**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Кафедра безопасности информационных систем**

**КУРСОВАЯ РАБОТА**

**«Разработка информационной системы для продажи книг»**

по дисциплине «Веб-технологии и защита информации»

**Исполнитель**:

Студент 3 курса, группы ИСТ-831

направление подготовки **09.03.02**

\_\_\_\_\_\_\_\_\_\_\_\_\_ /Е.И. Пономарев/

«\_\_\_»\_\_\_\_\_\_\_\_ 2020

**Принял**:

Старший преподаватель кафедры БИС,

\_\_\_\_\_\_\_\_\_\_\_\_\_/В.А. Евстигнеев/

«\_\_\_»\_\_\_\_\_\_\_\_ 2020

Санкт-Петербург

2020

РЕФЕРАТ

Отчет 13500 с., 128 стр., 17 рис., 21 табл., 5 источн., 7 прил.

ИНФОРМАЦИОННЫЕ СИСТЕМЫ, РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ, СТРУКТУРА ПРОЕКТА ИНФОРМАЦИОННОЙ СИСТЕМЫ, ВЕБ-ТЕХНОЛОГИИ, ВЕБ-ПРИЛОЖЕНИЯ.

Объект исследования данной курсовой работы — информационная система по продаже книг.

Цель проекта — спроектировать и разработать данную систему в соответствии со всеми актуальными и современными методами разработки.

Область применения результатов — сфера коммерческой продажи и развитие навыков профессиональная сфера в IT-технологиях.

Новизна объекта разработки - к сожалению, идея и функционал данного проекта не отличается особой новизной, что не позволяет проекту заработать свой банк постоянной аудитории без значительного коммерческого вмешательства со стороны.

В ходе выполнения данной курсовой работы были получены следующие результаты:

1. Разработана база данных в соответствии со всеми нормальными формами.
2. Разработана структура проекта.
3. Разработана программная часть в соответствии со всеми современными нормами разработки веб-приложений.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#__RefHeading___Toc1978_759293715)

[1 Постановка задачи 5](#__RefHeading___Toc1980_759293715)

[2 Определение целевой аудитории и групп пользователей 5](#__RefHeading___Toc2337_3930089672)

[3 Выбор средств 6](#__RefHeading___Toc1984_759293715)

[4 Разработка базы данных 7](#__RefHeading___Toc1986_759293715)

[4.1 Функции системы 7](#__RefHeading___Toc1988_759293715)

[4.2 Описание полей таблиц в базе данных 8](#__RefHeading___Toc1990_759293715)

[4.3 ER-диаграмма 17](#__RefHeading___Toc1591_870526701)

[5 Разработка карты сайта 17](#__RefHeading___Toc1992_759293715)

[6 Разработка структуры групп пользователей 20](#__RefHeading___Toc1994_759293715)

[7 Разработка программного кода 21](#__RefHeading___Toc2339_3930089672)

[7.1 Разработка авторизации и регистрации 24](#__RefHeading___Toc1996_759293715)

[7.2 Разработка маршрутизации 26](#__RefHeading___Toc1998_759293715)

[7.4 Разработка слоев обработки данных 28](#__RefHeading___Toc2000_759293715)

[7.5 Разработка слоя данных 30](#__RefHeading___Toc2002_759293715)

[7.6 Разработка вспомогательных утилит 30](#__RefHeading___Toc2004_759293715)

[7.7 Разработка html-страниц 32](#__RefHeading___Toc9885_3433738574)

[7.8 Разработка конфигурации проекта 33](#__RefHeading___Toc9887_3433738574)

[8 Разработка структуры дерева каталогов проекта 34](#__RefHeading___Toc9698_3614498399)

[9 Примеры работы информационной системы 35](#__RefHeading___Toc2006_759293715)

[ЗАКЛЮЧЕНИЕ 39](#__RefHeading___Toc2008_759293715)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 40](#__RefHeading___Toc2010_759293715)

[ПРИЛОЖЕНИЕ А 41](#__RefHeading___Toc2012_759293715)

[ПРИЛОЖЕНИЕ Б 47](#__RefHeading___Toc9545_3433738574)

[ПРИЛОЖЕНИЕ В 59](#__RefHeading___Toc9547_3433738574)

[ПРИЛОЖЕНИЕ Г 79](#__RefHeading___Toc9549_3433738574)

[ПРИЛОЖЕНИЕ Д 81](#__RefHeading___Toc9551_3433738574)

[ПРИЛОЖЕНИЕ Е 85](#__RefHeading___Toc9553_3433738574)

[ПРИЛОЖЕНИЕ И 127](#__RefHeading___Toc9555_3433738574)

# ВВЕДЕНИЕ

В 21 век произошел невероятный технологический скачок, давший начало новой информационной эпохе. Эпохе роботов, сети Интернет, смартфона и социальных сетей. Эпохе, которая спровоцировала появление большого количества новых профессий, которые облегчает жизнь другим людям, разрабатывая различной специфики сервисы. Начиная от доставки еды и заканчивая автоматизированной системы безопасности предприятия.

Но время не стоит на месте, и все больше старых вакансий становятся связаны с информационным технологиями, которые требуют базовое владение компьютерными навыками. В качестве примера можно привести пандемию 2020 года, которая буквально остановила развитие экономики многих стран и заставила представителей многих профессий переходить на дистанционный формат работы. Естественно, в первое время многие сталкивались с многими трудностями в освоение новой для себя сферы, что негативно сказывалось на эффективности работы различных предприятий. Стоит опять заметить важность профессии IT-работника, так как их эффективный труд обеспечил быстрый скачок от очной работы до дистанционной.

Исходя из вышесказанного стоит сказать, что с развитием технологий и информационного пространства в сегодняшнем мире даже обычному человеку следует обладать базовыми навыками программирования и разработки информационных системы, развитие которых достигает своего апогея во время непосредственной разработки. И в качестве примера стоит выбрать шаблонный проект, такой как интернет-магазин, так как для его успешной реализации требуется обладать знаниями в многочисленных сферах веб-приложений, что позволяет значительно повысить эффективность учебы.

# **1** **Постановка задачи**

Необходимо разработать информационную систему «Интернет-магазин по продажи книг». Для большего удобства декомпозируем задачу. Требуется:

1. разработать базу данных;
2. продумать карту сайта;
3. разработать дизайн сайта;
4. прописать программную логику;

Разработка программного кода состоит из нескольких этапов:

1. выбрать средства реализации;
2. выбрать структуру проекта;
3. написать эффективный программный код

Разработка базы данных состоит из нескольких этапов:

1. составление сущностей
2. выбор средств реализации
3. разработка таблиц

# **2 Определение целевой аудитории и групп пользователей**

Продаваемая продукция рассчитана для читающей аудитории.

Так как интернет-магазин подразумевает систему подбора книг исходя из возраста, следует выделить основные группы пользователей:

1. подростки(13-17 лет);
2. молодежь(18-30 лет);
3. люди среднего возраста(31-55 лет);
4. люди преклонного возраста(от 56 лет);

Распределение прав доступа приведено в таблице 1.

Таблица 1 — Распределение прав доступа к информационной системе

|  |  |  |
| --- | --- | --- |
| Тип пользователя | Права доступа | Описание группы |
| Все нижеперечисленные группы | Просмотр товаров |  |
| Админ-пользователь с правами доступа на чтение и запись | Редактирование, добавление, удаление товаров, просмотр финансовой статистики | Админ, имеющий право вносить изменения в базу данных посредство API и получать статистические данные |
| Админ-пользователь с правами доступа на чтение | Просмотр финансовой статистики | Админ, имеющий право только получать статистические данные |
| Зарегистрированный пользователь | Добавление в корзину, заказ товаров | Пользователь, имеющий личный кабинет на сайте |
| Незарегистрированный пользователь | Создание личного кабинета | Простой посетитель сайта |

# **3** **Выбор средств**

В качестве СУБД была выбрана MYSQL, так как данная СУБД является бесплатной и поддерживает многопоточность запросов, что при гипотетическом большом трафике посетителей информационной системы играет большую роль.

В качестве языка программирования разработки системы был выбран прогрессивный и быстрорастущий в сфере интернет-технологий язык Python из-за своей универсальности.

В качестве веб-фреймворка был выбран Flask из-за своей скорости по сравнению с аналогами.

Также были задействованы такие библиотеки Python как:

1. pymysql – для подключения к СУБД Mysql из программного кода;
2. flask-login – для реализация системы авторизации пользователях;
3. jinja2 – для рендеринга html-страниц;
4. smptlib,email – для отсылки электронных писем;

Также в frontend-разработке присутствует язык программирования JavaScript и дополнительные фреймворки, такие как:

1. Jquery – для отправки запросов;
2. Bootstrap – для материального дизайна;

В роли веб-сервера выступает Nginx в связке с Gunicorn3.

# **4** **Разработка базы данных**

## **4.1 Функции системы**

Данная база данных реализуется для поддержания данного функционала сайта:

1. вывод книг пользователю;
2. возможность пользователю управлять своей корзиной;
3. реализация отдельной админ-панели для владельца сайта, позволяющую просматривать финансовую статистику сайта;
4. реализация отдельной админ-панели сайта для модераторов, позволяющую редактировать, добавлять, удалять информацию о книгах;
5. фильтрация книг по тегам, по рейтингу;
6. вывод доступных тегов для фильтрации;
7. хранения пользовательской информации;

## **4.2 Описание полей таблиц в базе данных**

Описание полей таблицы «Пользователь» приведено в таблице 2.

Таблица 2 — Описание полей таблицы «Пользователь»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| id | INT(11) | AUTOINCREMENT,NOT NULL  PRIMARY KEY | Уникализация пользователя |
| last\_name | VARCHAR(45) |  | Фамилия пользователя |
| first\_name | VARCHAR(45) | NOT NULL | Имя пользователя |
| email | VARCHAR(45) | NOT NULL | EMAIL пользователя |
| birthdate | DATE | NOT NULL | Дата рождения |
| password\_hash | VARCHAR(45) | NOT NULL | Хеш пароля |

Описание полей таблицы «Товар» приведено в таблице 3.

Таблица 3 — Описание полей таблицы «Товар»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| id | INT(11) | AUTOINCREMENT  PRIMARY KEY | Уникализация товара |
| title | VARCHAR(45) | NOT NULL | Название книги |
| author | VARCHAR(45) | NOT NULL | Автор книги |
| desc | VARCHAR(255) |  | Описание книги |
| cost\_sale | FLOAT | NOT NULL | Стоимость продажи |
| quantity | INT | NOT NULL | Количество товара на складе |
| imageLink | VARCHAR(45) |  | Название файла фотографии изображения товара |

Описание полей таблицы «Заказ» приведено в таблице 4.

Таблица 4 — Описание полей таблицы «Заказ»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| id | INT | AUTOINCREMENT  PRIMARY KEY | Уникализация записи |

Продолжение таблицы 4

| Название поля | Тип данных | Доп параметры | Назначение |
| --- | --- | --- | --- |
| user\_id | INT | NOT NULL  FOREIGN KEY | Идентификатор пользователя |
| date | TIMESTAMP | NOT NULL  DEFAULT:CURRENT\_TIMESTAMP | Дата заказа |
| status | INT | NOT NULL | Статус заказа |
| address\_id | INT | NOT NULL  FOREIGN KEY | Идентификатор адреса |
| description | VARCHAR(255) |  | Описание заказа |
| email | VARCHAR(45) |  | Email клиента |
| phone | VARCHAR(11) | NOT NULL | Телефон клиента |

Описание полей таблицы «Корзина» приведено в таблице 5.

Таблица 5 — Описание полей таблицы «Корзина»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| user\_id | INT | NOT NULL  FOREIGN KEY | Идентификация пользователя |
| product\_id | INT | NOT NULL  FOREIGN KEY | Идентификация товара |
| count | INT | NOT NULL | Количество товара |

Описание полей таблицы «Забронированная книга» приведено в таблице 6.

Таблица 6 — Описание полей таблицы «Забронированная книга»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| user\_id | INT | NOT NULL  FOREIGN KEY | Идентификация пользователя |
| product\_id | INT | NOT NULL  FOREIGN KEY | Идентификация товара |
| count | INT | NOT NULL | Количество товара |
| price | FLOAT | NOT NULL | Цена товара |

Описание полей таблицы «Персонал» приведено в таблице 7.

Таблица 7 — Описание полей таблицы «Персонал»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| id | INT | AUTOINCREMENT  PRIMARY KEY | Уникализация записи |
| last\_name | VARCHAR(45) | NOT NULL | Фамилия |
| first\_name | VARCHAR(45) | NOT NULL | Имя |
| birthdate | DATE |  | Дата рождения |
| psp | VARCHAR(45) | NOT NULL | Паспорт сотрудника |
| email | VARCHAR(45) |  | Почтовый адрес |

Продолжение таблицы 7

| Название поля | Тип данных | Доп параметры | Назначение |
| --- | --- | --- | --- |
| phone | VARCHAR(12) | NOT NULL | Телефон |
| vacancy | VARCHAR(15) | NOT NULL | Должность сотрудника |

Описание полей таблицы «Поставщик» приведено в таблице 8.

Таблица 8 — Описание полей таблицы «Поставщик»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| id | INT | AUTOINCREMENT  PRIMARY KEY | Уникализация записи |
| company\_name | VARCHAR(45) | NOT NULL | Название компании |

Описание полей таблицы «Адрес» приведено в таблице 9.

Таблица 9 — Описание полей таблицы «Адрес»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| id | INT | AUTOINCREMENT  PRIMARY KEY | Уникализация записи |
| district | VARCHAR(35) | NOT NULL | Район |
| street | VARCHAR(35) | NOT NULL | Улица |

Продолжение таблицы 9

| Название поля | Тип данных | Доп параметры | Назначение |
| --- | --- | --- | --- |
| flat | VARCHAR(4) |  | Номер квартиры |
| floor | VARCHAR(4) |  | Этаж |
| porch | VARCHAR(3) | NOT NULL | Номер парадная |
| house | VARCHAR(5) | NOT NULL | Номер дома |

Описание полей таблицы «Рейтинг» приведено в таблице 10.

Таблица 10 — Описание полей таблицы «Рейтинг»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| product\_id | INT | FOREIGN KEY | Идентификатор товара |
| user\_id | INT | FOREIGN KEY | Идентификатор пользователя |
| mark | INT | NOT NULL | Оценка |

Описание полей таблицы «Админ» приведено в таблице 11.

Таблица 11 — Описание полей таблицы «Админ»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| user\_id | INT | FOREIGN KEY | Идентификатор пользователя |

Продолжение таблицы 11

| Название поля | Тип данных | Доп параметры | Назначение |
| --- | --- | --- | --- |
| staff\_id | INT | FOREIGN KEY | Данные о персонале |
| level\_of\_access | INT | NOT NULL | Уровень доступа |

Описание полей таблицы «Логи» приведено в таблице 12.

Таблица 12 — Описание полей таблицы «Логи»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| id | INT | AUTOINCREMENT  PRIMARY KEY | Уникализация записи |
| admin\_id | INT | NOT NULL  FOREIGN KEY | Идентификатор админа(модератора) |
| operation | VARCHAR(255) | NOT NULL | Выполненная операция |
| date | TIMESTAMP | NOT NULL  DEFAULT: CURRENT\_TIMESTAMP | Дата выполнения |

Описание полей таблицы «Закупки» приведено в таблице 13.

Таблица 13 — Описание полей таблицы «Закупки»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| id | INT | AUTOINCREMENT  PRIMARY KEY | Уникализация записи |
| date | TIMESTAMP | NOT NULL  DEFAULT:CURRENT\_TIMESTAMP | Дата закупки |
| product\_id | INT | NOT NULL  FOREIGN KEY | Идентификатор товара |
| quantity | INT | NOT NULL | Количество товара |
| cost\_purchase | FLOAT | NOT NULL | Цена товара |
| supplier\_id | INT | NOT NULL  FOREIGN KEY | Идентификатор поставщика |

Описание полей таблицы «Курьер» приведено в таблице 14.

Таблица 14 — Описание полей таблицы «Курьер»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| employer\_id | INT | FOREIGN KEY | Идентификатор сотрудника |
| count\_of\_delivery | INT | NOT NULL  DEFAULT:0 | Количество доставленных заказов |

Описание полей таблицы «Взятый\_заказ» приведено в таблице 15.

Таблица 15 — Описание полей таблицы «Взятый\_заказ»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| delivery\_man | INT | FOREIGN KEY | Идентификатор курьера |
| order\_id | INT | FOREIGN KEY | Идентификатор заказа |

Описание полей таблицы «Теги» приведено в таблице 16.

Таблица 16 — Описание полей таблицы «Теги»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| id | INT | PRIMARY KEY  AUTOINCREMENT | Идентификатор курьера |
| tag | VARCHAR(32) | NOT NULL,  UNIQUE | Тег товара |

Описание полей таблицы «Тег\_товара» приведено в таблице 17.

Таблица 17 — Описание полей таблицы «Тег\_товара»

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Доп параметры | Назначение |
| tag\_id | INT | FOREIGN KEY | Идентификатор теги |
| product\_id | INT | FOREIGN KEY | Идентификатор товара |

## **4.3 ER-диаграмма**

ER-диаграмма приведена на рисунке 1

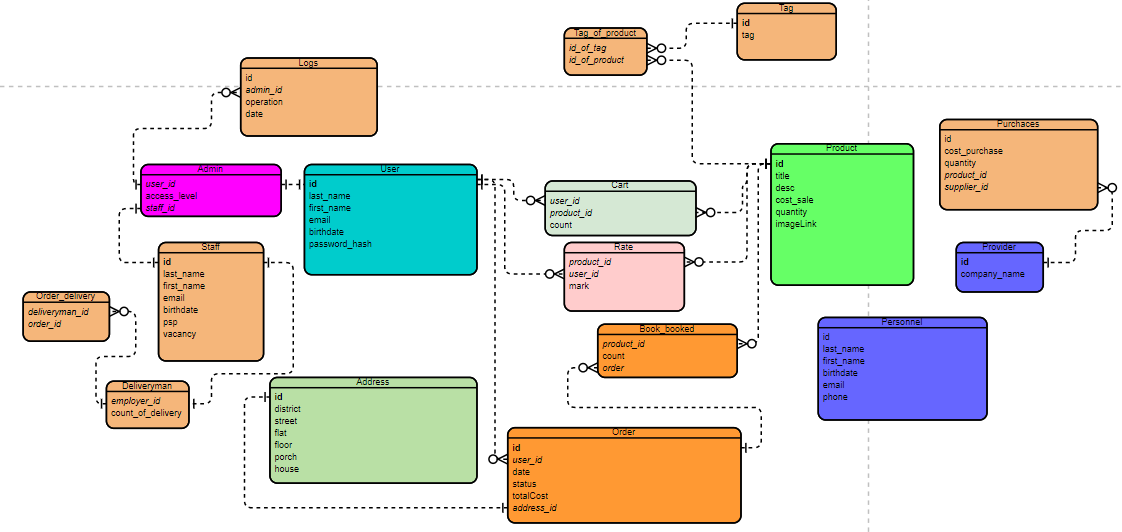


Рисунок 1 - ER-диаграмма

# **5 Разработка карты сайта**

Карта сайта приведена на рисунках 2 и 3.

