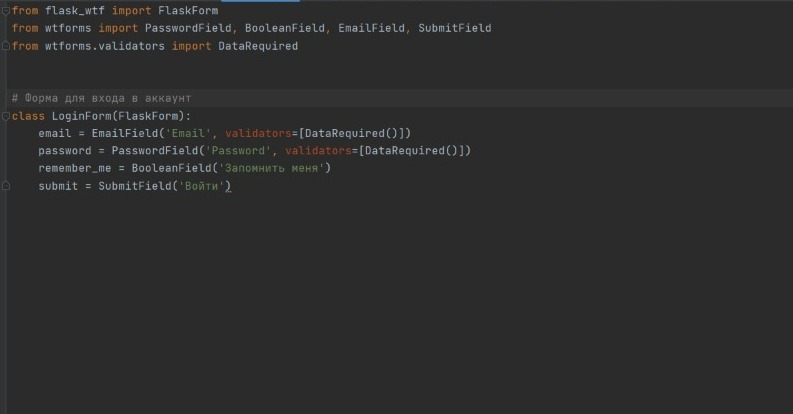


А также я использовала формы для запроса и получения данных от пользователя.

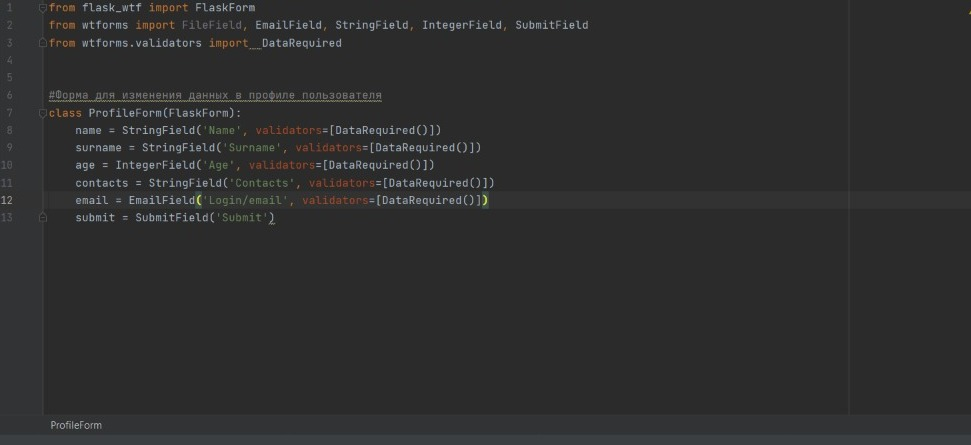
RegisterForm – форма для регистрации. Она запрашивает имя, фамилию, возраст, номер телефона, почту, пароль дважды и корзина(пустая сначала)



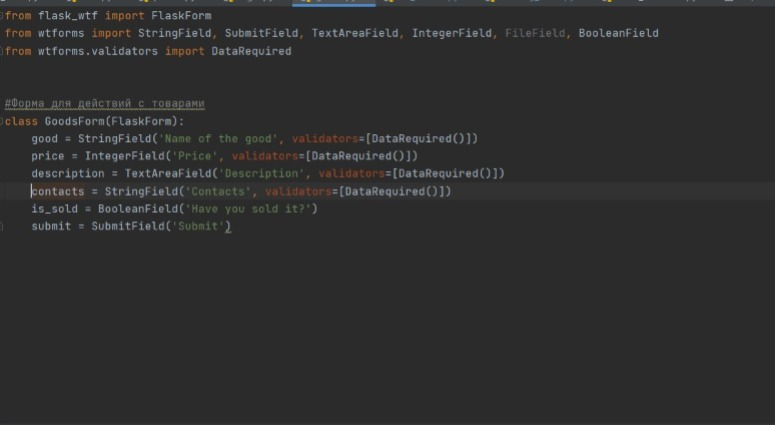
LoginForm – форма для входа в аккаунт. Запрашивает почту, пароль и булевое значение «Запомнить меня»



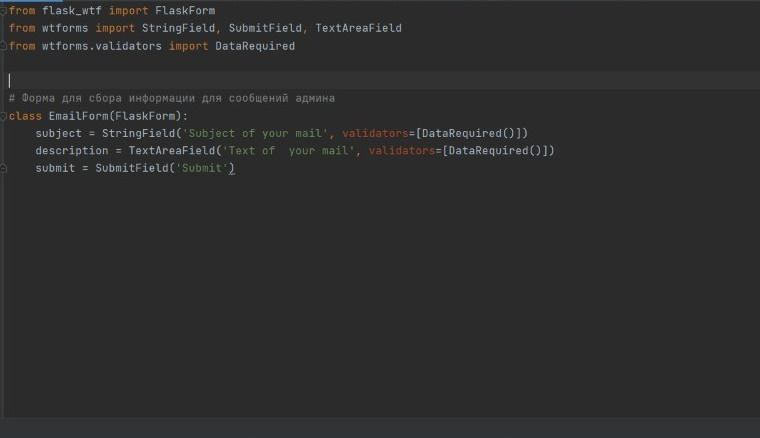
ProfileForm – форма для редактирования профиля. Запрашивает имя, фамилию, возраст, почту и контакт(похож на регистрацию, но без пароли)



