**На чём и с помощью чего была написана программа (требования):**

* PostgreSQL 10.10
* Python 3.7.3
* Библиотеки для Python:
  + - * sqlalchemy==1.3.10
      * aiopg==1.0.0
      * aiohttp==3.6.2
      * aiohttp\_jinja2==1.2.0
      * jinja2==2.10.1
      * yaml==5.1.2
      * csv==1.0
      * random
      * datetime
      * pathlib

**Запуск программы:**

**Есть два способа запуска программы:**

1. Установка с нуля (Пошаговая инструкция представлена ниже)
2. Перейти по ссылке: <http://88.200.215.139:8080> (Может не сработать из-за динамического IP. Если не сработает, но захотите попробовать – свяжитесь со мной. Отправлю актуальный IP.)

**Установка с нуля:**

1. Скачать и установить все программы и библиотеки, приведенные в требовании.
2. В PostgresSQL:
   1. Создать пользователя с логином/паролем admin/admin и с правом создавать БД.
   2. Создать БД с именем book\_db
3. Запустить файл init\_db.py, который находится в папке проекта. Данный файл создаст таблицы и заполнит их.
4. Запустить файл «main\main.py» для запуска веб-сервера

*Примечание: Логину, паролю и названию БД можно присвоить другие значения. Но в данном случае придется отредактировать файл «config\book.yaml»*

**Схема БД**

Помимо минимального набора таблиц в БД была добавлена таблица «Ассортимент магазина». А также таблица «Товар» получила дополнительную колонку price:

* Пользователь - User(id, name, surname, fathers\_name, email);
* Товар(Книга) - Book(id, name, author, isbn, price);
* Магазин - Shop(id, name, address, post\_code);
* Заказ - Order(id, reg\_date(дата оформления заказа), user\_id(внешгий ключ для User.id));
* Позиция заказа - OrderItem(id, order\_id(внешний ключ для Order.id), book\_id(внешний ключ для Book.id), book\_quantity(количество экземпляров книги), shop\_id(внешний ключ для Shop.id));
* Ассортимент магазина – Shop\_inventory(id, shop.id, book.id, book\_quantity).

**Реализованные API со следующим набором функций:**

**Получение данных пользователя (имя, адрес эл. Почты и т.п.);**

Выбран метод GET

Данные можно получить 3-мя способами:

1. Данные всех пользователей: <http://127.0.0.1:8080/users-info>
2. Данные конкретного пользователя: http://127.0.0.1:8080/users-info/id/{число} , где {число} это цифра между 1 и 40.
3. Данные конкретного пользователя: http://127.0.0.1:8080/users-info/email/{mail} , где {mail} это mail из БД. Пример: *name123@google.gov*

**Просмотр истории заказов пользователя**

Выбран метод GET

Историю можно получить 2-мя способами:

1. По id пользователя: http://127.0.0.1:8080/orders/id/{число}
2. По Email пользователя: http://127.0.0.1:8080/orders/email/{mail}

**Добавление нового заказа (N книг каждая из которых в M количестве);**

Выбран метод POST, но получился post (добавление нового заказа) + put (редактирование количества книг оставшихся после нового закза)

Как добавить: На стартовой странице заполнить форму и нажать Add order.

**Просмотр ассортимента определенного магазина;**

Выбран метод GET

Ассортимент можно посмотреть двумя способами:

1. По id магазина: http://127.0.0.1:8080/shop-stock/id/{число} где {число} это цифра между 1 и 40.
2. По названию магазина: http://127.0.0.1:8080/shop-stock/name/{name}, где {name} – название магазина. Пример: «Милльда»

*Примечания: Если ввести значения, которых нет, сервер сообщит, что пользователя/книги/магазина нет в списке.*