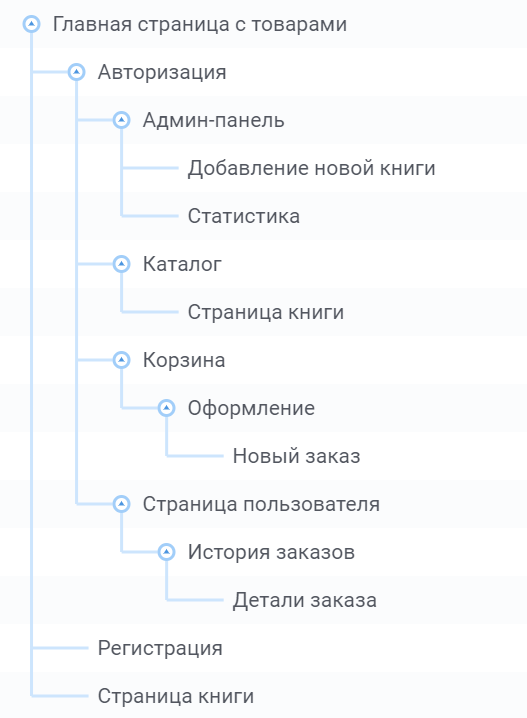


Рисунок 2 — Карта сайта

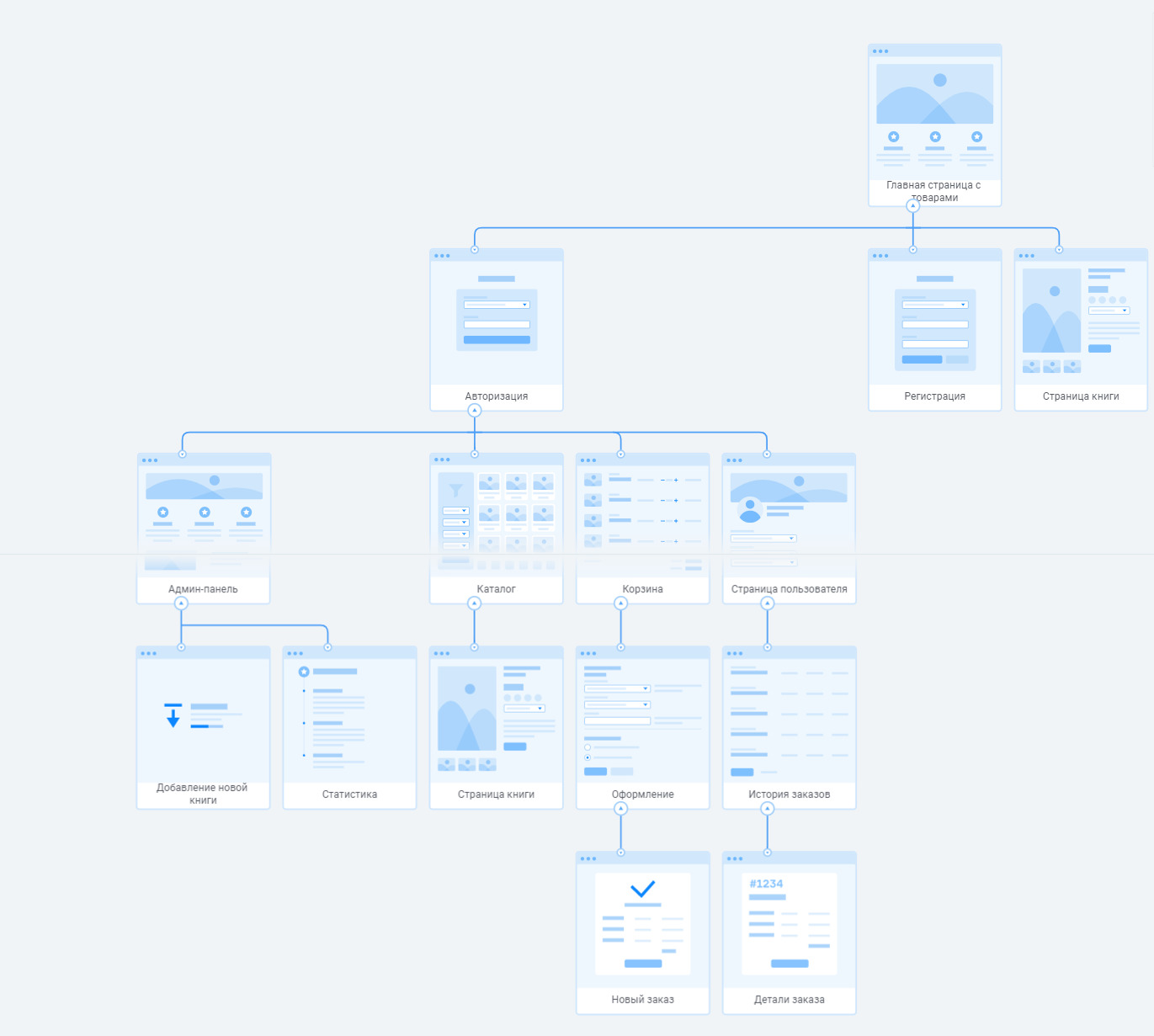


Рисунок 3 — Карта сайта

# **6** **Разработка структуры групп пользователей**

Для понимания структуры идентификации админ-пользователей стоит заметить, что статус админа и уровень прав доступа определяется во время стандартной операции авторизации, которую проходят даже обычные пользователи для доступа к личному кабинету. Другими словами, и для общедоступных операций купли-продажи, и для доступа к приватной части сайта используется один и тот же объект пользователя, что позволяет избавиться от второй системы авторизации специально для пользователей с правами админа.

Более подробная структура приведена на рисунке 4.

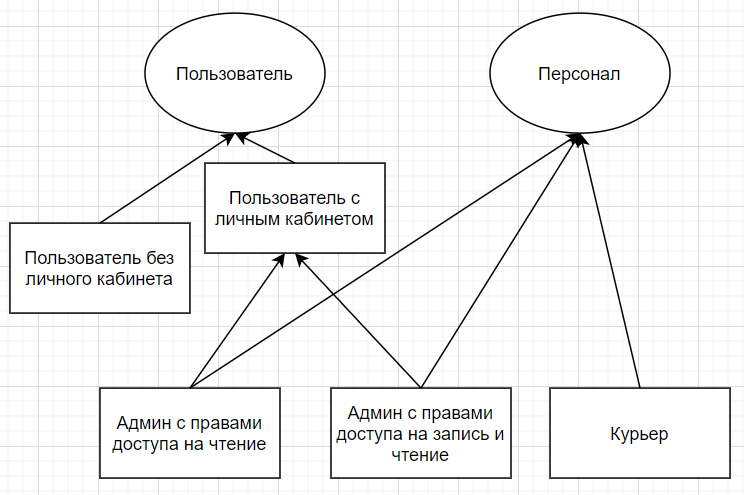


Рисунок 4 — Структура групп пользователей

# **7** **Разработка программного кода**

Для начала следует продумать структуру проекта

Структура проекта с программной точке зрения приведена на рисунке 5.

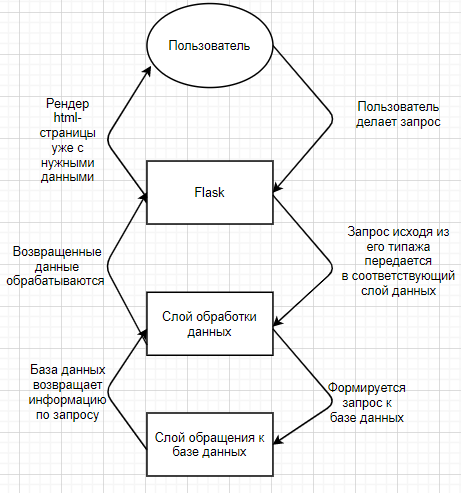


Рисунок 5 — Программная структура проекта

Следует сказать, что данные возвращаются из слоев обработки в формате словаря, где:

1. обязательно присутствует ключ status, содержащий в себе статус выполненной операции
2. обязательно присутствует ключ message, содержащий в себе краткое описание статуса выполненной операции
3. присутствует ключ data, в котором содержатся полученные от запроса данные

Данный подход позволяет в будущем мигрировать серверную часть проекта в отдельную самодостаточную структуру, реализующую API-подход для доступа различных frontend-сервисов.

Так как структура большинства html-страниц похожа, то они будут подчинять шаблону, приведенному на рисунке

Структура большинства html-страниц приведена на рисунке 6.

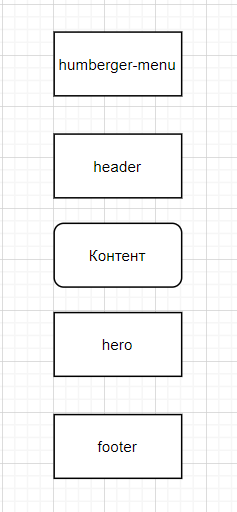


Рисунок 6 — Структура html-страниц

Стоит раскрыть понятия, приведенные на рисунке выше.

Humberger-menu – представление html-страницы для мобильных устройств

Пример такого меню показан на рисунке 7.

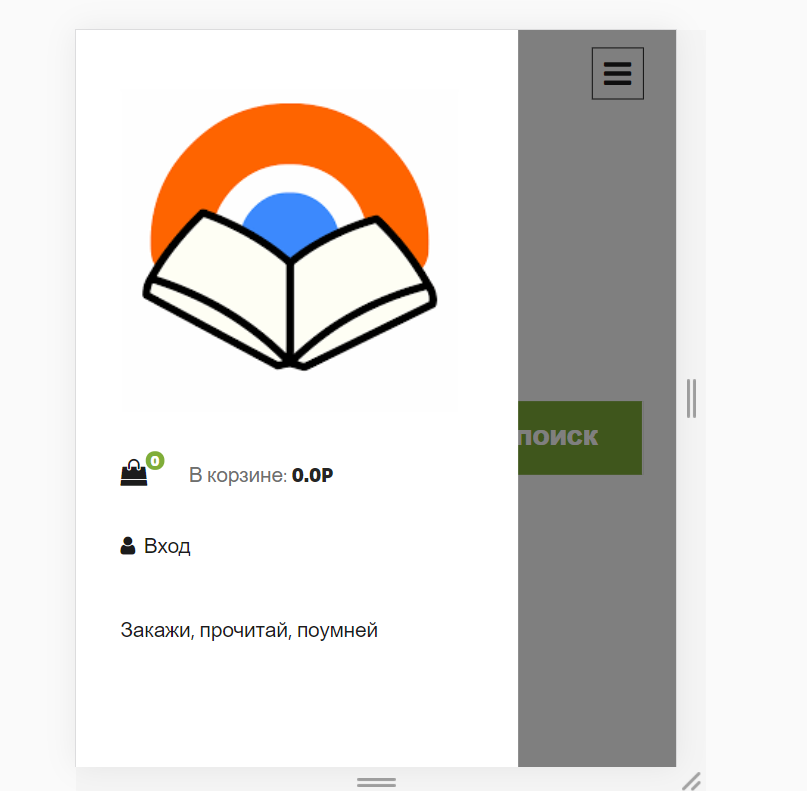


Рисунок 7 — Пример мобильной версии сайта

Header - это блок в верхней части страницы сайта, который виден на всех страницах сайта. Как правило, содержит логотип, меню, контакты, переключатель языков или корзину, если в интернет-магазине несколько страниц и нужна общая корзина. Пример этого блока показан на рисунке 8.

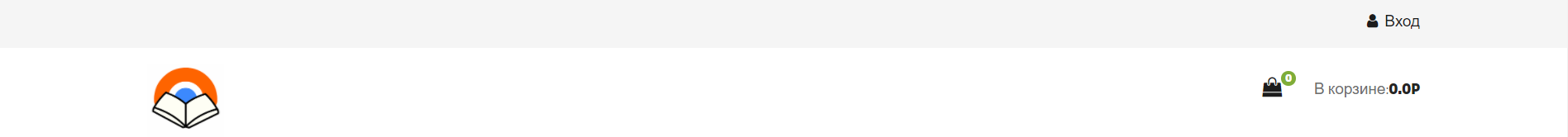


Рисунок 8 — Пример заголовка html-страницы

Hero – блок сайта, содержащий в себе элемент поиска по каталогу

Пример этого блока показан на рисунке 9.

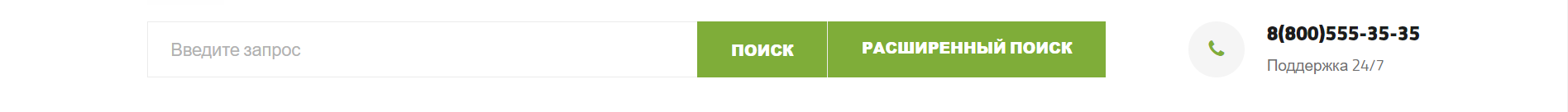


Рисунок 9 — Пример блока поиска на hml-странице

Footer - подвал, это визуальный элемент, расположенный в нижней части страниц. В его поле обычно размещают ссылки на другие материалы ресурса, дублируют меню, располагают иконки соцсетей, форму подписки.

Пример этого блока показан на рисунке 10.

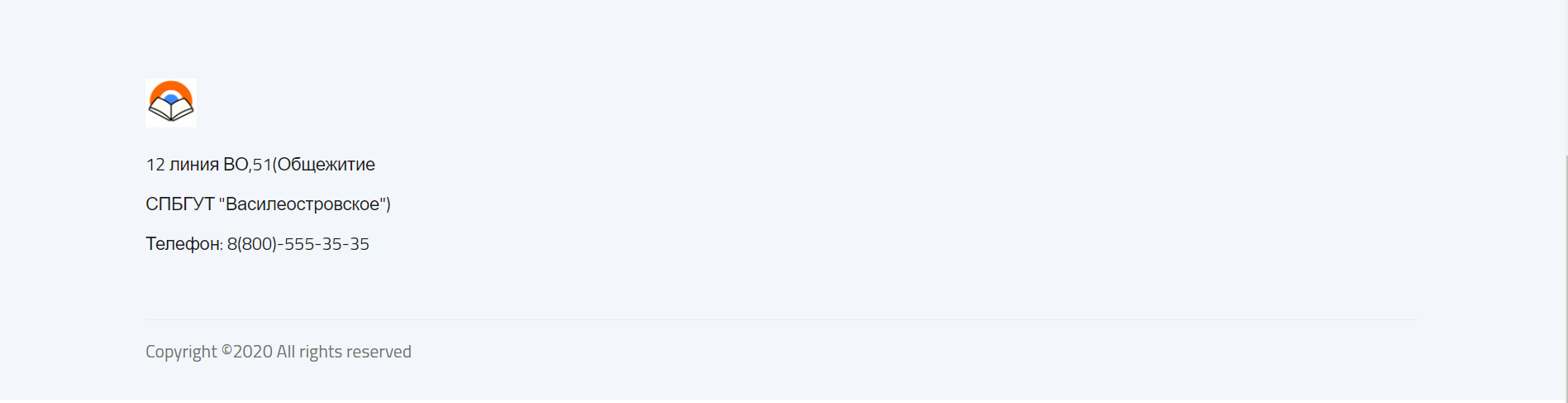


Рисунок 10 — Пример подвала html-страницы

Поэтому чтобы уменьшить количество повторяющейся html-верстки следует вынести перечисленные ниже блоки в отдельные файлы и добавлять их путем использования библиотеки jinja2.

## **7.1** **Разработка авторизации и регистрации**

За функционировании системы отвечает фреймворк flask-login.

Данный фреймворк автоматически помещает идентификатор пользователя в сеанс, что позволяет значительно упростить разработку со стороны идентификации пользователей.

Для его функционирования необходимо прописать метод ,помеченный специальным декоратором и принимающий на вход идентификатор пользователя и возвращающий соответствующий объект, хранящий пользовательскую информацию. Данный метод представлен в приложении с обозначением А.

Проверить аутентифицирован ли пользователь можно с помощью специального свойства is\_authenticated. Если же пользователь не проходил процесс аутентификации, то flask-login будет возвращать AnonymousUserMixin, у которого вышесказанное свойство всегда будет False. Данная логика позволяет нам идентифицировать авторизирован ли пользователь или нет.

Примеры html-страниц авторизации и регистрации показаны на рисунках 11 и 12 соответственно.

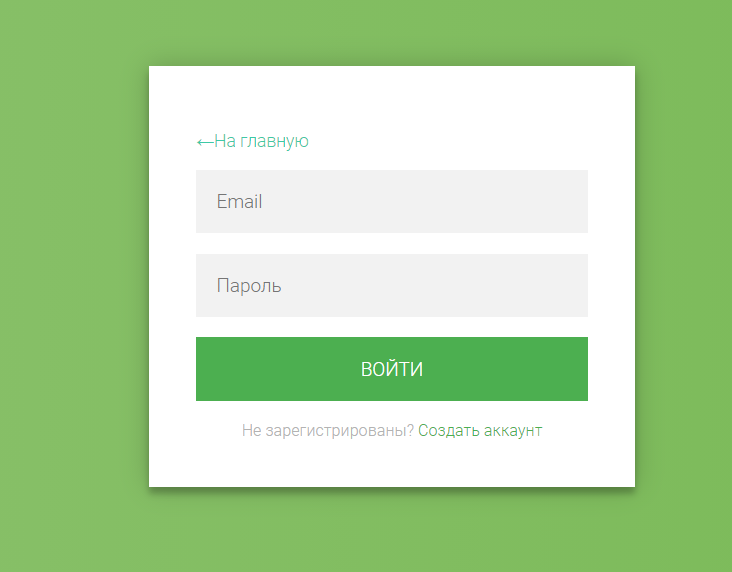


Рисунок 11 — Пример html-страницы авторизации

# 

Рисунок 12 — Пример html-страницы регистрации

## **7.2** **Разработка маршрутизации**

Ниже в таблице 18 приведена маршрутизация сайта.

Таблица 18 — Описание маршрутизации сайта

|  |  |  |
| --- | --- | --- |
| URL | Описание | Права |
| / | Редирект на /books/1 | Все пользователи |
| /search | Поиск книг | Все пользователи |
| /checkout | Оформление заказа | Только авторизованные пользователи |
| /order/<int:orderID>/cancel | Отмена заказа с идентификатором orderID | Только авторизованные пользователи |
| /order/<int:orderID> | Просмотр деталей заказа с идентификатором orderID | Только авторизованные пользователи |
| /user/orders' | Просмотр заказов пользователя | Только авторизованные пользователи |
| /order/new | Просмотр нового заказа | Только авторизованные пользователи |
| /cart | Просмотр корзины | Только авторизованные пользователи |
| /login | Страница авторизации | Только неавторизованные пользователи |
| /registration | Страница регистрации | Только неавторизованные пользователи |
| /user | Профиль пользователя | Только авторизованные пользователи |
| /logout | Выход пользователя | Только авторизованные пользователи |
| /books | Просмотр продукта | Все пользователи |
| /books/<int:page> | Просмотр продукта на странице page | Все пользователи |
| /books/details/<int:productID> | Просмотр деталей книги с идентификатор productID | Все пользователи |
| /admin | Админ-панель | Только авторизованные пользователи с админ-правами |

Продолжение таблицы 18

|  |  |  |
| --- | --- | --- |
| URL | Описание | Права |
| /admin/orders | Управление заказами | Только авторизованные пользователи с админ-правами |
| /admin/stat | Статистика магазина | Только авторизованные пользователи с админ-правами |
| /admin/product | Модерация книг | Только авторизованные пользователи с админ-правами |
| /admin/add | Добавление новой книги | Только авторизованные пользователи с админ-правами |

Листинг кода с маршрутизацией представлен в приложении с обозначением Б.

## **7.4** **Разработка слоев обработки данных**

Ниже в таблице 19 приведены описание слоев обработки данных.

Таблица 19 — Описание слоев обработки данных

|  |  |  |
| --- | --- | --- |
| Слой обработки данных | Функционал | Описание |
| Address.py | Добавление нового адреса | Слой отвечает за обработку данных, связанных с адресами |

Продолжение таблицы 19

| Слой обработки данных | Функционал | Описание |
| --- | --- | --- |
| Cart.py | Получение количества продуктов в корзине,получение полной стоимости корзины, удаление продукта из корзины,  добавление продукта в корзину, получение количество продукта в корзине, получение всех продуктов в корзине | Слой отвечает за обработку данных, связанных с корзиной пользователя |
| Order.py | Получение всех заказов пользователя, отмена конкретного заказа пользователем, получение описания заказа, добавления нового заказа, проверка наличия конкретного заказа у определенного пользователя, проверка наличия заказа вообще | Слой отвечает за обработку данных, связанных с заказами пользователя |
| Product.py | Получение всех продуктов с фильтрацией по id, по названию, по тегам, получение доступных тегов,добавление новой оценки продукта, получение рейтинга конкретного продукта,получение детали определенного продукта | Слой отвечает за обработку данных, связанных с продуктами для продажи |

Продолжение таблицы 19

| Слой обработки данных | Функционал | Описание |
| --- | --- | --- |
| User.py | Регистрация нового пользователя, получения информации о пользователе, валидация введенного пароля | Слой отвечает за обработку данных, связанных с пользователем |
| Admin.py | Добавление нового поставщика, новой книги,новой поставки, редактирование любой информации о книге, получение статистики | Слой отвечает за обработку данных, связанных с админ-панелью |

Листинг кода с маршрутизацией представлен в приложении с обозначением В.

## **7.5** **Разработка слоя данных**

За обращение к базе данных отвечает класс SqlExecuter, содержащий в себе подключение к самой базе и методы, выполняющие запросы. Формат возвращаемой информации из базы данных — словарь(ключ: значение)

Листинг кода с слоем данных представлен в приложении с обозначением Г.

## **7.6** **Разработка вспомогательных утилит**

Таблица, описывающая дополнительные классы, облегчающие функционирования системы, приведена ниже

Листинг кода вспомогательных утилит представлен в приложении с обозначением Д.

Ниже в таблице 20 приведено описание вспомогательных утилит.