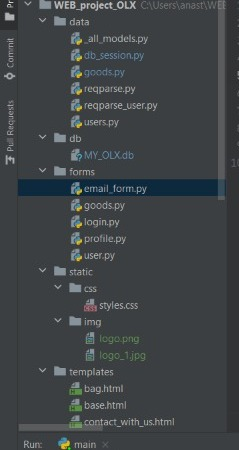
GoodsForm – форма для добавления товара. Запрашивает название, цену, описание, контактов и булевое значение «Продано или нет»



EmailForm – форма для отправки писем. Запрашивает тему, описание письма или объявления всем пользователям)

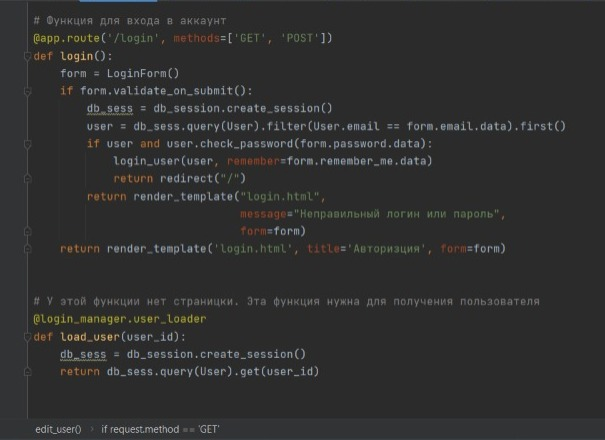


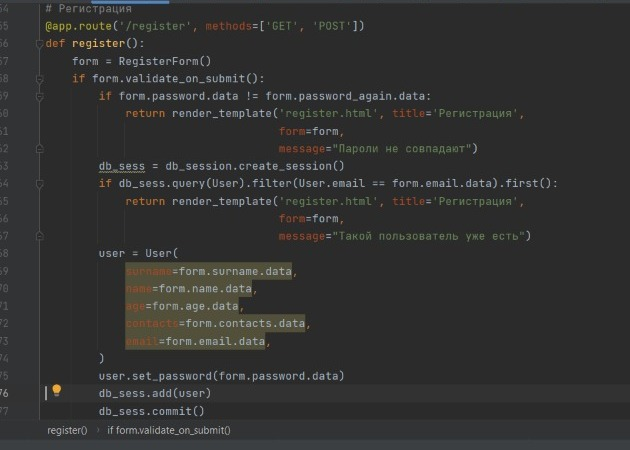
В data находятся ORM модели и подключения. В db находится база данных. В forms находятся формы. В static есть две папки: css и img. В css есть файл styles.css для дизайна страниц, а в img находятся картинки. В папке templates находятся шаблоны страниц. Файлы, находящиеся вне папок являются либо тестирующими, либо основными.



Интересные технологии:

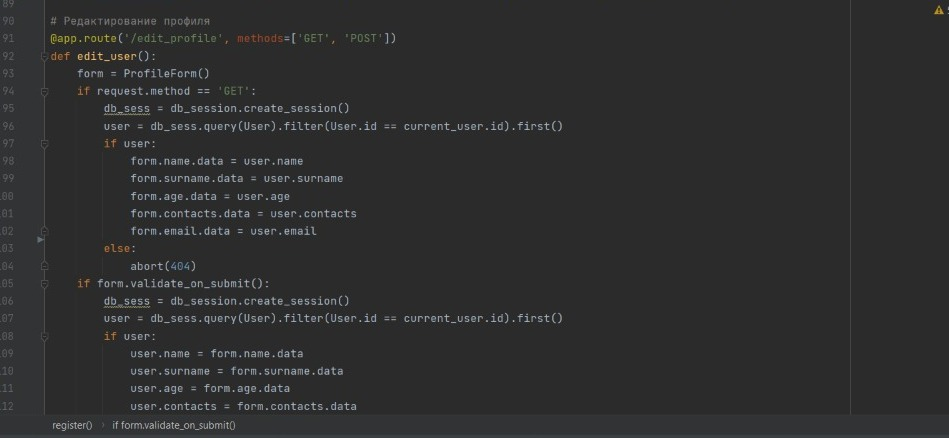
1. Регистрация и авторизация





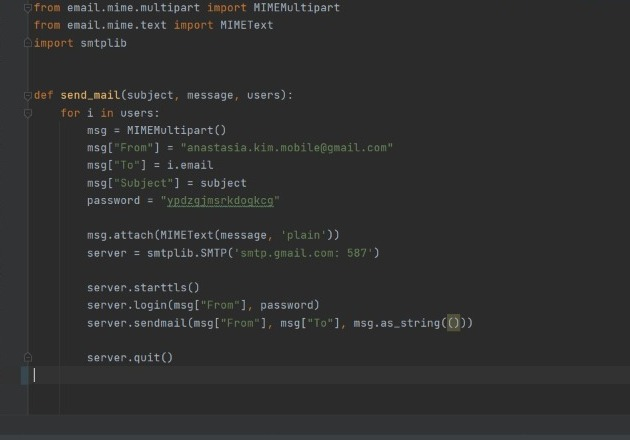
1. Редактирование профиля

Остановлюсь на втором(так как первый был разобран в классе). Кнопка редактирования профиля находится в профиле каждого авторизованного пользователя. Эта кнопка отправляет пользователя на ‘/edit\_profile’, а обработчиком является edit\_user. В начале запускаю форму, получаю ответ пользователя. Если данные в форме действительны, то вношу изменения в базу данных и отправляю пользователя на его профиль. Профиль же представлен в виде таблицы со всеми данными. Чтобы перейти в профиль нужно нажать на кнопку в верхнем меню



1. Почта

Реализовано с помощью smtplib и email. В файле sending\_mails(subject, message, users). Импортирую MIMEMultipar и MIMEText. Каждому пользователю отпраляю письмо, поэтому использую цикл. Записываю почту-источник, адресата, тему и пароль. Создаю сервер для отправки, логинюсь и отправляю письмо, а затем выхожу из сервера (дружелюбие к среде). Чтобы удобнее получать тему и содержание письма, создам форму mail.html, где запрашиваю тему и содержание письма, а потом отправляю

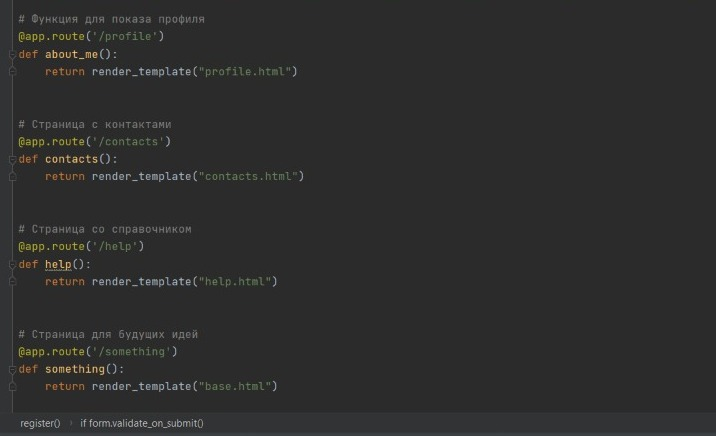


1. Я сделала несколько более мелких страничек, которые есть почти в каждом сайте.

Страничка «Помощь» нужна как инструкция для пользователей, если у них возникли вопросы.

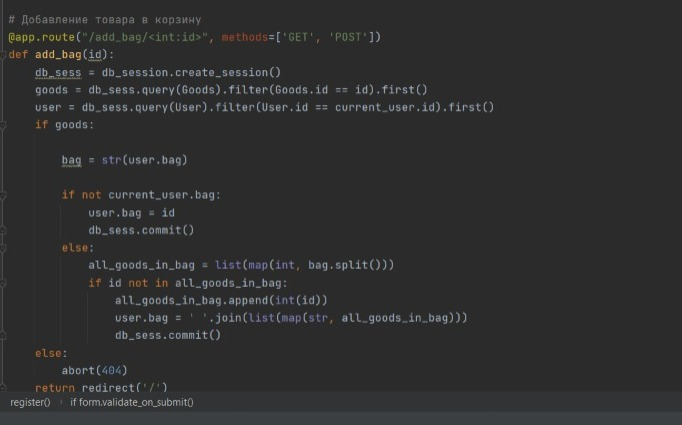
Страничка «Мои контакты» нужна для жалоб или пожеланий сайту, или как техподдержка.

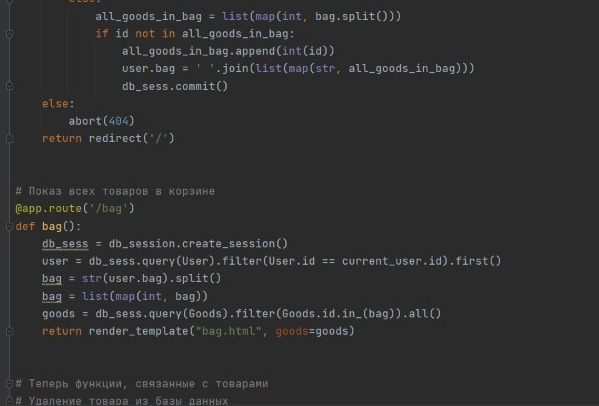
Страничка «Что-нибудь еще» нужна, если вдруг появится идея для доработки или обновления сайта.