Таблица 20 — Описание вспомогательных утилит

|  |  |
| --- | --- |
| Модуль | Описание |
| addressBuilder | Содержит в себе функцию buildAddress, возвращающую отформатированную строчку адреса, которой указали при оформления заказа |
| dateHelper | Содержит в себе функцию getDate, возвращающую текущую дату |
| emailSender | Содержит в себе класс EmailSender,который заключает в себя функцию sendEmailTo, отправляющую электронное письмо с деталями заказа на определенный адрес |
| imageHelper | Содержит в себе функцию makeFullPathToImage, возвращающую полный путь до файла картинки |
| md5Helper | Содержит в себе функции encrypt и validateS, где:   1. ecnrypt – возвращает хеш полученной строки; 2. validateS – сравнивает полученный 2 хеша; |
| pageHelper | Содержит в себе функцию getRangeOfPages,возвращающую диапазон страниц по количеству страниц и текущей страницы |
| security | Содержит в себе класс Security,который заключает в себя функцию escape\_sql, экранирующую все символы, которые могут спровоцировать атаку Sql-инъекции |

Продолжение таблицы 20

| Модуль | Описание |
| --- | --- |
| sqlQueryHelper | Содержит в себе функции buildSqlQueryByTags и buildSqlQueryByTagsAndPage, где:   1. buildSqlQueryByTags – возвращает сконструированный sql- 2. запрос на получение книг сортируя по тегам 3. buildSqlQueryByTagsAndPage – возвращает сконструированный sql-запрос на получение книг сортируя по тегам и корректирую ответ исходя из текущей страницы |
| tagsHelper | Содержит в себе функции makeArrayOfTagsToStr и makeTagsStrToArray, реализующие преобразование строки с тегами в массив и обратно соответственно |
| validationForm | Содержит в себе функцию validationEmail, проверяющую введенный email пользователя на соответствие шаблону email’а |
| logger | Содержит в себе класс AdminLogger с функцией log, которая сохраняет в базу данных выполненную админом операцию |

## **7.7** **Разработка html-страниц**

Листинг с html-версткой представлен в приложении с обозначением Е.

## **7.8** **Разработка конфигурации проекта**

Для удобства разработки проекта и его функционирования вообще стоит прописать его конфигурацию. Описание глобальных переменных в конфигурации проекта приведено в таблице 21.

Листинг кода с конфигурацией проекта представлен в приложении с обозначением И.

Таблица 21 — Описание глобальных переменных в конфигурации проекта

|  |  |  |
| --- | --- | --- |
| Глобальная переменная | Значение переменной | Описание |
| UPLOAD\_FOLDER | "app\\static\\img\\book" | Данная переменная отвечает за путь к папке, в которой будут храниться фотографии продукта |
| PLACEHOLDER\_NAME | 'placeholder.jpg' | Данная переменная отвечает за название фото, которое будет загружаться при отсутствии в папке фотографии определенного продукта |
| SECRET\_KEY | os.environ.get("SECRET\_KEY") or os.urandom(50) | Необходимо для безопасности сессий на стороне клиента |

Продолжение таблицы 21

|  |  |  |
| --- | --- | --- |
| Глобальная переменная | Значение переменной | Описание |
| REMEMBER\_COOKIE\_DURATION | timedelta(days=7) | Данная переменная отвечает за продолжительность хранения куки-файлов |
| JSON\_AS\_ASCII | False | Необходимо для корректного вывода данных в форме кириллицы в формате json |

# **8** **Разработка структуры дерева каталогов проекта**

Структура дерева каталогов проекта приведена на рисунке 13.

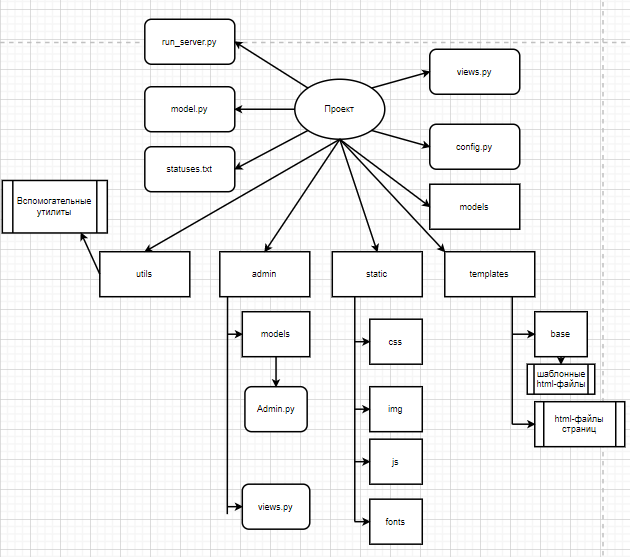


Рисунок 13 — Структура дерева каталогов проекта

# **9 Примеры работы информационной системы**

Пример страницы каталога приведен на рисунке 14.

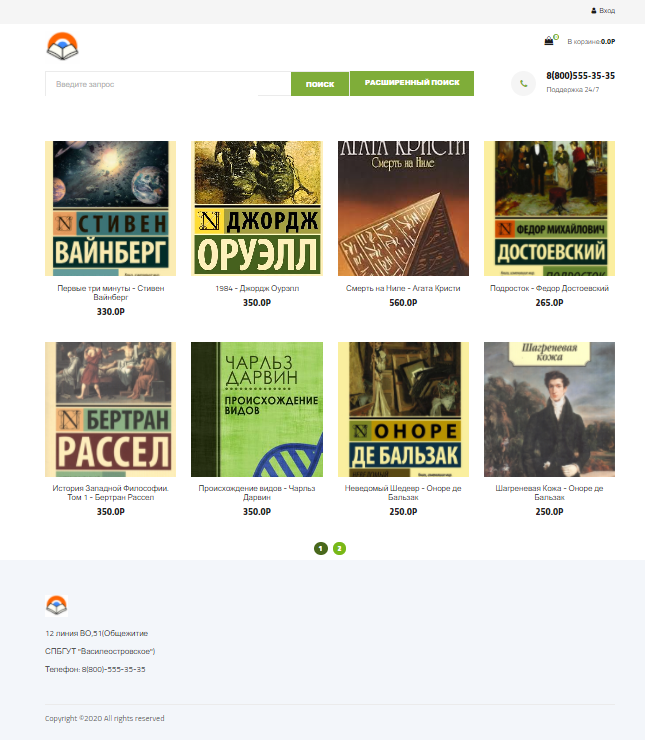
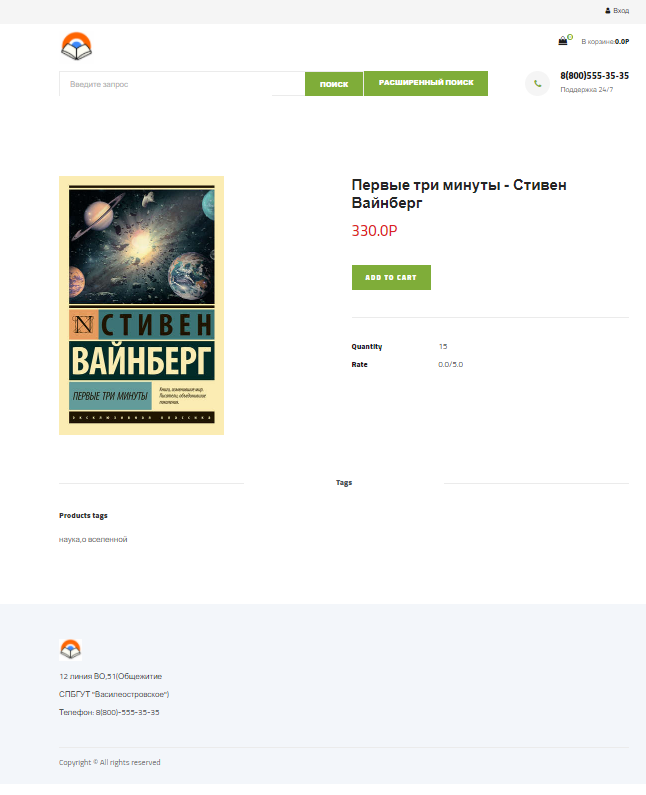


Рисунок 14 — Пример работы сайта

Пример страницы с деталями продукта приведен на рисунке 15.

Рисунок 15 — Пример работы сайта

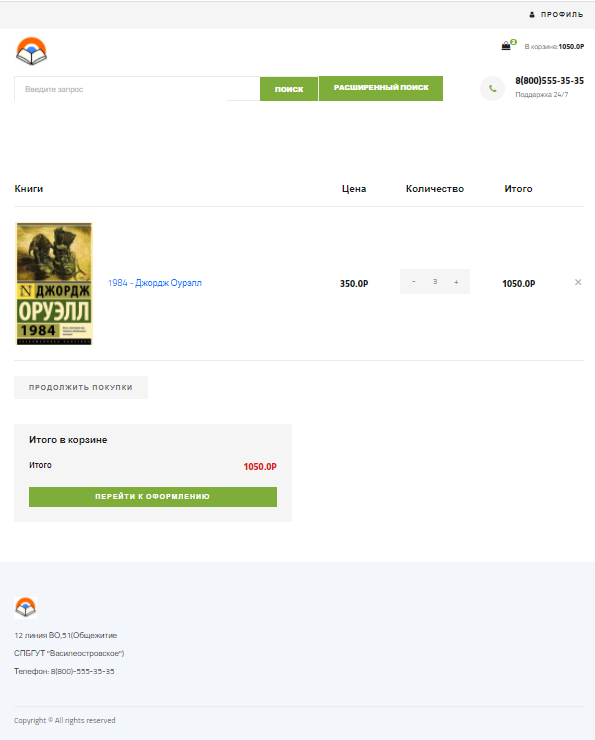
Пример страницы корзины приведен на рисунке 16.

Рисунок 16 — Пример работы сайта

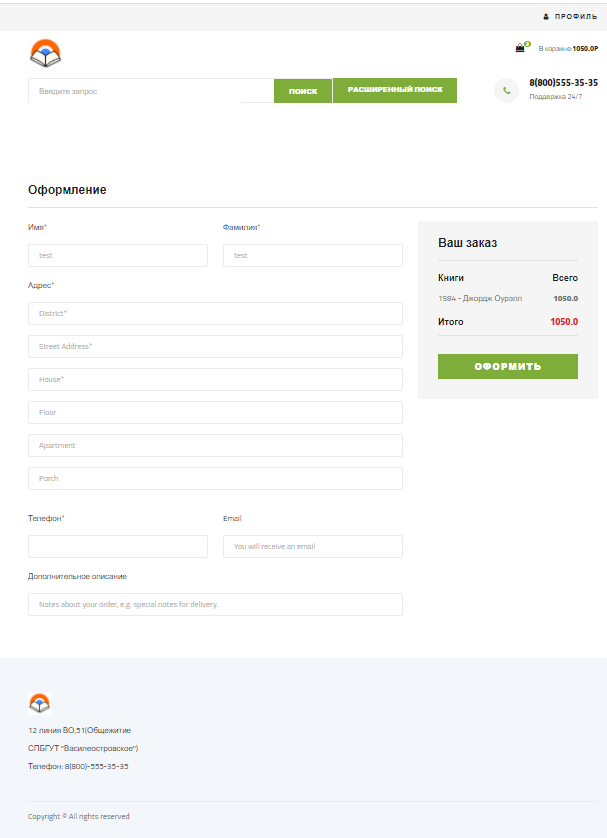
Пример страницы оформления заказа приведен на рисунке 17.

Рисунок 17 — Пример работы сайта

# ЗАКЛЮЧЕНИЕ

Веб-приложения прочно вошли в жизнь человека, что является предсказуемым этапом развития информационного общества. Поэтому среднестатистическому человеку следует знать основы разработки веб-приложений и программирования вообще, потому что это позволяет облегчить и автоматизировать работу в различных сферах жизни, в том числе и бытовых.

В ходе данной курсовой работы была разработана информационная система по продаже книг, реализующая весь необходимый функционал для интернет-магазина, и изучены базовые знания для последующей разработки веб-приложений, а конкретно:

1. была разработана база данных в соответствии со всеми нормальными формами;
2. была разработана структура проекта;
3. была разработана программная часть в соответствии со всеми современными нормами разработки веб-приложений;

В будущем данный проект можно использовать как шаблон для изучения технологий веб-разработки, потому что он использует весь современный на данный момент стек технологий.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Фримен Э., Робсон Э. «Изучаем программирование на JavaScript»-2015.-С. 250-465
2. Flask-login docs[Электронный ресурс].URL: https://flask-login.readthedocs.io/en/latest/ (дата обращения: 10.10.2020).
3. Flask docs[Электронный ресурс].URL:https://flask.palletsprojects.com/en/1.1.x/ (дата обращения: 09.09.2020).
4. Python[Электронный ресурс].URL: https://www.python.org/ (дата обращения: 01.09.2020).
5. Github[Электронный ресурс].URL: https://github.com/ponomaREG/viziProjectShop (дата обращения: 11.11.2020).

# ПРИЛОЖЕНИЕ А

**Разработка авторизации и регистрации**

**А.1 Листинг html-файла login.html**

<!DOCTYPE html>

<html>

   <head>

      <title>Войти</title>

   </head>

   <style type="text/css">

      @import url(https://fonts.googleapis.com/css?family=Roboto:300);

      .login-page {

      width: 360px;

      padding: 8% 0 0;

      margin: auto;

      }

      .form {

      position: relative;

      z-index: 1;

      background: #FFFFFF;

      max-width: 360px;

      margin: 0 auto 100px;

      padding: 35px;

      text-align: center;

      box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);

      }

      .form input {

      font-family: "Roboto", sans-serif;

      outline: 0;

      background: #f2f2f2;

      width: 100%;

      border: 0;

      margin: 0 0 15px;

      padding: 15px;

      box-sizing: border-box;

      font-size: 14px;

      }

      .form button {

      font-family: "Roboto", sans-serif;

      text-transform: uppercase;

      outline: 0;

      background: #4CAF50;

      width: 100%;

      border: 0;

      padding: 15px;

      color: #FFFFFF;

      font-size: 14px;

      -webkit-transition: all 0.3 ease;

      transition: all 0.3 ease;

      cursor: pointer;

      }

      .form button:hover,.form button:active,.form button:focus {

      background: #43A047;

      }

      .form .message {

      margin: 15px 0 0;

      color: #b3b3b3;

      font-size: 12px;

      }

      .form .back {

      color: #b3b3b3;

      font-size: 13px;

      text-align: start;

      }

      .form .back a {

      color: #3ec9a6;

      text-decoration: none;

      }

      .form .error {

      margin: 1px 0 3px;

      color: #EF3B3A;

      font-size: 15px;

      }

      .form .message a {

      color: #4CAF50;

      text-decoration: none;

      }

      .container {

      position: relative;

      z-index: 1;

      max-width: 300px;

      margin: 0 auto;

      }

      .container:before, .container:after {

      content: "";

      display: block;

      clear: both;

      }

      .container .info {

      margin: 50px auto;

      text-align: center;

      }

      .container .info h1 {

      margin: 0 0 15px;

      padding: 0;

      font-size: 36px;

      font-weight: 300;

      color: #1a1a1a;

      }

      .container .info span {

      color: #4d4d4d;

      font-size: 12px;

      }

      .container .info span a {

      color: #000000;

      text-decoration: none;

      }

      .container .info span .fa {

      color: #EF3B3A;

      }

      body {

      background: #76b852; /\* fallback for old browsers \*/

      background: -webkit-linear-gradient(right, #76b852, #8DC26F);

      background: -moz-linear-gradient(right, #76b852, #8DC26F);

      background: -o-linear-gradient(right, #76b852, #8DC26F);

      background: linear-gradient(to left, #76b852, #8DC26F);

      font-family: "Roboto", sans-serif;

      -webkit-font-smoothing: antialiased;

      -moz-osx-font-smoothing: grayscale;

      }

   </style>

   <body>

      <div class="login-page">

         <div class="form">

            <p class="back"><a href="{{ url\_for('showBooks',page=1) }}">&larr;На главную</a></p>

            <form action="{{ url\_for('loginUser')}}" class="login-form" method="POST">

               <input type="text" placeholder="Email" name="email"/>

               <input type="password" placeholder="Пароль" name="password"/>

               {% if error%}

               <p class="error">Error:{{ error }}</p>

               {% endif %}

               <button type="submit">Войти</button>

               <p class="message">Не зарегистрированы? <a href="{{ url\_for('registrationUser') }}">Создать аккаунт</a></p>

            </form>

         </div>

      </div>

   </body>

</html>

**А.2 Листинг html-файла login.html**

<!DOCTYPE html>

<html>

   <head>

      <title>Регистрация</title>

   </head>

   <style type="text/css">

      @import url(https://fonts.googleapis.com/css?family=Roboto:300);

      .login-page {

      width: 360px;

      padding: 8% 0 0;

      margin: auto;

      }

      .form {

      position: relative;

      z-index: 1;

      background: #FFFFFF;

      max-width: 360px;

      margin: 0 auto 100px;

      padding: 45px;

      text-align: center;

      box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);

      }

      .form .back {

      color: #b3b3b3;

      font-size: 13px;

      text-align: start;

      }

      .form .back a {

      color: #3ec9a6;

      text-decoration: none;

      }

      .form input {

      font-family: "Roboto", sans-serif;

      outline: 0;

      background: #f2f2f2;

      width: 100%;

      border: 0;

      margin: 0 0 15px;

      padding: 15px;

      box-sizing: border-box;

      font-size: 14px;

      }

      .form button {

      font-family: "Roboto", sans-serif;

      text-transform: uppercase;

      outline: 0;

      background: #4CAF50;

      width: 100%;

      border: 0;

      padding: 15px;

      color: #FFFFFF;

      font-size: 14px;

      -webkit-transition: all 0.3 ease;

      transition: all 0.3 ease;

      cursor: pointer;

      }

      .form button:hover,.form button:active,.form button:focus {

      background: #43A047;

      }

      .form .message {

      margin: 15px 0 0;

      color: #b3b3b3;

      font-size: 12px;

      }

      .form .message a {

      color: #4CAF50;

      text-decoration: none;

      }

      .form .register-form {

      display: block;

      }

      .form .error {

      margin: 1px 0 3px;

      color: #EF3B3A;

      font-size: 15px;

      }

      .container {

      position: relative;

      z-index: 1;

      max-width: 300px;

      margin: 0 auto;

      }

      .container:before, .container:after {

      content: "";

      display: block;

      clear: both;

      }

      .container .info {

      margin: 50px auto;

      text-align: center;

      }

      .container .info h1 {

      margin: 0 0 15px;

      padding: 0;

      font-size: 36px;

      font-weight: 300;

      color: #1a1a1a;

      }

      .container .info span {

      color: #4d4d4d;

      font-size: 12px;

      }

      .container .info span a {

      color: #000000;

      text-decoration: none;

      }

      .container .info span .fa {

      color: #EF3B3A;

      }

      body {

      background: #76b852;

      background: -webkit-linear-gradient(right, #76b852, #8DC26F);

      background: -moz-linear-gradient(right, #76b852, #8DC26F);

      background: -o-linear-gradient(right, #76b852, #8DC26F);

      background: linear-gradient(to left, #76b852, #8DC26F);

      font-family: "Roboto", sans-serif;

      -webkit-font-smoothing: antialiased;

      -moz-osx-font-smoothing: grayscale;

      }

   </style>

   <body>

      <div class="login-page">

         <div class="form">

            <p class="back"><a href="{{ url\_for('showBooks',page=1) }}">&larr;На главную</a></p>

            <form action="{{ url\_for('registrationUser') }}" class="register-form" method="POST">

               <input type="text" placeholder="Имя" name="firstName" required/>

               <input type="text" placeholder="Фамилия" name="lastName" required/>

               <input type="password" placeholder="Пароль" name="pswd" required/>

               <input type="password" placeholder="Повторите пароль" name="pswd2" required/>

               <input type="text" placeholder="Email" name="email" required/>

               Дата рождения:<input type="date" name="birthDate"/>

               {% if error%}

               <p class="error">Error:{{ error }}</p>

               {% endif %}

               <button type="submit">Создать</button>

               <p class="message">Уже зарегистрированы? <a href="{{ url\_for('loginUser') }}">Войти</a></p>

            </form>

         </div>

      </div>

   </body>

</html>

**А.3 Листинг файла model.py**

from app import db

from utils import md5helper, sqlQueryHelper, dateHelper

from app import login\_manager

from flask\_login import UserMixin

from app.models.User import \*

@login\_manager.user\_loader

def load\_user(userID):

    row = SqlExecuter.getOneRowsPacked('select \* from Покупатель where id = {};'.format(userID))

    if(row is None):

        return None

    user = User(userID = row['id'],email = row['email'],

    password\_hash = row['password\_hash'],last\_name=row['last\_name'],

    first\_name =row['first\_name'] ,birthdate = row['birthdate'])

    if(checkIfUserAdmin(userID)):

        user.set\_level\_of\_access(getLevelOfAccess(userID))

        user.set\_admin(True)

    return user

def checkIfUserAdmin(userID):

    return SqlExecuter.getOneRowsPacked('select \* from Админ where user\_id = {};'.format(userID)) is not None

def getLevelOfAccess(userID):

    if(checkIfUserAdmin(userID)):

        row = SqlExecuter.getOneRowsPacked('select \* from Админ where user\_id = {};'.format(userID))

        return int(row['level\_of\_access'])

# ПРИЛОЖЕНИЕ Б

**Разработка машрутизации**

**Б.1 Листинг файла views.py**

from flask import jsonify, request, redirect, url\_for, render\_template

from app import app

from app import login\_manager

from app.model import load\_user

from app.models.User import User

from app.models.Product import Product

from app.models.Cart import Cart

from app.models.Order import Order

from app.admin.models import \*

from app.admin.views import \*

from utils import pageHelper

import flask\_login

from utils import sqlQueryHelper, tagsHelper

from werkzeug.utils import secure\_filename

from utils.security import Security

import os

@login\_manager.unauthorized\_handler

def unauthorized\_handler():

    return redirect(url\_for('loginUser'))

@app.route("/",methods=['GET'])

def main():

    return redirect(url\_for('showBooksDefault'))

@app.route('/search')

def searchAdvanced():

    availableTags = Product.getAvailableTags()

    return render\_template('advanced-search.html',user=flask\_login.current\_user,tags = availableTags['data'])

@app.route('/checkout',methods=['GET'])

@flask\_login.login\_required

def checkout():

        cart = Cart.getCartOfUser(flask\_login.current\_user.userID) #TODO: CHECK STATUS

        if(cart['status'] == 0):

            return render\_template('checkout.html',

            user = flask\_login.current\_user,

            productsInCart=cart['data'])

        elif(cart['status'] == 2):

            return render\_template('checkout.htlm',

            user = flask\_login.current\_user,

            error = cart['message']

            )

@app.route('/order/<int:orderID>/cancel')

@flask\_login.login\_required

def cancelOrder(orderID):

    cancelOrder = Order.cancelOrder(flask\_login.current\_user.userID,orderID)

    if(cancelOrder['status'] == 0):

        return redirect(url\_for('showDetailsOfOrder',orderID = orderID))

    elif(cancelOrder['status'] == 3 or cancelOrder['status'] == 40):

        return redirect(url\_for('main'))

@app.route('/order/<int:orderID>')

@flask\_login.login\_required

def showDetailsOfOrder(orderID):

    orderDetails = Order.getDetailsOfOrder(flask\_login.current\_user.userID,orderID)

    if(orderDetails['status'] == 0):

        return render\_template('order-page.html',

        orderInfo = orderDetails['info'],

        user = flask\_login.current\_user,

        data = orderDetails['data'],

        address = orderDetails['address'])

    elif(orderDetails['status'] == 3 or orderDetails['status'] == 40):

        return redirect(url\_for('main'))

    else:

        return render\_template('order-page.html',

        user = flask\_login.current\_user,

        error = orderDetails['message'],

        orderDetails = orderDetails)

@app.route('/user/orders',methods=['GET'])

@flask\_login.login\_required

def showOrders():

    orders = Order.getOrdersOfUser(flask\_login.current\_user.userID)

    if(orders['status'] == 0):

        return render\_template('orders-page.html',ordersList = orders['data'],user = flask\_login.current\_user)

    else:

        return render\_template('orders-page.html',error = orders['message'],user = flask\_login.current\_user)

@app.route('/order/new',methods=['GET','POST'])

@flask\_login.login\_required

def newOrder():

        if(request.method == 'POST'):

            district = request.form.get('district')

            street = request.form.get('street')

            flat = request.form.get('flat',type=str,default = '')

            floor = request.form.get('floor',type=str,default = '')

            porch = request.form.get('porch',type=str,default = '')

            house = request.form.get('house')

            email = request.form.get('email',type=str,default='')

            description = request.form.get('desc',type=str,default='')

            phone = request.form.get('phone',type=str,default='')

            newOrder = Order.addNewOrder(flask\_login.current\_user.userID,district,

            flat,house,floor,street,porch,email=email,desc=description,phone=phone)

            if(newOrder['status'] == 0):

                return render\_template('order.html',user = flask\_login.current\_user,data = newOrder['data']['data'])

            else:

                return render\_template('order.html',user = flask\_login.current\_user,error = newOrder['message'])

        else:

            return redirect(url\_for('main'))

@app.route('/cart',methods=['GET'])

@flask\_login.login\_required

def cart():

        cart = Cart.getCartOfUser(flask\_login.current\_user.userID) #TODO: CHECK STATUS

        if(cart['status'] == 0):

            print('len ' + str(len(cart['data'])))

            return render\_template('shoping-cart.html',

            user = flask\_login.current\_user,

            productsInCart = cart['data'])

        else:

            return render\_template('shoping-cart.html',

            user = flask\_login.current\_user,

            error = cart['message'])

@app.route("/login",methods = ['GET','POST'])

def loginUser():

    if(flask\_login.current\_user.is\_authenticated): #Проверяем вошел ли уже пользователь

        return redirect(url\_for('userInfo')) #Перекидываем на страницу профиля

    if(request.method == 'POST'): # Если метод обращения к url POST

        email = Security.escape\_sql(request.form.get('email',type=str)) # Получаем введенный email

        password = Security.escape\_sql(request.form.get('password',type=str)) # Получаем введенный пароль

        userID = User.validateUserAndReturnUserID(email,password) #Получаем ID пользователя по введенному email и паролю

        if(userID != -1):# Если пользователь найден

            flask\_login.login\_user(load\_user(userID),remember=True)#Логиним пользователя в системе

            return redirect(url\_for('showBooks',page=1))# Перекидываем на страницу книг

        else:

            return render\_template('login.html',error = "Not found")#Выводим страницу логина с ошибкой

    else:

        return render\_template('login.html')# Если метод обращения не POST , то выводим html страницу логина

@app.route('/registration',methods=['GET','POST'])

def registrationUser():

    if(flask\_login.current\_user.is\_authenticated):

        return redirect(url\_for('userInfo'))

    if(request.method == 'POST'):

        email = Security.escape\_sql(request.form.get('email',type=str)) # Получаем введенный email пользователя

        pswd = Security.escape\_sql(request.form.get('pswd',type=str)) # Получаем введенный пароль пользователя

        pswd2 = Security.escape\_sql(request.form.get('pswd2',type=str)) # Получаем введенный 2 пароль пользователя

        first\_name = Security.escape\_sql(request.form.get('firstName',type=str)) # Получаем введенныое имя пользователя

        last\_name = Security.escape\_sql(request.form.get('lastName',type=str)) # Получаем введенную фамилию пользователя

        birthdate = Security.escape\_sql(request.form.get('birthDate'))# Получаем введенную дата рождения пользователя

        if(pswd != pswd2):

            return render\_template('registration.html',error = 'Password mismatch') # Возвращаем html с ошибкой

        resultRegisterOperation = User.registerUser(email,pswd,last\_name,first\_name,birthdate) # Создаем пользователя

        if(resultRegisterOperation["status"] == 8):

            return render\_template('registration.html',error = 'User already exists') # Возвращаем html с ошибкой

        elif(resultRegisterOperation['status'] == 7):

            return render\_template('registration.html',error = 'Incorrect email') # Возвращаем html с ошибкой

        flask\_login.login\_user(load\_user(resultRegisterOperation["userID"]),remember=True) # Авторизируем пользователя

        return redirect(url\_for('userInfo'))

    else:

        return render\_template('registration.html')# Возвращаем html с формой

@app.route('/user',methods=['GET'])

@flask\_login.login\_required # Пример декоратора

def userInfo():

        return render\_template('user-profile.html',user = flask\_login.current\_user)

@app.route('/logout',methods=['GET'])

@flask\_login.login\_required

def logoutUser():

    flask\_login.logout\_user()

    return redirect(url\_for('main'))

@app.route('/books',methods = ['GET'])

def showBooksDefault():

    return showBooks(1)

@app.route('/books/<int:page>',methods=["GET"])

def showBooks(page):

    if(page < 1):

        return render\_template('books.html',error = 'Incorrect page',user = flask\_login.current\_user)

    OFFSET = 8

    userQuerySearch = request.args.get('q',default=None,type=str)

    tagsFilter = request.args.getlist('tags')

    tagsFilterStr = None

    if(userQuerySearch is not None):

        userQuerySearch = Security.escape\_sql(userQuerySearch)

        result = Product.getAllProfuctsFilteredByQuery(userQuerySearch,page,OFFSET)

        countOfRows = Product.getQuantityOfRowsInTable(

            'select count(\*) as "count" from Товар where title like "%{0}% " \

            or author like "%{0}%" order by rate;'.format(userQuerySearch))['count']

    elif(tagsFilter):

        countOfRows = Product.getQuantityOfRowsInTable(sqlQueryHelper.buildSqlQueryByTags('select count(\*) as "count" from Товар',tagsFilter))['count']

        result = Product.getAllProductsFilteredByTags(tagsFilter,page,OFFSET)

        tagsFilterStr = tagsHelper.makeArrayOfTagsToStr(tagsFilter)

        tagsFilterStr = Security.escape\_sql(tagsFilterStr)

    else:

        countOfRows = Product.getQuantityOfRowsInTable('select count(\*) as "count" from Товар;')['count']

        result = Product.getAllProfuctsFilteredById(page,OFFSET)

    if(result['status'] == 0):

        countOfPages = countOfRows // OFFSET

        if(countOfRows % OFFSET > 0):

            countOfPages += 1

        countOfPagesRange = pageHelper.getRangeOfPages(countOfPages,page)

        return render\_template(

            'books.html', #TODO : IF STATUS

            products = result['data'],

            countOfPagesRange = countOfPagesRange,

            user = flask\_login.current\_user,

            q = userQuerySearch,

            currentPage=page,

            tagsAlreadySearched = tagsFilterStr)

    elif(result['status'] == 2):

        return render\_template('books.html',

        error = 'Not found',

        user = flask\_login.current\_user,

        q = userQuerySearch,

        tagsAlreadySearched = tagsFilterStr)

    elif(result['status'] == 1):

        return render\_template('books.html',

        error = 'SQL runtime error',

        user = flask\_login.current\_user)

    return render\_template('books.html',error='ERROR',user = flask\_login.current\_user)

@app.route('/books/details/<int:productID>')

def showDetailsOfBook(productID):

    details = Product.getDetailsOfProduct(productID)

    if(details['status'] == 0):

        details['data'][0]['rate']= Product.getRateOfProduct(productID)

        if(flask\_login.current\_user.is\_authenticated):

            return render\_template('shop-details.html',

            user = flask\_login.current\_user,

            details = details['data'][0],

            userMark = flask\_login.current\_user.getUserMarkTo(productID),

            quantityInCart = Cart.getQuantityOfProductInCart(flask\_login.current\_user.userID,productID))

        else:

            return render\_template('shop-details.html',

            user = flask\_login.current\_user,

            details = details['data'][0])

    elif(details['status'] == 2):

        return render\_template('shop-details.html',

        user = flask\_login.current\_user,error = details['message'])

    elif(details['status'] == 1): #TODO: ERROR

        return render\_template('shop-details.html',

        user = flask\_login.current\_user,error = details['message'])

    else:

        return redirect(url\_for('showBooks',page=1))

@app.errorhandler(404)

def page\_not\_found(e):

    return redirect(url\_for('main'))

**Б.2 Листинг файла admin/views.py**  
  
from flask import request,redirect,url\_for,render\_template

from app import app

from app.admin.models.Admin import Admin

import flask\_login

import os

from werkzeug.utils import secure\_filename

@app.route('/admin',methods=['GET','POST'])

def loginAdmin():

    if(flask\_login.current\_user.is\_authenticated): #Проверяем вошел ли уже пользователь

        if(flask\_login.current\_user.is\_admin):

            return render\_template('admin-page.html') #Перекидываем на страницу профиля

        else:

            return redirect(url\_for('main'))

    else:

        return redirect(url\_for('loginUser'))

@app.route('/admin/orders',methods=['GET','POST'])

@flask\_login.login\_required

def adminManageOrders():

    if(flask\_login.current\_user.is\_admin):

         if(request.method == 'POST'):

            method = request.form.get('method',default=1,type=int)

            if(method == 1 or method == 2):

                date\_b = request.form.get('date\_b')

                date\_e = request.form.get('date\_e')

                resultStat = None

                if(method == 1):

                    resultStat = Admin.getAllOrdersByPeriod(date\_b,date\_e,flask\_login.current\_user.userID)

                elif(method == 2):

                    resultStat = Admin.getCountOfOrderByPeriod(date\_b,date\_e,flask\_login.current\_user.userID)

                else:

                    return render\_template('admin-manage-orders.html',message = 'Incorrect method')

                if resultStat is not None:

                    if(resultStat['status'] == 3):

                        return render\_template('admin-manage-orders.html',date\_e = date\_e,date\_b = date\_b,message = resultStat['message'])

                    elif(resultStat['status'] == 0):

                        return render\_template('admin-manage-orders.html',date\_e = date\_e,date\_b=date\_b,resultOfResponse = resultStat)

                else:

                    return render\_template('admin-manage-orders.html',date\_e = date\_e,date\_b = date\_b)

            elif(method == 3):

                orderID = request.form.get('orderID-3',type = int)

                resultStat = Admin.getInfoOfOrderBy(orderID,flask\_login.current\_user.userID)

                if(resultStat['status'] == 0):

                    return render\_template('admin-manage-orders.html',resultOfResponse = resultStat)

                elif(resultStat['status'] == 3):

                    return render\_template('admin-manage-orders.html',message = resultStat['message'])

            elif(method == 4):

                status = request.form.get('status',type = int)

                resultOfResponse = Admin.getOrdersByStatus(status,flask\_login.current\_user.userID)

                if(resultOfResponse['status'] == 0):

                    return render\_template('admin-manage-orders.html',resultOfResponse = resultOfResponse)

                elif(resultOfResponse['status'] == 3):

                    return render\_template('admin-manage-orders.html',message = resultOfResponse['message'])

            elif(method == 5):

                newStatus = request.form.get('newStatus',type = int)

                orderID = request.form.get('orderID-5',type=int)

                if(Admin.is\_admin\_can\_write(flask\_login.current\_user.userID)):

                    resultOfResponse = Admin.setNewStatusOfOrder(newStatus,orderID,flask\_login.current\_user.userID)

                    return render\_template('admin-manage-orders.html',resultOfResponse = resultOfResponse)

                return render\_template('admin-manage-orders.html',message = 'Access denied!')

            elif(method == 6):

                supplierName = request.form.get('supplier-name')

                if(Admin.is\_admin\_can\_write(flask\_login.current\_user.userID)):

                    resultOfResponse = Admin.addNewSupplier(supplierName,flask\_login.current\_user.userID)

                    return render\_template('admin-manage-orders.html',resultOfResponse = resultOfResponse)

                return render\_template('admin-manage-orders.html',message = 'Access denied!')

         else:

            return render\_template('admin-manage-orders.html')

    else:

        return redirect(url\_for('main'))

@app.route('/admin/stat',methods=['GET','POST'])

@flask\_login.login\_required

def adminStat():

    if(flask\_login.current\_user.is\_admin):

        if(request.method == 'POST'):

            date\_b = request.form.get('date\_b')

            date\_e = request.form.get('date\_e')

            method = request.form.get('method',default=1,type=int)

            resultStat = None

            if(method == 1):

                resultStat=Admin.getAllIncomeByPeriod(date\_b,date\_e,flask\_login.current\_user.userID)

            elif(method == 2):

                resultStat = Admin.getRatingPopularityOfBooksByPeriod(date\_b,date\_e,flask\_login.current\_user.userID)

            else:

                return render\_template('admin-stat.html',message = 'Incorrect method')

            if resultStat is not None:

                if(resultStat['status'] == 3):

                    return render\_template('admin-stat.html',date\_e = date\_e,date\_b = date\_b,message = resultStat['message'])

                elif(resultStat['status'] == 0):

                    return render\_template('admin-stat.html',date\_e = date\_e,date\_b=date\_b,resultOfResponse = resultStat)

            else:

                return render\_template('admin-stat.html',date\_e = date\_e,date\_b = date\_b)

        else:

            return render\_template('admin-stat.html')

    else:

        return redirect(url\_for('loginUser'))

@app.route('/admin/product',methods=['GET','POST'])

@flask\_login.login\_required

def adminGetProductInfo():

     if(flask\_login.current\_user.is\_admin):

         columnNames = Admin.getColumnsOfTable('Товар')

         suppliers = Admin.getSuppliers()

         if(request.method == 'GET'):

             return render\_template('admin-product.html',columnNames = columnNames['data'],suppliers=suppliers)

         else:

             method = request.form.get('method',type=int)

             if(method == 1):

                column = request.form.get('column')

                value = request.form.get('value')

                resultOfResponseToDB = Admin.getInfoOfBookBy(column,value,flask\_login.current\_user.userID)

             elif(method == 2):

                quantity = request.form.get('quantity',type = int)

                productID = request.form.get('productID-2',type = int)

                if(Admin.is\_admin\_can\_write(flask\_login.current\_user.userID)):

                    resultOfResponseToDB = Admin.setNewValueBook(productID,"quantity",quantity,flask\_login.current\_user.userID)

                else:

                     resultOfResponseToDB = {}

                     resultOfResponseToDB['status'] = 131

             elif(method == 3):

                 desc = request.form.get('desc')

                 productID = request.form.get('productID-3',type=int)

                 if(Admin.is\_admin\_can\_write(flask\_login.current\_user.userID)):

                    resultOfResponseToDB = Admin.setNewValueBook(productID,"description",desc,flask\_login.current\_user.userID)

                 else:

                     resultOfResponseToDB = {}

                     resultOfResponseToDB['status'] = 131

             elif(method == 4):

                 price = request.form.get('price',type=float)

                 productID = request.form.get('productID-4',type=int)

                 if(Admin.is\_admin\_can\_write(flask\_login.current\_user.userID)):

                    resultOfResponseToDB = Admin.setNewValueBook(productID,'cost\_sale',price,flask\_login.current\_user.userID)

                 else:

                     resultOfResponseToDB = {}

                     resultOfResponseToDB['status'] = 131

             elif(method == 6):

                 tags = request.form.get('tags',type=str)

                 productID = request.form.get('productID-6',type=int)

                 if(Admin.is\_admin\_can\_write(flask\_login.current\_user.userID)):

                    resultOfResponseToDB = Admin.setNewValueBook(productID,'tags',tags,flask\_login.current\_user.userID)

                 else:

                     resultOfResponseToDB = {}

                     resultOfResponseToDB['status'] = 131

             elif(method == 5):

                 productID = request.form.get('productID-5',type=int)

                 if 'image' in request.files:

                    file = request.files['image']

                    if(file.filename != ''):

                        filename = secure\_filename(file.filename)

                        filePath = os.path.join(app.config['UPLOAD\_FOLDER'],filename)

                        file.save(filePath)

                        if(Admin.is\_admin\_can\_write(flask\_login.current\_user.userID)):

                            resultOfResponseToDB = Admin.setNewValueBook(productID,'imageLink',filename,flask\_login.current\_user.userID)

                        else:

                            resultOfResponseToDB = {}

                            resultOfResponseToDB['status'] = 131

                    else:

                        return render\_template('admin-product.html',columnNames = columnNames['data'],message = 'Image not uploaded 1',suppliers=suppliers)

                 else:

                     return render\_template('admin-product.html',columnNames = columnNames['data'],message = 'Image not uploaded 2',suppliers=suppliers)

             elif(method == 7):

                 productID = request.form.get('productID-7',type=int)

                 quantity = request.form.get('quantity-delivery',type=int)

                 supplierID = request.form.get('supplierID',type=int)

                 cost\_purchase = request.form.get('cost-purchase-delivery',type=float)

                 if(Admin.is\_admin\_can\_write(flask\_login.current\_user.userID)):

                    resultOfResponseToDB = Admin.addNewDeliveryOfProducts(productID,quantity,cost\_purchase,supplierID,flask\_login.current\_user.userID)

                 else:

                            resultOfResponseToDB = {}

                            resultOfResponseToDB['status'] = 131

             else:

                 return render\_template('admin-product.html',columnNames = columnNames['data'],message = 'What???Method:{}'.format(method),suppliers=suppliers)

             if(resultOfResponseToDB['status'] == 0):

                 return render\_template('admin-product.html',resultOfResponse = resultOfResponseToDB,columnNames = columnNames['data'],suppliers=suppliers)

             elif(resultOfResponseToDB['status'] == 3):

                 return render\_template('admin-product.html',message = 'Empty',columnNames = columnNames['data'],suppliers=suppliers)

             elif(resultOfResponseToDB['status'] == 130):

                 return render\_template('admin-product.html',message = resultOfResponseToDB['message'],columnNames = columnNames['data'],suppliers=suppliers)

             elif(resultOfResponseToDB['status'] == 131):

                 return render\_template('admin-product.html',message = 'ACCESS DENIED!',columnNames = columnNames['data'],suppliers=suppliers)

@app.route('/admin/add',methods=['GET','POST'])

@flask\_login.login\_required

def adminAddNew():

    if(flask\_login.current\_user.is\_admin):

        if(request.method == 'GET'):

            return render\_template('admin-add.html')

        else:

            title = request.form.get('title')

            author = request.form.get('author')

            desc = request.form.get('desc')

            cost\_purchase = request.form.get('cost\_purchase',type = float)

            cost\_sale = request.form.get('cost\_sale',type = float)

            quantity = request.form.get('quantity',type=int)

            tags = request.form.get('tags')

            filename = app.config['PLACEHOLDER\_NAME']

            if 'image' in request.files:

                file = request.files['image']

                if(file.filename != ''):

                    filename = secure\_filename(file.filename)

                    filePath = os.path.join(app.config['UPLOAD\_FOLDER'],filename)

                    file.save(filePath)

            if(Admin.is\_admin\_can\_write(flask\_login.current\_user.userID)):

                result = Admin.insertNewBook(title,author,desc,cost\_sale,cost\_purchase,quantity,filename,tags,flask\_login.current\_user.userID)

            else:

                return render\_template('admin-add.html',message = 'Access Denied!')