1. Корзина

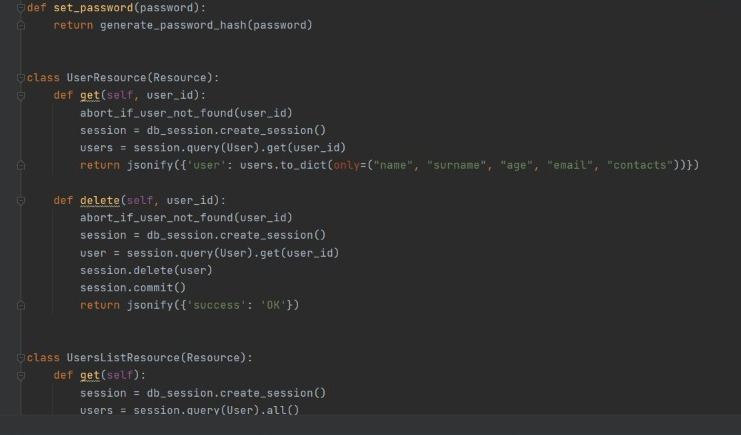
Если пользователь нашел какой-то хороший товар, но хочет еще немного поискать, но ему нужно сохранить где-то предыдущий товар. Для этого нужна корзина. Он добавляет все такие товары в корзину, а потом может посмотреть «Корзину» , где будут отображены все сохраненные товары. В коде есть специальный раздел для корзины(в коде обозначено как «bag»)

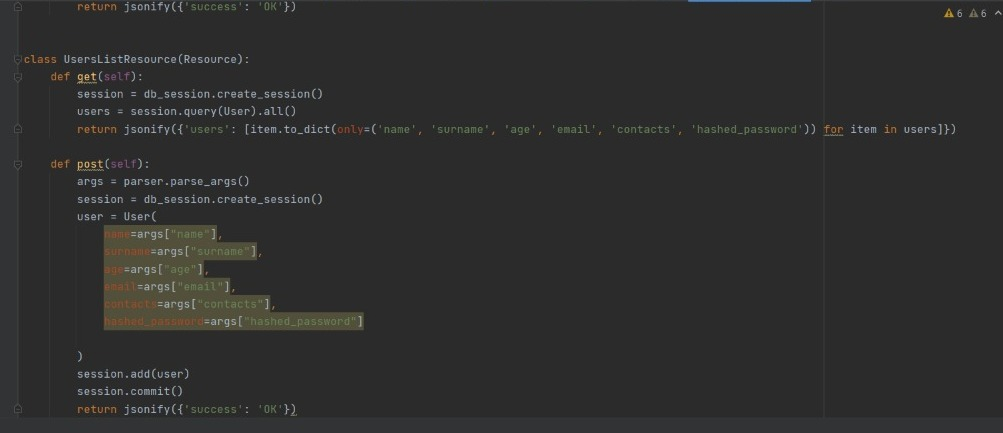




1. Я подключила API и также есть файл test.py для его тестов. За API отвечают файлы users\_resource, goods\_resource.

В каждом из них по два класса. Первый для работы с одним объектом, второй для группы объектов. В классах для одного объекта реализованы методы get и delete для получения и удаления через API, а в классах для группы объектов реализованы методы для получения и добавления через get и post.





1. В файле styles.css сделан дизайн сайта

Описание технологии

Я использовала flask, flask\_login, flask\_restful, flask\_wtf, sqlalchemy, WTForms, Werkzeug, requests для основы сайта, то есть формы, шаблоны, запуск страничек, безопасность, аза данных, регистрация и авторизация

Для дополнительных функций пользовалась python-dotenv, email, smtplib

Все необходимые для запуска библиотеки

aniso8601==9.0.1  
certifi==2021.10.8  
charset-normalizer==2.0.12  
click==8.1.0  
colorama==0.4.4  
Flask==2.1.0  
Flask-Login==0.6.0  
Flask-RESTful==0.3.9  
Flask-WTF==1.0.1  
greenlet==1.1.2  
idna==3.3  
importlib-metadata==4.11.3  
itsdangerous==2.1.2  
Jinja2==3.1.1  
MarkupSafe==2.1.1  
python-dotenv==0.20.0  
pytz==2022.1  
requests==2.27.1  
six==1.16.0  
SQLAlchemy==1.4.34  
SQLAlchemy-serializer==1.4.1  
urllib3==1.26.9  
Werkzeug==2.1.0  
WTForms==3.0.1  
zipp==3.7.0