            if(result['status'] == 0):

                return render\_template('admin-add.html',message = 'Added!',productID = result['data'][0])

            else:

                return render\_template('admin-add.html',message = 'No added(')

    else:

        return redirect(url\_for('loginUser'))

# ПРИЛОЖЕНИЕ В

**Разработка маршрутизации**

**В.1 Листинг файла слоя обработки данных Address.py**

from app import db

from app.models.SqlExecuter import SqlExecuter

class Address:

    @staticmethod

    def addNewAddress(district,house,floor,flat,porch,street):

        lastrowid = SqlExecuter.executeModif('insert into Адрес(`district`,`house`,`floor`,`flat`,`porch`,`street`) values("{}","{}","{}","{}","{}","{}");'.format(district,house,floor,flat,porch,street))

        return lastrowid

**В.2 Листинг файла слоя обработки данных Cart.py**

from app import db

from utils import imageHelper

from app.models.SqlExecuter import SqlExecuter

class Cart:

    @staticmethod

    def getQuantityOfProductInCart(userID,productID):

        result = {}

        row = SqlExecuter.getOneRowsPacked('select count from Корзина where user\_id = {} and product\_id = {};'

        .format(userID,productID))

        print("QUANTITY:" + str(row))

        if(row is None):

            result['status'] = 2

            result['message'] = 'Product doesnt exist in cart'

            result['data'] = {'count':0}

            return result

        result['status'] = 0

        result['message'] = 'OK'

        result['data'] = {'count':row['count']}

        return result

    @staticmethod

    def countTotalCostOfUser(userID):

        row = SqlExecuter.getOneRowsPacked('select SUM(cart.count\*pr.cost\_sale) as "totalCost" \

                from Товар as pr \

                inner join Корзина as cart on pr.id = product\_id \

                and user\_id = {};'.format(userID))

        if(row is None or row['totalCost'] is None):

            return 0.0

        else:

            return float(row['totalCost'])

    @staticmethod

    def removeItemInCartOfUser(userID,productID):

        result = {}

        try:

            row = SqlExecuter.getOneRowsPacked(

                'select \* from Корзина where product\_id = {} and user\_id = {};'.format(productID,userID)

            )

        except IndexError:

            result['status'] = 1

            result['message'] = 'SQL Runtime error'

            return result

        if(row is not None):

            if(row['count']>1):

                try:

                    SqlExecuter.executeModif('update Корзина set count = count - 1 where \

                    product\_id = {} and user\_id = {};'.format(productID,userID))

                except IndexError:

                    result['status'] = 1

                    result['message'] = 'SQL Runtime error'

                    return result

                result['status'] = 0

                result['message'] = 'OK'

                return result

            try:

                SqlExecuter.executeModif('delete from Корзина where \

                    product\_id = {} and user\_id = {};'.format(productID,userID))

            except IndexError:

                result['status'] = 1

                result['message'] = 'SQL Runtime error'

                return result

            result['status'] = 0

            result['message'] = 'OK'

            return result

        result['status'] = 2

        result['message'] = 'Not found item'

        return result

    @staticmethod

    def addItemInCartOfUser(userID,productID):

        result = {}

        print(productID)

        row = SqlExecuter.getOneRowsPacked(

            'select \* from Товар where id = {};'.format(productID)

        )

        if(row is None):

            result['status'] = 2

            result['message'] = 'No such product'

            return result

        if(row['quantity'] == 0):

            result['status'] = 4

            result['message'] = 'Zero quantity of called product'

            return result

        quantityOfProduct = row['quantity']

        try:

            row = SqlExecuter.getOneRowsPacked(

                'select \* from Корзина where product\_id = {} and user\_id = {};'.format(productID,userID)

            )

        except:

            result['status'] = 1

            result['message'] = 'SQL Runtime error'

            return result

        if(row is not None):

            if(row['count'] >= quantityOfProduct):

                result['status'] = 5

                result['message'] = 'Not enough products'

                return result

            try:

                SqlExecuter.executeModif(

                    'update Корзина set count = count + 1 \

                    where product\_id = {} and user\_id = {}'.format(productID,userID))

            except:

                result['status'] = 1

                result['message'] = 'SQL Runtime error'

                return result

            result['status'] = 0

            result['message'] = 'Inc count'

            return result

        try:

            SqlExecuter.executeModif('insert into Корзина values({},{},1)'.format(userID,productID))

        except:

            result['status'] = 1

            result['message'] = 'SQL Runtime error'

            return result

        result['status'] = 0

        result['message'] = 'Add new values'

        return result

    @staticmethod

    def getCountOfItemsInCart(userID):

        result = {}

        try:

            row = SqlExecuter.getOneRowsPacked(

                'select SUM(count) as "count" from Корзина where user\_id = {};'.format(userID))

            if(row is None or row['count'] is None):

                result['status'] = 2

                result['message'] = 'Empty cart'

                result['data'] = [0]

                return result

            # if(row['count'] is None):

            #     result['status'] = 2

            #     result['message'] = 'Empty cart'

            #     result['data'] = [0]

            #     return result

            result['status'] = 0

            result['message'] = 'OK'

            result['data'] = [row['count']]

            return result

        except:

            result['status'] = 1

            result['message'] = 'SQL runtime error'

            result['data'] = [-1]

            return result

    @staticmethod

    def getCartOfUser(userID):

        result = {}

        try:

            data = SqlExecuter.getAllRowsPacked('select concat(pr.book\_title," - ",pr.author) as "title",pr.cost\_sale as "cost",cart.count,cart.count\*pr.cost\_sale as "total",pr.id,pr.imageLink,pr.author from Товар as pr inner join Корзина as cart on pr.id = product\_id and user\_id = {};'.format(userID))

        except:

            result['status'] = 1

            result['message'] = 'SQL runtime error'

            result['data'] = []

            return result

        if(data is None or len(data) == 0):

            result['status'] = 2

            result['message'] = 'Empty cart'

            result['data'] = []

            return result

        for row in data:

            row['imageLink'] = imageHelper.makeFullPathToImage(row['imageLink'])

        result['data'] = data

        result['count'] = len(data)

        result['status'] = 0

        result['message'] = 'OK'

        result['totalCost'] = SqlExecuter.getOneRowsPacked("select SUM(cart.count\*pr.cost\_sale) as 'totalCost' from Товар as pr inner join Корзина as cart on pr.id = product\_id and user\_id = {};".format(userID))['totalCost']

        return result

**В.3 Листинг файла слоя обработки данных Order.py**

from app.models.Cart import Cart

from app.models.Address import Address

from app.models.Product import Product

from app.models.SqlExecuter import SqlExecuter

from utils import imageHelper

from utils import emailSender

from utils import addressBuilder

class Order:

    @staticmethod

    def getOrdersOfUser(userID):

        result = {}

        allRows = SqlExecuter.getAllRowsPacked('select ord.\*,addr.district,addr.street,addr.flat,addr.floor,\

        addr.porch,addr.house,address\_id as "address" \

        from Заказ as ord inner join Адрес as addr\

         where ord.address\_id = addr.id and user\_id = {} order by id DESC;'.format(userID))

        if(allRows is None or len(allRows) == 0):

            result['status'] = 2

            result['message'] = 'Empty response'

            result['data'] = []

            return result

        query = 'select SUM(price\*count) as "total" from Забронированная\_книга where order\_id = {};'

        for row in allRows:

            rowWithTotalOfOrder = SqlExecuter.getOneRowsPacked(query.format(row['id']))

            row['total'] = rowWithTotalOfOrder['total']

        result['status'] = 0

        result['message'] = 'OK'

        result['data'] = allRows

        return result

    @staticmethod

    def cancelOrder(userID,orderID):

        result = {}

        if(not Order.checkIfUserHaveOrderWith(userID,orderID)):

            result['status'] = 3

            result['message'] = 'Why are you so curious?'

            result['data'] = []

            return result

        if(not Order.checkIfOrderExists(orderID)):

            result['status'] = 40

            result['message'] = 'Not founded'

            result['data'] = []

            return result

        query = 'update Заказ set status = 4 where id = {};'.format(orderID)

        SqlExecuter.executeModif(query)

        result['status'] = 0

        result['message'] = 'OK'

        result['data'] = []

        return result

    @staticmethod

    def getDetailsOfOrder(userID,orderID):

        result = {}

        if(not Order.checkIfUserHaveOrderWith(userID,orderID)):

            result['status'] = 3

            result['message'] = 'Why are you so curious?'

            result['data'] = []

            return result

        if(not Order.checkIfOrderExists(orderID)):

            result['status'] = 40

            result['message'] = 'Not founded'

            result['data'] = []

            return result

        data = SqlExecuter.getAllRowsPacked("select pr.id,pr.imageLink,pr.book\_title,bk.count,bk.price\*bk.count as 'total' from Забронированная\_книга as bk inner join Товар as pr on bk.order\_id = {} and pr.id = bk.product\_id;".format(orderID))

        for row in data:

            row['imageLink'] = imageHelper.makeFullPathToImage(row['imageLink'])

        rowAddressPacked = SqlExecuter.getOneRowsPacked(

            "select ord.id as 'orderID',addr.\*,ord.date,ord.status,ord.total,sum(book.rpice\*book.price) as 'total',concat(addr.district,' district,',addr.street,', ',addr.house) as 'address' from Заказ as ord  inner join Адрес as addr inner join Забронированная\_книга as book \

            where ord.address\_id = addr.id and ord.id = {};".format(orderID))

        result['address'] = rowAddressPacked

        result['info'] = {'orderID':rowAddressPacked['orderID'],'date':rowAddressPacked['date'],'status':rowAddressPacked['status'],

                            'total':rowAddressPacked['total']}

        result['status'] = 0

        result['message'] = 'OK'

        result['data'] = data

        return result

    @staticmethod

    def addNewOrder(userID,district,flat,house,floor,street,porch='',email='',desc = '',phone=''):

        result = {}

        try:

            quantityInCart = Cart.getCountOfItemsInCart(userID)

            if(quantityInCart['status'] == 2):

                result['status'] = 2

                result['message'] = 'Empty cart'

                result['data'] = []

                return result

        except IndexError:

            result['status'] = 1

            result['message'] = 'SQL runtime error'

            result['data'] = []

            return result

        cartOfUser = Cart.getCartOfUser(userID)['data']

        isError = False

        for row in cartOfUser:

            product = SqlExecuter.getOneRowsPacked('select \* from Товар where id = {};'.format(row['id']))

            if(product['quantity'] < row['count']):

                isError = True

                SqlExecuter.executeModif('delete from Корзина where product\_id = {} and user\_id = {};'.format(row['id'],userID))

        if(isError):

                result['status'] = 10

                result['message'] = 'Attempt to order product with incorrect quantity'

                return result

        try:

            lastrowidAddress = Address.addNewAddress(district,house,floor,flat,porch,street)

        except IndexError:

                result['status'] = 1

                result['message'] = 'SQL runtime error'

                result['data'] = []

                return result

        try:

            lastrowid = SqlExecuter.executeModif('insert into Заказ(`user\_id`,`status`,`address\_id`,`description`,`email`,`phone`) values({},{},{},"{}","{}","{}");'.format(

                    userID,

                    0,

                    lastrowidAddress,

                    desc,

                    email,

                    phone))

        except IndexError:

            result['status'] = 1

            result['message'] = 'SQL runtime error'

            result['data'] = []

            return result

        data = {'id':lastrowid,"data":cartOfUser}

        for row in cartOfUser:

            try:

                SqlExecuter.executeModif('insert into Забронированная\_книга \

                    values({},{},{},{})'.format(row['id'],row['count'],lastrowid,row['cost']))

                SqlExecuter.executeModif('delete from Корзина where \

                    user\_id = {} and product\_id = {};'.format(userID,row['id']))

                SqlExecuter.executeModif('update Товар set quantity = quantity - {} where id = {};'.format(row['count'],row['id']))

            except IndexError:

                result['status'] = 1

                result['message'] = 'SQL runtime error'

                result['data'] = []

                return result

        result['status'] = 0

        result['message'] = 'OK'

        result['data'] = data

        # if(email is not None):

        #     emailSender.EmailSender.sendEmailTo([email],orderDetails=result['data'])

        return result

    @staticmethod

    def checkIfUserHaveOrderWith(userID,orderID):

        return SqlExecuter.getOneRowsPacked('select \* from Заказ where user\_id = {} and id = {};'.format(userID,orderID)) is not None

    @staticmethod

    def checkIfOrderExists(orderID):

        return SqlExecuter.getOneRowsPacked('select \* from Заказ where id = {} and status != 4 and status != 5;'.format(orderID)) is not None

**В.4 Листинг файла слоя обработки данных Product.py**

from app import db

from utils import sqlQueryHelper

from utils import imageHelper

from utils import tagsHelper

from app.models.SqlExecuter import SqlExecuter

class Product:

    id = -1

    title = None

    cost\_sale = -1.0

    quantity = -1

    def \_\_init\_\_(self,title,cost\_sale,quantity,id):

        self.id = id

        self.title = title

        self.cost\_sale = cost\_sale

        self.quantity = quantity

    @staticmethod

    def \_\_preparePackedProducts(allRows):

        result = {}

        if(allRows is None or len(allRows) == 0):

            result['status'] = 2

            result['message'] = "Empty data"

            result['data'] = []

            return result

        for row in allRows:

            row['imageLink'] = imageHelper.makeFullPathToImage(row['imageLink'])

        result['data'] = allRows

        result['status'] = 0

        result['message'] = 'OK'

        return result

    @staticmethod

    def \_\_prepareProducts(cursor):

        allRows = cursor.fetchall()

        result = {}

        if(len(allRows) == 0):

            result['status'] = 2

            result['message'] = "Empty data"

            result['data'] = []

            cursor.close()

            return result

        result['data'] = []

        for rw in allRows:

            rowDict = {

                'title':rw[9]+' - '+rw[1],'desc':rw[2],

                'rate':rw[7],'cost':rw[4],

                'quantity':rw[5],'tags':rw[6],

                'id':rw[0],'imageLink':imageHelper.makeFullPathToImage(rw[8]),

                'author':rw[9],'bookTitle':rw[1]

                }

            result['data'].append(rowDict)

        result['status'] = 0

        result['count'] = len(allRows)

        result['message'] = "OK"

        cursor.close()

        return result

    @staticmethod

    def getQuantityOfRowsInTable(query):

        result = {}

        try:

            row = SqlExecuter.getOneRowsPacked(query)

        except:

            result['status'] = 1

            result['message'] = "Runtime error while executing sql query"

            result['data'] = []

            return result

        if(row is not None):

            result['status'] = 0

            result['message'] = 'OK'

            result['count'] = row['count']

        else:

            result['status'] = 2

            result['message'] = 'Not founded'

            result['count'] = 0

        return result

    @staticmethod

    def getAllProfuctsFilteredById(page,offset):

        result = {}

        try:

            allRows = SqlExecuter.getAllRowsPacked(

                'select \*,cost\_sale as "cost",concat(book\_title," - ",author) as "title" from Товар order by id DESC LIMIT {} OFFSET {};'

                .format(offset,offset\*(page-1)))

        except:

            result['status'] = 1

            result['message'] = "Runtime error while executing sql query"

            result['data'] = []

            return result

        return Product.\_\_preparePackedProducts(allRows)

    @staticmethod

    def getAllProfuctsFilteredByQuery(query,page=-1,offset=-1):

        result = {}

        try:

            allRows = SqlExecuter.getAllRowsPacked(

                'select \*,cost\_sale as "cost",concat(book\_title," - ",author) as "title" from Товар where book\_title like "%{0}%" or author like "%{0}%" order by id DESC LIMIT {1} OFFSET {2};'

                .format(query,offset,offset\*(page-1)))

        except:

            result['status'] = 1

            result['message'] = "Runtime error while executing sql query"

            result['data'] = []

            return result

        return Product.\_\_preparePackedProducts(allRows)

    @staticmethod

    def getAllProductsFilteredByTags(tags,page,offset):

        result = {}

        try:

            allRows = SqlExecuter.getAllRowsPacked(sqlQueryHelper.buildSqlQueryByTagsAndPage('select \*,cost\_sale as "cost",concat(book\_title," - ",author) as "title" from Товар',tags,page,offset))

        except:

            result['status'] = 1

            result['message'] = "Runtime error while executing sql query"

            result['data'] = []

            return result

        return Product.\_\_preparePackedProducts(allRows)

    @staticmethod

    def getAvailableTags():

        result = {}

        allRows = SqlExecuter.getAllRowsPacked('select tags from Товар;')

        if(allRows is None or len(allRows) == 0):

            result['status'] = 2

            result['message'] = 'Empty data'

            result['data'] = []

            return result

        tagsArrUNIQUE = []

        for row in allRows:

            tags = row['tags']

            tagsArr = tagsHelper.makeTagsStrToArray(tags)

            tagsArrUNIQUE = list(set(tagsArrUNIQUE + tagsArr))

        result['status'] = 0

        result['message'] = 'OK'

        result['data'] = tagsArrUNIQUE

        return result

    @staticmethod

    def addNewRate(userID,productID,mark):

        result = {}

        row = SqlExecuter.getOneRowsPacked('select \* from Рейтинг where user\_id = {} and product\_id = {};'.format(userID,productID))

        if(row is not None):

            result['status'] = 9

            result['message'] = 'User already set mark'

            result['data'] = []

            return result

        lastrowid = SqlExecuter.executeModif('insert into Рейтинг values({},{},{});'.format(productID,userID,mark))

        if(lastrowid != -1):

            result['status'] = 0

            result['message'] = 'OK'

            result['data'] = lastrowid

        else:

            result['status'] = 1

            result['message'] = 'SQL Runtime error'

            result['data'] = []

        return result

    @staticmethod

    def getRateOfProduct(productID):

        row = SqlExecuter.getOneRowsPacked('select round(coalesce(avg(mark),0),2) as "count" from Рейтинг where product\_id = {};'.format(productID))

        return float(row['count'])

    @staticmethod

    def getDetailsOfProduct(productID):

        result = {}

        try:

            row = SqlExecuter.getOneRowsPacked('select \*,cost\_sale as "cost",concat(book\_title," - ",author) as "title",round(avg(rate.mark),2) as "rate" from Товар inner join Рейтинг as rate where id = {};'.format(productID))

        except:

            result['status'] = 1

            result['message'] = 'SQL Runtime error'

            result['data'] = []

            return result

        if(row is None):

            result['status'] = 2

            result['message'] = 'Empty data'

            result['data'] = []

            return result

        else:

            row['imageLink'] = imageHelper.makeFullPathToImage(row['imageLink'])

            result['status'] = 0

            result['message'] = 'OK'

            result['data'] = [row]

            return result

**В.5 Листинг файла слоя обработки данных User.py**

from flask\_login import UserMixin

from app import db

from app.models.Cart import Cart

from utils import md5helper

from utils import validationForm

from app.models.SqlExecuter import SqlExecuter

class User(UserMixin):

    userID = -1

    first\_name = None

    last\_name = None

    birthdate = None

    email = None

    password\_hash = None

    is\_admin = False

    level\_of\_access = -1

    def \_\_init\_\_(self,userID,email,password\_hash,last\_name,first\_name,birthdate):

        self.userID = userID

        self.email = email

        self.password\_hash = password\_hash

        self.first\_name = first\_name

        self.last\_name = last\_name

        self.birthdate = birthdate

    def is\_can\_write(self):

        return self.is\_admin and self.level\_of\_access > 0

    def is\_can\_read(self):

        return self.is\_admin and self.level\_of\_access > -1

    def is\_active(self):

        return True

    def is\_authenticated(self):

        return True

    def is\_anonymous(self):

        return False

    def set\_admin(self,boolean):

        self.is\_admin = boolean

    def set\_level\_of\_access(self,level):

        self.level\_of\_access = level

    def get\_id(self):

        return str(self.userID)

    def getTotalCostOfCart(self):

        return Cart.countTotalCostOfUser(self.userID)

    def getCountOfItemsInCart(self):

        return Cart.getCountOfItemsInCart(self.userID)["data"][0]

    def getUserMarkTo(self,productID):

        row = SqlExecuter.getOneRowsPacked('select mark from Рейтинг where user\_id = {} and product\_id = {};'.format(self.userID,productID))

        if(row is None):

            return None

        else:

            return float(row['mark'])

    @staticmethod

    def registerUser(email,password,last\_name,first\_name,birthdate):

        result = {}

        row = SqlExecuter.getOneRowsPacked('select \* from Покупатель where email="{}";'.format(email))

        if(row is not None):# Проверка,если пользователь с таким email существует

            result['status'] = 8

            result['message'] = 'User with same email already exists'

            return result

        if(not validationForm.validationEmail(email)):# Если неправильно введен email

            result['status'] = 7

            result['message'] = 'Incorrect email'

            return result

        lastrowid = SqlExecuter.executeModif(

            'insert into Покупатель(`email`,`last\_name`,`first\_name`,`birthdate`,`password\_hash`) \

                values("{}","{}","{}","{}","{}")'.format(

                    email,last\_name,first\_name,birthdate,md5helper.ecnrypt(password))

                )

        result["status"] = 0

        result["message"] = "OK"

        result["userID"] = lastrowid

        return result

    @staticmethod

    def getInfo(value,column='id'):

        result = {}

        row = SqlExecuter.getOneRowsPacked('select \* from Покупатель where {} = {};'.format(column,value))

        if(row is None):

            result['data'] = {}

            result['status'] = 1

            result['message'] = 'User not found'

            return result

        result['data'] = row

        result['status'] = 0

        result['message'] = 'OK'

        return result

    #LEGACY

    @staticmethod

    def addUser(data):

        result = {}

        try:

            last\_name = data['last\_name']

            first\_name = data['first\_name']

            password = data['password']

            birthdate = data['birthdate']

            email = data['email']

        except KeyError:

            result['status'] = 6

            result['message'] = 'Required field is missing'

            return result

        try:

            SqlExecuter.executeModif(

                'insert into Покупатель values("{}","{}",NULL,"{}","{}","{}");'.format(

                    last\_name,first\_name,

                    email,birthdate,

                    md5helper.ecnrypt(password)

                ))

        except Exception:

            result['status'] = 1

            result['message'] = 'SQL runtime error'

            return result

        result['status'] = 0

        result['message'] = 'OK'

        return result

    @staticmethod

    def validateUser(email,password):

        row = SqlExecuter.getOneRowsPacked(

            'select \* from Покупатель where email \

             = "{}" and password\_hash = "{}"'.format(email,md5helper.ecnrypt(password))

            )

        return row is not None

    @staticmethod

    def validateUserAndReturnUserID(email,password):

        row = SqlExecuter.getOneRowsPacked(

            'select \* from Покупатель where email \

             = "{}" and password\_hash = "{}"'.format(email,md5helper.ecnrypt(password))

            )

        if(row is None):

            return -1

        else:

            return row['id'] # Возвращаем ID пользователя

**В.6 Листинг файла слоя обработки данных Admin.py**

from app import db

from app.models.SqlExecuter import connection

from app.admin.logger import AdminLogger as Logger

class Admin:

    @staticmethod

    def \_\_executeAndGetAllRowsAndKeys(sqlQuery):

        cursor = connection.cursor()

        cursor.execute(sqlQuery)

        allRows = cursor.fetchall()

        columns\_names = [i[0] for i in cursor.description]

        cursor.close()

        return {'data':allRows,'keys':columns\_names}

    @staticmethod

    def \_\_makeResultResponse(sqlQuery):

        res =  Admin.\_\_executeAndGetAllRowsAndKeys(sqlQuery)

        if(len(res['data']) == 0):

            res['status'] = 3

            res['message'] = 'Empty'

            res['data'] = []

        else:

            res['status'] = 0

            res['message'] = 'OK'

        return res

    @staticmethod

    def getSuppliers():

        return Admin.\_\_makeResultResponse('select \* from Поставщик;')

    @staticmethod

    def addNewSupplier(supplierName,adminID):

        cursor = connection.cursor()

        query = 'insert into Поставщик(`company\_name`) values("{}");'.format(supplierName)

        cursor.execute(query)

        lastrowid = cursor.lastrowid

        cursor.close()

        Logger.log(adminID,query)

        connection.commit()

        result = {}

        result['keys'] = ['new id']

        if(lastrowid > 0):

            result['data'] = [lastrowid]

            result['message'] = 'OK'

            result['status'] = 0

        else:

            result['data'] = []

            result['message'] = 'Not > 0'

            result['status'] = 125

        return result

    @staticmethod

    def setNewValueBook(productID,column,value,adminID):

        result = {}

        cursor = connection.cursor()

        try:

            if(type(value) is int or type(value) is float):

                query = 'update Товар set {}={} where id = {};'.format(column,value,productID)

            else:

                query = 'update Товар set {}="{}" where id = {};'.format(column,value,productID)

            cursor.execute(query)

            Logger.log(adminID,query)

        except:

            result['status'] = 130

            result['message'] = 'ADMIN: sql error because admin entered incorrect input data'

            return result

        lastrowid = cursor.lastrowid

        cursor.close()

        connection.commit()

        result['status'] = 0

        result['message'] = 'OK'

        result['data'] = [lastrowid]

        result['keys'] = ['updated id']

        return result

    @staticmethod

    def incQuantityOfBook(productID,quantity,adminID):

        result = {}

        if(quantity<0):

            result['status'] = 130

            result['message'] = 'ADMIN:incorrect input data'

            return result

        cursor = connection.cursor()

        query = 'update Товар set quantity=quantity+{} where id = {};'.format(quantity,productID)

        cursor.execute(query)

        Logger.log(adminID,query)

        lastrowid = cursor.lastrowid

        cursor.close()

        connection.commit()

        result['status'] = 0

        result['message'] = 'OK'

        result['data'] = [lastrowid]

        result['keys'] = ['updated id']

        return result

    @staticmethod

    def getInfoOfBookBy(columnName,value,adminID):

        query = 'select \* from Товар where {} like "%{}%";'.format(columnName,value)

        Logger.log(adminID,query)

        return Admin.\_\_makeResultResponse(query)

    @staticmethod

    def addNewDeliveryOfProducts(productID,quantity,cost\_purhase,supplier\_id,adminID):

        query = 'insert into Закупки(`quantity`,`cost\_purchase`,`supplier\_id`,`product\_id`) values({},{},{},{})'.format(quantity,cost\_purhase,supplier\_id,productID)

        cursor = connection.cursor()

        cursor.execute(query)

        Logger.log(adminID,query)

        return Admin.incQuantityOfBook(productID,quantity,adminID)

    @staticmethod

    def getColumnsOfTable(tableName):

        return Admin.\_\_makeResultResponse('SELECT `COLUMN\_NAME` FROM `INFORMATION\_SCHEMA`.`COLUMNS` WHERE `TABLE\_NAME`="{}";'.format(tableName))

    @staticmethod

    def insertNewBook(title,author,desc,cost\_sale,cost\_purchase,quantity,imageName,tags,adminID):

        cursor = connection.cursor()

        query = 'insert into Товар(`book\_title`,`author`,`description`,`cost\_sale`,`quantity`,`imageLink`,`tags`) \

            values("{}","{}","{}",{},{},"{}","{}")'.format(title,author,desc,cost\_sale,quantity,imageName,tags)

        cursor.execute(query)

        Logger.log(adminID,query)

        lastrowid = cursor.lastrowid

        query = 'insert into Закупки(`product\_id`,`cost\_purchase`,`quantity`) values({},{},{});'.format(lastrowid,cost\_purchase,quantity)

        cursor.execute(query)

        connection.commit()

        cursor.close()

        result = {}

        result['keys'] = ['new id']

        if(lastrowid>0):

            result['status'] = 0

            result['message'] = 'OK'

            result['data'] = [lastrowid,]

        else:

            result['status'] = 125

            result['message'] = 'Not > 0'

            result['data'] = []

        return result

    @staticmethod

    def getAllIncomeByPeriod(date\_b,date\_e,adminID):

        query = 'select sum(total) as "total" from Заказ where date>="{}" and date<= "{}";'.format(date\_b,date\_e)

        Logger.log(adminID,query)

        return Admin.\_\_makeResultResponse(query)

    @staticmethod

    def getRatingPopularityOfBooksByPeriod(date\_b,date\_e,adminID):

        query = "select Товар.book\_title,Товар.id,sum(Забронированная\_книга.count) as 'count',sum(Забронированная\_книга.price) as 'total' from Забронированная\_книга inner join Заказ inner join Товар where Заказ.id = order\_id and product\_id = Товар.id and Заказ.date >='{}' and Заказ.date <='{}'\

          GROUP by Товар.id order by sum(Забронированная\_книга.count) DESC;".format(date\_b,date\_e)

        Logger.log(adminID,query)

        return Admin.\_\_makeResultResponse(query)

    @staticmethod

    def getAllOrdersByPeriod(date\_b,date\_e,adminID):

        query = "select \* from Заказ as ord inner join Адрес as addr inner join Покупатель as usr where date>='{}' and date<= '{}' and ord.address\_id = addr.id and usr.id = ord.user\_id;".format(date\_b,date\_e)

        Logger.log(adminID,query)

        return Admin.\_\_makeResultResponse(query)

    @staticmethod

    def getCountOfOrderByPeriod(date\_b,date\_e,adminID):

        query = "select count(\*) as 'count' from Заказ where date>='{}' and date<= '{}';".format(date\_b,date\_e)

        Logger.log(adminID,query)

        return Admin.\_\_makeResultResponse(query)

    @staticmethod

    def getInfoOfOrderBy(orderID,adminID):

        query = 'select \* from Заказ as ord inner join Адрес as addr where ord.id = {} and address\_id = addr.id;'.format(orderID)

        Logger.log(adminID,query)

        return Admin.\_\_makeResultResponse(query)

    @staticmethod

    def getOrdersByStatus(status,adminID):

        query = 'select \* from Заказ where status = {};'.format(status)

        Logger.log(adminID,query)

        return Admin.\_\_makeResultResponse('select \* from Заказ where status = {};'.format(status))

    @staticmethod

    def \_\_unbookedBooksBy(orderID,adminID):

        query = 'select \* from Забронированная\_книга where order\_id = {};'.format(orderID)

        Logger.log(adminID,query)

        bookedBooks = Admin.\_\_executeAndGetAllRowsAndKeys(query)

        cursor = connection.cursor()

        query = 'delete from Забронированная\_книга where order\_id = {};'.format(orderID)

        cursor.execute(query)

        Logger.log(adminID,query)

        for row in bookedBooks['data']:

            query = 'update Товар set quantity=quantity+{} where id = {};'.format(row['count'],row['product\_id'])

            Logger.log(adminID,query)

            cursor.execute(query)

        cursor.close()

        connection.commit()

    @staticmethod

    def setNewStatusOfOrder(newStatus,orderID,adminID):

        cursor = connection.cursor()

        query = 'update Заказ set status = {} where id = {};'.format(newStatus,orderID)

        Logger.log(adminID,query)

        cursor.execute(query)

        lastrowid = cursor.lastrowid

        cursor.close()

        if(newStatus == 4 or newStatus == 5):

            Admin.\_\_unbookedBooksBy(orderID,adminID)

        result = {}

        result['keys'] = ['updated id']

        if(lastrowid>0):

            result['status'] = 0

            result['message'] = 'OK'

            result['data'] = [lastrowid,]

        else:

            result['status'] = 125

            result['message'] = 'Not > 0'

            result['data'] = []

        return result

    @staticmethod

    def getBookedBooksBy(orderID,adminID):

        query = 'select \* from Заброннированная\_книга where order\_id = {};'.format(orderID)

        Logger.log(adminID,query)

        return Admin.\_\_makeResultResponse(query)

    @staticmethod

    def is\_admin\_can\_write(adminID):

        query = 'select \* from Админ where user\_id = {};'.format(adminID)

        cursor = connection.cursor()

        cursor.execute(query)

        row = cursor.fetchone()

        return row['level\_of\_access']> 0

# ПРИЛОЖЕНИЕ Г

**Разработка слоя данных**

**Г.1 Листинг файла слоя данных SqlExecuter.py**

from app import db

import pymysql

from pymysql.cursors import DictCursor

connection = pymysql.connect(

    host = 'localhost',

    user = 'root',

    password = YOUR\_PASSWORD',

    db = 'shop',

    charset = 'utf8',

    use\_unicode = True,

    cursorclass = DictCursor

)

class SqlExecuter:

    @staticmethod

    def getOneRow(query):

        cursor = db.execute(query)

        row = cursor.fetchone()

        cursor.close()

        return row

    @staticmethod

    def getAllRows(query):

        cursor = db.execute(query)

        allRows = cursor.fetchall()

        cursor.close()

        return allRows

    @staticmethod

    def getOneRowAndColumns(query):

        cursor = connection.cursor()

        cursor.execute(query)

        row = cursor.fetchone()

        cursor.close()

        return row

    @staticmethod

    def getAllRowAndColumns(query):

        cursor = connection.cursor()

        cursor.execute(query)

        rows = cursor.fetchall()

        cursor.close()

        return rows

    @staticmethod

    def prepareDataByOneRow(row,columns):

        if(row is None or len(row) == 0):

            return None

        keyword = {}

        for i in range(len(columns)):

            keyword[columns[i]] = row[i]

        return keyword

    @staticmethod

    def prepareDataByManyRows(allRows,columns):

        if(allRows is None or len(allRows) == 0):

            return None

        data = []

        for row in allRows:

            data.append(SqlExecuter.prepareDataByOneRow(row,columns))

        return data

    @staticmethod

    def getAllRowsPacked(query):

        try:

            return SqlExecuter.getAllRowAndColumns(query)

        except:

            return None

    @staticmethod

    def getOneRowsPacked(query):

        try:

            return SqlExecuter.getOneRowAndColumns(query)

        except:

            return None

    @staticmethod

    def executeModif(query):

        cursor = connection.cursor()

        cursor.execute(query)

        lastrowid = cursor.lastrowid

        connection.commit()

        cursor.close()

        return lastrowid

# ПРИЛОЖЕНИЕ Д

**Разработка вспомогательных утилит**

**Д.1 Листинг файла вспомогательной утилиты addressBuilder.py**

def buildAddress(district,street,flat,floor,porch,house):

    return "{} район, {}, {}".format(district,street,house)

**Д.2 Листинг файла вспомогательной утилиты dateHelper.py**

from datetime import date

def getDate():

    return date.today()

**Д.3 Листинг файла вспомогательной утилиты emailSender.py**

import smtplib

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

from flask import render\_template

server = 'smtp.yandex.ru'

user = 'YOOUR\_email'

pswd = 'why are you so curious'

sender = 'support@booker.ru'

msg = 'TEST'

class EmailSender:

    @staticmethod

    def sendEmailTo(recipients,orderDetails):

        msg = MIMEMultipart('alternative')

        msg['From'] = user

        msg['To'] = recipients[0]

        msg['Subject'] = 'Order №{} details'.format(orderDetails['id'])

        html = render\_template('order-email.html',order = orderDetails)

        msg.attach(MIMEText(html,'html'))

        session = smtplib.SMTP\_SSL(server)

        session.login(user=user,password=pswd)

        session.sendmail(user,recipients,msg.as\_string())

        session.close()

**Д.4 Листинг файла вспомогательной утилиты imageHelper.py**

def makeFullPathToImage(imageLink):

    return 'img/book/' + imageLink

**Д.5 Листинг файла вспомогательной утилиты md5helper.py**

import hashlib

def ecnrypt(s):

    result = hashlib.md5(s.encode())

    return str(result.hexdigest())

def validateS(sHash,sToValidate):

    return sHash == str(ecnrypt(sToValidate))

**Д.6 Листинг файла вспомогательной утилиты pageHelper.py**

def getRangeOfPages(countOfPages,currentPage):

    if(currentPage + 1 <=countOfPages) and (currentPage - 1 > 0):

            return range(currentPage-1,currentPage+2)

    elif(currentPage + 1 <= countOfPages):

            return range(currentPage,currentPage+2)

    elif(currentPage - 1 > 0):

        return range(currentPage-1,currentPage+1)

    else:

        return range(currentPage,currentPage+1)

**Д.7 Листинг файла вспомогательной утилиты security.py**

from app.models.SqlExecuter import connection

class Security:

    @staticmethod

    def escape\_sql(s):

        return connection.escape\_string(s)

**Д.8 Листинг файла вспомогательной утилиты sqlQueryHelper.py**

def buildSqlQueryByTags(baseSqlQuery,tags):

    resultQuery = "{} where ".format(baseSqlQuery)

    for i in range(len(tags)-1):

        resultQuery +=  "tags like '%{}%' and ".format(tags[i])

    resultQuery += "tags like '%{}%';".format(tags[len(tags)-1])

    return resultQuery

def buildSqlQueryByTagsAndPage(baseSqlQuery,tags,page,offset):

    buildedSqlWithoutOffset = buildSqlQueryByTags(baseSqlQuery,tags)

    buildedSqlWithoutOffset = buildedSqlWithoutOffset[:len(buildedSqlWithoutOffset)-1]

    buildedSqlWithoutOffset += " order by id DESC LIMIT {0} OFFSET {1};".format(offset,offset\*(page-1))

    print(buildedSqlWithoutOffset)

    return buildedSqlWithoutOffset

**Д.9 Листинг файла вспомогательной утилиты tagsHelper.py**

def makeArrayOfTagsToStr(tags):

    if(len(tags) == 0):

        return ''

    else:

        tagsStr = ''

        for i in range(len(tags)):

            tagsStr += tags[i]

            if(i != len(tags)-1):

                tagsStr += ','

        return tagsStr

def makeTagsStrToArray(tagStr):

    return tagStr.split(',')

**Д.10 Листинг файла вспомогательной утилиты validationForm.py**

import re

def validationEmail(email):

    return re.match("[^@]+@[^@]+\.[^@]+", email) is not None

from app.models.SqlExecuter import connection

import json

class AdminLogger:

**Д.11 Листинг файла вспомогательной утилиты logger.py**

    @staticmethod

    def log(adminID,operation):

        dumpsOperation = json.dumps(operation,ensure\_ascii = False)

        cursor = connection.cursor()

        cursor.execute('insert into логи(`admin\_id`,`operation`) values({},"{}");'.format(adminID,dumpsOperation[1:len(dumpsOperation)-2]))

        cursor.close()

        connection.commit()

# ПРИЛОЖЕНИЕ Е

**Разработка html-страниц**

**Е.1 Листинг html-файла admin-add.html**

<!DOCTYPE html>

<html>

   <head>

      <title>Админ-добавление</title>

   </head>

   <body>

      <form method="POST" action="{{url\_for('adminAddNew')}}" enctype='multipart/form-data'>

         <label>Название:</label><input id='title' type="text" name="title" required>

         <br>

         <label>Автор:</label><input type="text" name="author" required>

         <br>

         <label>Описание:</label><textarea name="desc"></textarea required>

         <br>

         <label>Изображение:</label><input type="file" name='image' accept="image/png,image/jpeg,image/jpg" required>

         <br>

         <label>Количество:</label><input type="text" name="quantity" required>

         <br>

         <label>Стоимость закупки:</label><input type="text" name="cost\_purchase" required>

         <br>

         <label>Стоимость продажи:</label><input type="text" name="cost\_sale" required>

         <br>

         <label>Теги(пример:тег1,тег2,....,тегN):</label><input type="text" name="tags" required>

         <button type="submit">Submit</button>

      </form>

      {% if message %}

      <h3>{{message}}</h3>

      {% if productID %}

      <br>

      <a href="{{ url\_for('showDetailsOfBook',productID=productID)}}">Ссылка</a>

      {% endif %}

      {% endif %}

   </body>

</html>

**Е.2 Листинг html-файла admin-manage-orders.html**

<!DOCTYPE html>

<html>

   <head>

      <title>Админ-статистика</title>

   </head>

   <body>

      <form method="POST" action="{{url\_for('adminManageOrders')}}">

         <div id="period-block">

            {% if date\_e and date\_b %}

            Дата начала периода:<input type="date" name="date\_b" value="{{ date\_b }}">

            Дата конца периода:<input type="date" name="date\_e" value="{{ date\_e }}">

            {% else %}

            Дата начала периода:<input type="date" name="date\_b">

            Дата конца периода:<input type="date" name="date\_e">

            {% endif %}

            <br>

         </div>

         <div id="order-info-block" hidden>

            <br>

            <input id=31 type="number" name="orderID-3">

            <label for=31 >Идентификатор заказа</label>

         </div>

         <div id="order-info-by-status-block" hidden>

            <br>

            <select name="status">

               <option value=0>Оформлен</option>

               <option value=1>Подтвержден</option>

               <option value=2>Получено курьером</option>

               <option value=3>Получено клиентом</option>

               <option value=4>Отменен по инициативе клиента</option>

               <option value=5>Отменен по инициативе магазина</option>

            </select>

         </div>

         <div id="order-set-new-status-block" hidden>

            <br>

            <input id=51 type="number" name="orderID-5">

            <label for=51 >Идентификатор заказа</label>

            <br>

            <select name="newStatus">

               <option value=0>Оформлен</option>

               <option value=1>Подтвержден</option>

               <option value=2>Получено курьером</option>

               <option value=3>Получено клиентом</option>

               <option value=4>Отменен по инициативе клиента</option>

               <option value=5>Отменен по инициативе магазина</option>

            </select>

         </div>

         <div id="supplier-add-new" hidden>

            <br>

            <input id=61 type="text" name="supplier-name">

            <label for=61 >Название поставщика</label>

         </div>

         <h4></h4>

         <input id = 2 type="radio" name="method" value=2>

         <label for=2>Вывести количество заказов</label>

         <br>

         <input id = 1 type="radio" name="method" value=1 checked>

         <label for=1>Вывести всех заказы</label>

         <br>

         <input id = 3 type="radio" name="method" value=3>

         <label for=3>Вывести информацию о заказе</label>

         <br>

         <input id = 4 type="radio" name="method" value=4>

         <label for=4>Вывести заказы по статусу заказа</label>

         <br>

         <input id = 5 type="radio" name="method" value=5>

         <label for=5>Установить новый статус заказа</label>

         <br>

         <input id = 6 type="radio" name="method" value=6>

         <label for=6>Добавить поставщика</label>

         <br>

         <button type="submit">ОК</button>

      </form>

      {% include 'base/admin-table.html' %}

   </body>

   <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

   <script type="text/javascript">

      var currentBlock = $('#period-block')

      $('input[type=radio][name=method]').change(function() {

        currentBlock.hide()

        if (this.value == 1 || this.value == 2) {

            currentBlock = $('#period-block')

        }

        else if (this.value == 3) {

            currentBlock = $('#order-info-block')

        }

        else if (this.value == 4) {

            currentBlock = $('#order-info-by-status-block')

        }

        else if (this.value == 5) {

            currentBlock = $('#order-set-new-status-block')

        }

        else if (this.value == 6) {

            currentBlock = $('#supplier-add-new')

        }

        currentBlock.show()

      });

   </script>

</html>

**Е.3 Листинг html-файла admin-page.html**

<!DOCTYPE html>

<html>

  <head><title>Админ</title></head>

<body>

    <p><a href="{{ url\_for('adminStat')}}">Статистика</a></p>

    <p><a href="{{ url\_for('adminAddNew')}}">Добавить книгу</a></p>

    <p><a href="{{  url\_for('adminGetProductInfo') }}">Получить информацию о книге</a></p>

    <p><a href="{{ url\_for('adminManageOrders') }}">Управлять заказами</a></p>

</body>

</html>

**Е.4 Листинг html-файла admin-product.html**

<!DOCTYPE html>

<html>

   <head>

      <title>Админ-статистика</title>

   </head>

   <body>

      <h3>Столбцы</h3>

      <form method="POST" action="{{ url\_for('adminGetProductInfo')  }}" enctype='multipart/form-data'>

         <h3>Методы</h3>

         <input type="radio" id = 1 name="method" value=1 checked>

         <label for=1>Получить информацию о книге</label>

         <br>

         <input type="radio" id = 2 name="method" value=2>

         <label for=2>Установить новое количество</label>

         <br>

         <input type="radio" id = 3 name="method" value=3>

         <label for=3>Установить новое описание</label>

         <br>

         <input type="radio" id = 4 name="method" value=4>

         <label for=4>Установить новую цену</label>

         <br>

         <input type="radio" id = 6 name="method" value=6>

         <label for=6>Установить новые теги</label>

         <br>

         <input type="radio" id = 5 name="method" value=5>

         <label for=5>Установить новое изображение</label>

         <br>

         <input type="radio" id = 7 name="method" value=7>

         <label for=7>Добавить новую поставку</label>

         <br>

         <div id="method-2-block" hidden>

            <br>

            <input id=20 type="number" name="quantity" min="0">

            <label for=20 >Новое количество</label>

            <br>

            <input id=21 type="number" name="productID-2">

            <label for=21 >Идентификатор продукта</label>

         </div>

         <br>

         <div id="method-1-block">

            <select id="admin-column-select" name="column">

               {% for column in columnNames %}

               <option value="{{column['COLUMN\_NAME']}}">{{loop.index}}. {{column['COLUMN\_NAME']}}</option>

               {% endfor %}

            </select>

            <br>

            Value:<input type="text" name="value">

         </div>

         <div id="method-4-block" hidden>

            <br>

            <input id=40 type="number" step="0.01" name="price" min="0.00">

            <label for=40 >Новая цена</label>

            <br>

            <input id=41 type="number" name="productID-4">

            <label for=41 >Идентификатор продукта</label>

         </div>

         <div id="method-3-block" hidden>

            <br>

            <input id=30 type="number" name="productID-3" min="0">

            <label for=30 >Идентификатор продукта</label>

            <br>

            <textarea id=31 name="desc">Введите новое описание</textarea>

            <label for=31 >Новое описание</label>

         </div>

         <div id="method-6-block" hidden>

            <br>

            <input id=60 type="number" name="productID-6" min="0">

            <label for=60 >Идентификатор продукта</label>

            <br>

            <input type="text" id=61 name="tags" hint='Enter new tags'></input>

            <label for=61 >Новые теги</label>

         </div>

         <div id='method-7-block' hidden>

            <br>

            <input id=70 type="number" name="productID-7">

            <label for=70 >Идентификатор продукта</label>

            <br>

            <input id=71 type="number" name="supplierID">

            <label for=71 >Идентификатор поставщика</label>

            <br>

            <input id=72 type="number" name="quantity-delivery">

            <label for=72 >Количество в поставке</label>

            <br>

            <input id=73 type="number" step="0.01" name="cost-purchase-delivery">

            <label for=73 >Стоимость закупки</label>

            <table>

               <caption>Поставщики</caption>

               <tr>

                   <th>Count</th>

                   {% for key in suppliers['keys'] %}

                       <th>{{key}}</th>

                   {% endfor %}

               </tr>

               {% for row in suppliers['data']%}

               <tr>

                   <th>{{ loop.index }}</th>

                   {% for key in suppliers['keys'] %}

                   <th>{{row[key]}}</th>

                   {% endfor %}

               </tr>

               {% endfor %}

           </table>

         </div>

         <div id="method-5-block" hidden>

            <br>

            <label>Изображение:</label><input type="file" name='image' accept="image/png,image/jpeg,image/jpg">

            <input id=50 type="number" name="productID-5" min="0">

            <label for=50 >Идентификатор продукта</label>

            <br>

         </div>

         <button type="submit">ОК</button>

      </form>

      {% include 'base/admin-table.html' %}

      <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

      <script type="text/javascript">

         var currentBlock = $('#method-1-block')

         $('input[type=radio][name=method]').change(function() {

           currentBlock.hide()

         //   console.log('#method-{this.value}-block')

           currentBlock = $(`#method-${this.value}-block`)

           currentBlock.show()

         });

      </script>

   </body>

</html>

**Е.5 Листинг html-файла admin-stat.html**

<!DOCTYPE html>

<html>

   <head>

      <title>Админ-статистика</title>

   </head>

   <body>

      <form method="POST" action="{{url\_for('adminStat')}}">

         {% if date\_e and date\_b %}

         Дата начала периода:<input type="date" name="date\_b" value="{{ date\_b }}">

         Дата конца периода:<input type="date" name="date\_e" value="{{ date\_e }}">

         {% else %}

         Дата начала периода:<input type="date" name="date\_b">

         Дата конца периода:<input type="date" name="date\_e">

         {% endif %}

         <br>

         <h4>Методы</h4>

         <input id = 2 type="radio" name="method" value=2>

         <label for=2>Вывести топ книг</label>

         <br>

         <input id = 1 type="radio" name="method" value=1>

         <label for=1>Получить всю прибыль</label>

         <br>

         <button type="submit">ОК</button>

      </form>

      {% include 'base/admin-table.html' %}

   </body>

</html>

**Е.6 Листинг html-файла advanced-search.html**

<!DOCTYPE html>

<html lang="zxx">

   <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <meta http-equiv="X-UA-Compatible" content="ie=edge">

      <title>Booker</title>

      <!-- Google Font -->

      <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;900&display=swap" rel="stylesheet">

      <!-- Css Styles -->

      <link rel="stylesheet" href="{{url\_for('static',filename='css/bootstrap.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/font-awesome.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/elegant-icons.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/nice-select.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/jquery-ui.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/owl.carousel.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/slicknav.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/style.css')}}" type="text/css">

   </head>

   <body>

      <!-- Page Preloder -->

      <div id="preloder">

         <div class="loader"></div>

      </div>

      {% include 'base/humberger.html' %}

      {% include 'base/header.html' %}

      {% include 'base/hero.html' %}

      <section class="shoping-cart spad">

         <div class="container">

            <form action="{{url\_for('showBooks',page=1)}}" method="GET">

               <div class='row'>

                  {% if not error%}

                  {% for tag in tags%}

                  <div class="tag">{{tag}}<input type="checkbox" name="tags" value="{{tag}}"></div>

                  {% endfor %}

                  {% else %}

                  <h3>{{error}}</h3>

                  {% endif %}

               </div>

               <button type="submit" class="site-btn">SEARCH</button>

            </form>

         </div>

      </section>

      <!-- Shoping Cart Section End -->

      {% include 'base/footer.html' %}

      <!-- Js Plugins -->

      <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/bootstrap.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.nice-select.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery-ui.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.slicknav.js')}}"></script>

      <script src="{{url\_for('static',filename='js/mixitup.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/main.js')}}"></script>

   </body>

</html>

**Е.7 Листинг html-файла books.html**

<!DOCTYPE html>

<html lang="zxx">

   <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <meta http-equiv="X-UA-Compatible" content="ie=edge">

      <title>Booker</title>

      <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;900&display=swap" rel="stylesheet">

      <link rel="stylesheet" href="{{  url\_for('static',filename='css/bootstrap.min.css')  }}" type="text/css">

      <link rel="stylesheet" href="{{  url\_for('static',filename='css/font-awesome.min.css')  }}" type="text/css">

      <link rel="stylesheet" href="{{  url\_for('static',filename='css/elegant-icons.css')  }}" type="text/css">

      <link rel="stylesheet" href="{{  url\_for('static',filename='css/nice-select.css')  }}" type="text/css">

      <link rel="stylesheet" href="{{  url\_for('static',filename='css/jquery-ui.min.css')  }}" type="text/css">

      <link rel="stylesheet" href="{{  url\_for('static',filename='css/owl.carousel.min.css')  }}" type="text/css">

      <link rel="stylesheet" href="{{  url\_for('static',filename='css/slicknav.min.css')  }}" type="text/css">

      <link rel="stylesheet" href="{{  url\_for('static',filename='css/style.css')  }}" type="text/css">

   </head>

   <body>

      <style type="text/css">

         .page-btn {

         font-size: 14px;

         color: #ffffff;

         font-weight: 800;

         text-transform: uppercase;

         text-align: center;

         display: inline-block;

         padding: 3px 10px 3px;

         background: #78b818;

         border: none;

         margin-bottom: 10px;

         margin-left: 5px;

         margin-right: 5px;

         border-radius: 50%;

         }

         .page-btn-active {

         font-size: 14px;

         color: #ffffff;

         font-weight: 800;

         text-align: center;

         text-transform: uppercase;

         display: inline-block;

         padding: 3px 10px 3px;

         background: #49681c;

         border: none;

         margin-bottom: 10px;

         margin-left: 5px;

         margin-right: 5px;

         border-radius: 50%;

         }

         #msg\_pop{

         background-color:#7fad39;

         display: none;

         position: fixed;

         z-index: 9998;

         bottom: 30px;

         left:30px;

         width: 250px;

         padding: 10px;

         color:#fff;

         font-size:13px;

         line-height:13px;

         -webkit-box-shadow: 0px 0px 10px #999;

         -moz-box-shadow: 0px 0px 10px #999;

         box-shadow: 0px 0px 10px #7fad39;

         -webkit-border-radius: 15px;

         -moz-border-radius: 15px;

         border-radius: 15px;

         }

         #msg\_pop h4{

         margin:0;

         text-align:center;

         color: #fff;

         font-size:15px;

         }

         #msg\_close{

         display:block;

         position:absolute;

         top:5px;

         right:10px;

         line-height:15px;

         width:16px;

         height:16px;

         text-align:center;

         color:#fff;

         cursor:pointer;

         -webkit-border-radius: 10px;

         -moz-border-radius: 10px;

         -ms-border-radius: 10px;

         -o-border-radius: 10px;

         border-radius: 10px;

         }

         #msg\_close:hover {

         background-color:#fff;

         color:#000;

         }

         .fadeIn{

         animation-name: fadeIn;

         -webkit-animation-name: fadeIn;

         animation-duration: 0.4s;

         -webkit-animation-duration: 0.4s;

         animation-timing-function: ease-in-out;

         -webkit-animation-timing-function: ease-in-out;

         visibility: visible !important;

         }

         @keyframes fadeIn {

         0% {transform: scale(0.7);opacity: 0.5;}

         80% {transform: scale(1.1);}

         100% {transform: scale(1);opacity: 1;}

         }

         @-webkit-keyframes fadeIn {

         0% {-webkit-transform: scale(0.7);opacity: 0.5;}

         80% {-webkit-transform: scale(1.1);}

         100% {-webkit-transform: scale(1);opacity: 1;}

         }

         /\* .header\_\_menu {

         } \*/

         /\* .header\_\_menu ul li .header\_\_menu\_\_dropdown {

         position: absolute;

         right: 50px;

         top: 50px;

         background: #222222;

         width: 180px;

         z-index: 9;

         padding: 5px 0;

         -webkit-transition: all, 0.3s;

         -moz-transition: all, 0.3s;

         -ms-transition: all, 0.3s;

         -o-transition: all, 0.3s;

         transition: all, 0.3s;

         opacity: 0;

         visibility: hidden;

         } \*/

      </style>

      <div id="preloder">

         <div class="loader"></div>

      </div>

      {% include 'base/humberger.html' %}

      {% include 'base/header.html' %}

      {% include 'base/hero.html' %}

      <section class="featured spad">

      <div class="container">

         <div class="row featured\_\_filter">

            {% if products %}

            {%  for product in products%}

            <div class="col-lg-3 col-md-4 col-sm-6 mix">

               <div class="featured\_\_item" >

                  <div class="featured\_\_item\_\_pic set-bg"

                     data-setbg="{{ url\_for('static',filename = product.imageLink)}}">

                     <a href="{{url\_for('showDetailsOfBook',productID=product.id)}}"></a>

                     <ul class="featured\_\_item\_\_pic\_\_hover">

                        <li><a class="shopping\_\_cart" id={{product.id}} value={{product.cost}}><i class="fa fa-shopping-cart"></i></a></li>

                     </ul>

                  </div>

                  <div class="featured\_\_item\_\_text">

                     <h6><a href="{{ url\_for('showDetailsOfBook',productID=product.id) }}">{{product.title}}</a></h6>

                     <h5>{{product.cost}}P</h5>

                  </div>

               </div>

            </div>

            {%  endfor %}

            {% endif %}

         </div>

      </div>

      {% if countOfPagesRange %}

      <div class="justify-content-center row">

         {% for page in countOfPagesRange%}

         {% if page == currentPage%}

         {% if tagsAlreadySearched%}

         <a class="page-btn-active" href="{{ url\_for('showBooks',page=page,tags=tagsAlreadySearched)}}">{{page}}</a>

         {% else %}

         {% if q%}

         <a class="page-btn-active" href="{{ url\_for('showBooks',page=page,q=q)}}">{{page}}</a>

         {% else %}

         <a class="page-btn-active" href="{{ url\_for('showBooks',page=page)}}">{{page}}</a>

         {% endif %}

         {% endif %}

         {% else%}

         {% if tagsAlreadySearched%}

         <a class="page-btn" href="{{ url\_for('showBooks',page=page,tags=tagsAlreadySearched)}}">{{page}}</a>

         {% else %}

         {% if q%}

         <a class="page-btn" href="{{ url\_for('showBooks',page=page,q=q)}}">{{page}}</a>

         {% else %}

         <a class="page-btn" href="{{ url\_for('showBooks',page=page)}}">{{page}}</a>

         {% endif %}

         {% endif %}

         {% endif %}

         {% endfor %}

      </div>

      {% endif %}

      <div id="msg\_pop">

         <h4>Успешно добавлено!</h4>

      </div>

      {% include 'base/footer.html' %}

      <script src="{{  url\_for('static',filename='js/jquery-3.3.1.min.js')  }}" type="text/javascript"></script>

      <script src="{{  url\_for('static',filename='js/bootstrap.min.js')  }}" type="text/javascript"></script>

      <script src="{{  url\_for('static',filename='js/jquery.nice-select.min.js')  }}" type="text/javascript"></script>

      <script src="{{  url\_for('static',filename='js/jquery-ui.min.js')  }}" type="text/javascript"></script>

      <script src="{{  url\_for('static',filename='js/jquery.slicknav.js')  }}" type="text/javascript"></script>

      <script src="{{  url\_for('static',filename='js/mixitup.min.js')  }}" type="text/javascript"></script>

      <script src="{{  url\_for('static',filename='js/owl.carousel.min.js')  }}" type="text/javascript"></script>

      <script src="{{  url\_for('static',filename='js/main.js')  }}" type="text/javascript"></script>

      <script async src="https://www.googletagmanager.com/gtag/js?id=UA-23581568-13" type="baea7c064f6a573d1522ee04-text/javascript"></script>

      <script src="https://ajax.cloudflare.com/cdn-cgi/scripts/7089c43e/cloudflare-static/rocket-loader.min.js" data-cf-settings="baea7c064f6a573d1522ee04-|49" defer=""></script><script defer src="https://static.cloudflareinsights.com/beacon.min.js" data-cf-beacon='{"rayId":"5d1b2c4d3b588474","version":"2020.9.0","si":10}'></script>

   </body>

</html>

**Е.8 Листинг html-файла checkout.html**

<!DOCTYPE html>

<html lang="zxx">

   <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <meta http-equiv="X-UA-Compatible" content="ie=edge">

      <title>Booker</title>

      <!-- Google Font -->

      <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;900&display=swap" rel="stylesheet">

      <!-- Css Styles -->

      <link rel="stylesheet" href="{{url\_for('static',filename='css/bootstrap.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/font-awesome.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/elegant-icons.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/nice-select.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/jquery-ui.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/owl.carousel.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/slicknav.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/style.css')}}" type="text/css">

   </head>

   <body>

      <!-- Page Preloder -->

      <div id="preloder">

         <div class="loader"></div>

      </div>

      {% include 'base/humberger.html' %}

      {% include 'base/header.html' %}

      {% include 'base/hero.html' %}

      <!-- Checkout Section Begin -->

      <section class="checkout spad">

         <div class="container">

            {% if error %}

            <h4>{{ error }}</h4>

            {% else %}

            <div class="checkout\_\_form">

               <h4>Оформление</h4>

               <form action="{{url\_for('newOrder')}}" method="POST">

                  <div class="row">

                     <div class="col-lg-8 col-md-6">

                        <div class="row">

                           <div class="col-lg-6">

                              <div class="checkout\_\_input">

                                 <p>Имя<span>\*</span></p>

                                 <input name="first\_name" type="text" value="{{ user.first\_name }}" required>

                              </div>

                           </div>

                           <div class="col-lg-6">

                              <div class="checkout\_\_input">

                                 <p>Фамилия<span>\*</span></p>

                                 <input name="last\_name" type="text"  value="{{ user.last\_name }}" required>

                              </div>

                           </div>

                        </div>

                        <div class="checkout\_\_input">

                           <p>Адрес<span>\*</span></p>

                           <input name="district" type="text" placeholder="District\*" class="checkout\_\_input\_\_add" required>

                           <input name="street" type="text" placeholder="Street Address\*" class="checkout\_\_input\_\_add" required>

                           <input name="house" type="text" placeholder="House\*" class="checkout\_\_input\_\_add" required>

                           <input name="floor" type="text" placeholder="Floor" class="checkout\_\_input\_\_add">

                           <input name="flat" type="text" placeholder="Apartment" class="checkout\_\_input\_\_add">

                           <input name="porch" type="text" placeholder="Porch" class="checkout\_\_input\_\_add">

                        </div>

                        <div class="row">

                           <div class="col-lg-6">

                              <div class="checkout\_\_input">

                                 <p>Телефон<span>\*</span></p>

                                 <input name="phone" type="text" required>

                              </div>

                           </div>

                           <div class="col-lg-6">

                              <div class="checkout\_\_input">

                                 <p>Email<span></span></p>

                                 <input name="email" type="text" placeholder="You will receive an email">

                              </div>

                           </div>

                        </div>

                        <!-- <div class="checkout\_\_input">

                           <p>Account Password<span>\*</span></p>

                           <input type="text">

                           </div> -->

                        <div class="checkout\_\_input">

                           <p>Дополнительное описание</p>

                           <input type="text" name='desc'

                              placeholder="Notes about your order, e.g. special notes for delivery.">

                        </div>

                     </div>

                     <div class="col-lg-4 col-md-6">

                        <div class="checkout\_\_order">

                           <h4>Ваш заказ</h4>

                           <div class="checkout\_\_order\_\_products">Книги <span>Всего</span></div>

                           <ul>

                              {% for product in productsInCart %}

                              <li>{{product.title}}<span>{{ product.total }}</span></li>

                              {% endfor %}

                           </ul>

                           <!-- <div class="checkout\_\_order\_\_subtotal">Subtotal <span>{{total}}</span></div> -->

                           <div class="checkout\_\_order\_\_total">Итого <span>{{user.getTotalCostOfCart()}}</span></div>

                           <div class="checkout\_\_input\_\_checkbox">

                           </div>

                           <button type="submit" class="site-btn">ОФОРМИТЬ</button>

                        </div>

                     </div>

                  </div>

               </form>

            </div>

            {% endif %}

         </div>

      </section>

      <!-- Checkout Section End -->

      {% include 'base/footer.html' %}

      <!-- Js Plugins -->

      <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/bootstrap.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.nice-select.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery-ui.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.slicknav.js')}}"></script>

      <script src="{{url\_for('static',filename='js/mixitup.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/main.js')}}"></script>

   </body>

</html>

**Е.9 Листинг html-файла order-email.html**

<!DOCTYPE html>

<html lang="zxx">

   <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <meta http-equiv="X-UA-Compatible" content="ie=edge">

      <title>Booker</title>

      <!-- Google Font -->

      <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;900&display=swap" rel="stylesheet">

      <!-- Css Styles -->

      <link rel="stylesheet" href="{{url\_for('static',filename='css/bootstrap.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/font-awesome.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/elegant-icons.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/nice-select.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/jquery-ui.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/owl.carousel.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/slicknav.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/style.css')}}" type="text/css">

   </head>

   <body>

      <!-- Checkout Section Begin -->

      <section class="checkout spad">

         <div class="container">

            {% if error %}

            <h4>{{ error }}</h4>

            {% else %}

            {% if message %}

            <h3>{{message}}</h3>

            {% endif %}

            <div class="checkout\_\_form">

               <h4>Order №{{order['id']}} Details</h4>

               {% for product in order['data'] %}

               <p>

               <h4>{{ product.title }}</h4>

               : x{{product.count}} = product.total</p>

               {% endfor %}

            </div>

            {% endif %}

         </div>

      </section>

      <!-- Checkout Section End -->

      {% include 'base/footer.html' %}

      <!-- Js Plugins -->

      <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/bootstrap.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.nice-select.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery-ui.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.slicknav.js')}}"></script>

      <script src="{{url\_for('static',filename='js/mixitup.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/main.js')}}"></script>

   </body>

</html>

**Е.10 Листинг html-файла order-page.html**

<!DOCTYPE html>

<html lang="zxx">

   <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <meta http-equiv="X-UA-Compatible" content="ie=edge">

      <title>Booker</title>

      <!-- Google Font -->

      <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;900&display=swap" rel="stylesheet">

      <!-- Css Styles -->

      <link rel="stylesheet" href="{{url\_for('static',filename='css/bootstrap.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/font-awesome.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/elegant-icons.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/nice-select.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/jquery-ui.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/owl.carousel.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/slicknav.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/style.css')}}" type="text/css">

      <style type="text/css">

         .pro-qty-cart {

         width: 140px;

         height: 50px;

         display: inline-block;

         position: relative;

         text-align: center;

         background: #f5f5f5;

         margin-bottom: 5px;

         }

         .pro-qty-cart input {

         height: 100%;

         width: 100%;

         font-size: 16px;

         color: #6f6f6f;

         width: 50px;

         border: none;

         background: #f5f5f5;

         text-align: center;

         }

         .pro-qty-cart .qtybtn {

         width: 35px;

         font-size: 16px;

         color: #6f6f6f;

         cursor: pointer;

         display: inline-block;

         }

      </style>

   </head>

   <body>

      <!-- Page Preloder -->

      <div id="preloder">

         <div class="loader"></div>

      </div>

      {% include 'base/humberger.html' %}

      {% include 'base/header.html' %}

      {% include 'base/hero.html' %}

      <!-- Shoping Cart Section Begin -->

      <section class="shoping-cart spad">

         <div class="container">

            <div class="row">

               <div class="col-lg-6">

                  <h3>Информация о заказе</h3>

                  <div class="row">

                     <div class="col-lg-6">

                        <div class="checkout\_\_input">

                           <p>Дата</p>

                           <input type="text" value="{{ orderInfo.date }}" disabled>

                        </div>

                     </div>

                     <div class="col-lg-6">

                        <div class="checkout\_\_input">

                           <p>Адрес</p>

                           <input type="text"  value="{{ address.address}}" disabled>

                        </div>

                     </div>

                  </div>

                  <div class="row">

                     <!-- <div class="col-lg-6">

                        <div class="checkout\_\_input">

                            <p>Phone</p>

                            <input name="phone" type="text" required>

                        </div>

                        </div> -->

                     <div class="col-lg-6">

                        <div class="checkout\_\_input">

                           {% if orderInfo.status == 0 %}

                           <p>Статус: оформлен</p>

                           {% endif %}

                           {% if orderInfo.status == 1 %}

                           <p>Статус: подтвержден</p>

                           {% endif %}

                           {% if orderInfo.status == 2 %}

                           <p>Статус: получен курьером</p>

                           {% endif %}

                           {% if orderInfo.status == 3 %}

                           <p>Статус: получен Вами!</p>

                           {% endif %}

                           {% if orderInfo.status == 4 %}

                           <p>Статус: отменен Вами(</p>

                           {% endif %}

                           {% if orderInfo.status == 5 %}

                           <p>Статус: отменен нами(</p>

                           {% endif %}

                        </div>

                     </div>

                  </div>

                  {% if orderInfo.status == 0 or orderInfo.status == 1 %}

                  <div class="row">

                        <a href="{{ url\_for('cancelOrder',orderID = orderInfo.orderID) }}" class="primary-btn cart-btn">Отменить заказ</a>

                  </div>

                  {% endif %}

               </div>

            </div>

            <div class="row">

               <div class="col-lg-12">

                  <div class="shoping\_\_cart\_\_table">

                     <table>

                        <thead>

                           <tr>

                              <th class="shoping\_\_product">Книги</th>

                              <!-- <th>Price</th> -->

                              <th>Количество</th>

                              <th>Итого</th>

                              <th></th>

                           </tr>

                        </thead>

                        <tbody>

                           {%if not error%}

                           {% if data|length == 0 %}

                           <tr>

                              <td>Empty cart</td>

                           </tr>

                           {% else %}

                           {% for product in data %}

                           <tr id={{ product.id }}>

                              <td class="shoping\_\_cart\_\_item">

                                 <img style="max-width:25%;" src="{{ url\_for('static',filename=product.imageLink) }}" alt="">

                                 <h5><a style="text-decoration: none;" href="{{ url\_for('showDetailsOfBook',productID=product.id)}}">{{ product.title}}</a></h5>

                              </td>

                              <td class="shoping\_\_cart\_\_quantity">

                                 <div class="quantity">

                                    <div class="pro-qty-cart">

                                       <input type="text" value="{{product.count}}" disabled>

                                    </div>

                                 </div>

                              </td>

                              <td class="shoping\_\_cart\_\_total" id="{{product.id}}">

                                 {{product.total}}P

                              </td>

                           </tr>

                           {% endfor %}

                           {%endif%}

                           {% else %}

                           <tr>

                              <td>{{error}}</td>

                           </tr>

                           {% endif %}

                        </tbody>

                     </table>

                  </div>

               </div>

            </div>

            <div class="col-lg-6">

               <div class="shoping\_\_checkout">

                  <h5>Итоговая стоимость</h5>

                  <ul>

                     <li>Итого <span>{{  orderInfo.total  }}P</span></li>

                  </ul>

                  {% if error %}

                  <h3>{{error}}</h3>

                  {% endif %}

               </div>

            </div>

            <div class="row">

               <div class="col-lg-6" style="margin-top: 10px; margin-left: 15px;">

                  <div class="shoping\_\_cart\_\_btns">

                     <a href="{{ url\_for('showBooks',page=1) }}" class="primary-btn cart-btn">ПРОДОЛЖИТЬ ПОКУПКИ</a>

                  </div>

               </div>

            </div>

         </div>

      </section>

      <!-- Shoping Cart Section End -->

      {% include 'base/footer.html' %}

      <!-- Js Plugins -->

      <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/bootstrap.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.nice-select.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery-ui.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.slicknav.js')}}"></script>

      <script src="{{url\_for('static',filename='js/mixitup.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/main.js')}}"></script>

   </body>

</html>

**Е.11 Листинг html-файла order.html**

<!DOCTYPE html>

<html lang="zxx">

   <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <meta http-equiv="X-UA-Compatible" content="ie=edge">

      <title>Booker</title>

      <!-- Google Font -->

      <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;900&display=swap" rel="stylesheet">

      <!-- Css Styles -->

      <link rel="stylesheet" href="{{url\_for('static',filename='css/bootstrap.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/font-awesome.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/elegant-icons.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/nice-select.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/jquery-ui.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/owl.carousel.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/slicknav.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/style.css')}}" type="text/css">

   </head>

   <body>

      <!-- Page Preloder -->

      <div id="preloder">

         <div class="loader"></div>

      </div>

      {% include 'base/humberger.html' %}

      {% include 'base/header.html' %}

      {% include 'base/hero.html' %}

      <!-- Checkout Section Begin -->

      <section class="shoping-cart spad">

         <div class="container">

         <div class="row">

            <div class="col-lg-12">

               <div class="shoping\_\_cart\_\_table">

                  {% if not error %}

                  <h3>Заказ оформлен. Скоро с Вами свяжутся наши операторы!</h3>

                  {% endif %}

                  <table>

                     <thead>

                        <tr>

                           <th class="shoping\_\_product">Книги</th>

                           <!-- <th>Price</th> -->

                           <th>Количество</th>

                           <th>Итого</th>

                           <th></th>

                        </tr>

                     </thead>

                     <tbody>

                        {%if not error%}

                        {% if data|length == 0 %}

                        <tr>

                           <td>Что?</td>

                        </tr>

                        {% else %}

                        {% for product in data %}

                        <tr id={{ product.id }}>

                           <td class="shoping\_\_cart\_\_item">

                              <img style="max-width:25%;" src="{{ url\_for('static',filename=product.imageLink) }}" alt="">

                              <h5><a style="text-decoration: none;" href="{{ url\_for('showDetailsOfBook',productID=product.id)}}">{{ product.title}}</a></h5>

                           </td>

                           <td class="shoping\_\_cart\_\_quantity">

                              <div class="quantity">

                                 <div class="pro-qty-cart">

                                    <input type="text" value="{{product.count}}" disabled>

                                 </div>

                              </div>

                           </td>

                           <td class="shoping\_\_cart\_\_total" id="{{product.id}}">

                              {{product.total}}P

                           </td>

                        </tr>

                        {% endfor %}

                        {%endif%}

                        {% else %}

                        <tr>

                           <td>{{error}}</td>

                        </tr>

                        {% endif %}

                     </tbody>

                  </table>

               </div>

            </div>

         </div>

      </section>

      <!-- Checkout Section End -->

      {% include 'base/footer.html' %}

      <!-- Js Plugins -->

      <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/bootstrap.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.nice-select.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery-ui.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.slicknav.js')}}"></script>

      <script src="{{url\_for('static',filename='js/mixitup.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/main.js')}}"></script>

   </body>

</html>

**Е.12 Листинг html-файла orders-page.html**

<!DOCTYPE html>

<html lang="zxx">

   <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <meta http-equiv="X-UA-Compatible" content="ie=edge">

      <title>Booker</title>

      <!-- Google Font -->

      <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;900&display=swap" rel="stylesheet">

      <!-- Css Styles -->

      <link rel="stylesheet" href="{{url\_for('static',filename='css/bootstrap.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/font-awesome.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/elegant-icons.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/nice-select.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/jquery-ui.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/owl.carousel.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/slicknav.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/style.css')}}" type="text/css">

      <style type="text/css">

         .pro-qty-cart {

         width: 140px;

         height: 50px;

         display: inline-block;

         position: relative;

         text-align: center;

         background: #f5f5f5;

         margin-bottom: 5px;

         }

         .pro-qty-cart input {

         height: 100%;

         width: 100%;

         font-size: 16px;

         color: #6f6f6f;

         width: 50px;

         border: none;

         background: #f5f5f5;

         text-align: center;

         }

         .pro-qty-cart .qtybtn {

         width: 35px;

         font-size: 16px;

         color: #6f6f6f;

         cursor: pointer;

         display: inline-block;

         }

      </style>

   </head>

   <body>

      <!-- Page Preloder -->

      <div id="preloder">

         <div class="loader"></div>

      </div>

      {% include 'base/humberger.html' %}

      {% include 'base/header.html' %}

      {% include 'base/hero.html' %}

      <!-- Shoping Cart Section Begin -->

      <section class="shoping-cart spad">

         <div class="container">

            <div class="row">

               <div class="col-lg-12">

                  <div class="shoping\_\_cart\_\_table">

                     <table>

                        <thead>

                           <tr>

                              <th class="shoping\_\_product">Заказы</th>

                              <th>Статус</th>

                              <th>Дата</th>

                              <th>Итого</th>

                              <th></th>

                           </tr>

                        </thead>

                        <tbody>

                           {%if not error%}

                           {% if ordersList|length == 0 %}

                           <tr>

                              <td>Пусто</td>

                           </tr>

                           {% else %}

                           {% for order in ordersList %}

                           <tr>

                              <td class="shoping\_\_cart\_\_item">

                                 {% if order.status == 4 or order.status == 5%}

                                 <h5>Заказr№{{ order.id}}</h5>

                                 {% else %}

                                 <h5><a style="text-decoration: none;" href="{{ url\_for('showDetailsOfOrder',orderID=order.id)}}">Заказ№{{ order.id}}</a></h5>

                                 {% endif %}

                              </td>

                              <td class="shoping\_\_cart\_\_price">

                                 {% if order.status == 0 %}

                                 Оформлен

                                 {% endif %}

                                 {% if order.status == 1 %}

                                 Подтвержден

                                 {% endif %}

                                 {% if order.status == 2 %}

                                 Получен курьером

                                 {% endif %}

                                 {% if order.status == 3 %}

                                 Получен Вами!

                                 {% endif %}

                                 {% if order.status == 4 %}

                                 Отменен Вами(

                                 {% endif %}

                                 {% if order.status == 5 %}

                                 Отменен нами(

                                 {% endif %}

                              </td>

                              <td class="shoping\_\_cart\_\_quantity">

                                 <div class="quantity">

                                    <div class="pro-qty-cart">

                                       {{order.date}}

                                    </div>

                                 </div>

                              </td>

                              <td class="shoping\_\_cart\_\_total">

                                 {{order.total}}P

                              </td>

                           </tr>

                           {% endfor %}

                           {%endif%}

                           {% else %}

                           <tr>

                              <td>{{error}}</td>

                           </tr>

                           {% endif %}

                        </tbody>

                     </table>

                  </div>

               </div>

            </div>

            <div class="row">

               <div class="col-lg-12">

                  <div class="shoping\_\_cart\_\_btns">

                     <a href="{{ url\_for('showBooks',page=1) }}" class="primary-btn cart-btn">ПРОДОЛЖИТЬ ПОКУПКИ</a>

                  </div>

               </div>

            </div>

         </div>

      </section>

      <!-- Shoping Cart Section End -->

      {%  include 'base/footer.html' %}

      <!-- Js Plugins -->

      <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/bootstrap.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.nice-select.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery-ui.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.slicknav.js')}}"></script>

      <script src="{{url\_for('static',filename='js/mixitup.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/main.js')}}"></script>

   </body>

</html>

**Е.13 Листинг html-файла shop-details.html**

<!DOCTYPE html>

<html lang="zxx">

   <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <meta http-equiv="X-UA-Compatible" content="ie=edge">

      <title>Booker</title>

      <!-- Google Font -->

      <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;900&display=swap" rel="stylesheet">

      <!-- Css Styles -->

      <link rel="stylesheet" href="{{url\_for('static',filename='css/bootstrap.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/font-awesome.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/elegant-icons.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/nice-select.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/jquery-ui.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/owl.carousel.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/slicknav.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/style.css')}}" type="text/css">

   </head>

   <body>

      <!-- Page Preloder -->

      <div id="preloder">

         <div class="loader"></div>

      </div>

      {% include 'base/humberger.html' %}

      {% include 'base/header.html' %}

      {% include 'base/hero.html' %}

      <!-- Product Details Section Begin -->

      <section class="product-details spad">

         <div class="container">

            <div class="row">

               <div class="col-lg-6 col-md-6">

                  <div class="product\_\_details\_\_pic">

                     <div class="product\_\_details\_\_pic\_\_item\_\_small">

                        {%if not error%}

                        <img class="product\_\_details\_\_pic\_\_item\_\_small"

                           src="{{ url\_for('static',filename = details.imageLink)}}" alt="">

                        {% else%}

                        <img class="product\_\_details\_\_pic\_\_item\_\_small"

                           src="{{ url\_for('static',filename = 'img/book/placeholder.jpg')}}" alt="">

                        {% endif%}

                     </div>

                  </div>

               </div>

               <div class="col-lg-6 col-md-6">

                  <div class="product\_\_details\_\_text">

                     {% if error%}

                     <h3>{{ error }}</h3>

                     {% else%}

                     <h3>{{ details.title}}</h3>

                     {% endif %}

                     <!-- <div class="product\_\_details\_\_rating hidden">

                        <i class="fa fa-star"></i>

                        <i class="fa fa-star"></i>

                        <i class="fa fa-star"></i>

                        <i class="fa fa-star"></i>

                        <i class="fa fa-star-half-o"></i>

                        <span>(18 reviews)</span>

                        </div> -->

                     {% if error%}

                     <div class="product\_\_details\_\_price">{{ error }}</div>

                     {% else%}

                     <div class="product\_\_details\_\_price">{{ details.cost}}P</div>

                     {% endif %}

                     {% if error%}

                     <p>{{ error }}</p>

                     {% else%}

                     <p>{{details.description}}</p>

                     {% endif %}

                     {% if user.is\_authenticated%}

                     {% if quantityInCart.status == 0 %}

                     <div class="product\_\_details\_\_quantity">

                        <div class="quantity">

                           <div class="pro-qty"  value='{{ details.id }}'>

                              <span class="dec qtybtn">-</span><input type="text" value="{{quantityInCart.data.count }}" disabled><span class="inc qtybtn">+</span>

                           </div>

                        </div>

                     </div>

                     <p class="primary-btn" id="primary\_\_btn\_\_add\_\_to\_\_cart" style="display:none">ДОБАВИТЬ В КОРЗИНУ</p>

                     {% else %}

                     <div class="product\_\_details\_\_quantity">

                        <div class="quantity">

                           <div style="display:none" class="pro-qty"  value='{{ details.id }}'>

                              <span class="dec qtybtn">-</span><input type="text" value="{{ quantityInCart.data.count }}" disabled><span class="inc qtybtn">+</span>

                           </div>

                        </div>

                     </div>

                     <p class="primary-btn" id="primary\_\_btn\_\_add\_\_to\_\_cart">ДОБАВИТЬ В КОРЗИНУ</p>

                     {% endif %}

                     {% else %}

                     <a href="/login" class="primary-btn" id="primary-btn">ДОБАВИТЬ В КОРЗИНУ</a>

                     {% endif %}

                     <ul>

                        {% if error%}

                        <li>{{ error }}</li>

                        {% else%}

                        <li><b>Количество</b> <span>{{details.quantity}}</span></li>

                        <li><b>Рейтинг</b> <span>{{details.rate}}/5.0</span></li>

                        {% if user.is\_authenticated %}

                        {% if userMark %}

                        <li><b>Ваша оценка</b><span>{{userMark}}</span></li>

                        {% endif %}

                        {% endif %}

                        {% endif %}

                     </ul>

                     {% if user.is\_authenticated %}

                     {% if not userMark %}

                     <h3>Оцените данную книгу!</h3>

                     <select id="rating-bar" value={{details.id}}>

                        <option value="1">1</option>

                        <option value="2">2</option>

                        <option value="3">3</option>

                        <option value="4">4</option>

                        <option value="5">5</option>

                     </select>

                     <button id="rating-bar-submit">Рейтинг</button>

                     <p id='rating-bar-status'></p>

                     {% endif %}

                     {% endif %}

                  </div>

               </div>

               <div class="col-lg-12">

                  <div class="product\_\_details\_\_tab">

                     <ul class="nav nav-tabs" role="tablist">

                        <li class="nav-item">

                           <a class="nav-link active" data-toggle="tab" href="#tabs-1" role="tab"

                              aria-selected="true">Теги</a>

                        </li>

                        <!-- <li class="nav-item">

                           <a class="nav-link" data-toggle="tab" href="#tabs-2" role="tab"

                               aria-selected="false">Information</a>

                           </li> -->

                     </ul>

                     <div class="tab-content">

                        <div class="tab-pane active" id="tabs-1" role="tabpanel">

                           <div class="product\_\_details\_\_tab\_\_desc">

                              <h6>Теги продукта</h6>

                              {% if error%}

                              <p>{{ error }}</p>

                              {% else%}

                              <p>{{ details.tags}}</p>

                              {% endif %}

                           </div>

                        </div>

                        <!-- <div class="tab-pane" id="tabs-2" role="tabpanel">

                           <div class="product\_\_details\_\_tab\_\_desc">

                               <h6>Products Infomation</h6>

                           </div>

                           </div> -->

                     </div>

                  </div>

               </div>

            </div>

         </div>

      </section>

      <!-- Product Details Section End -->

      {% include 'base/footer.html' %}

      <!-- Js Plugins -->

      <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/bootstrap.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.nice-select.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery-ui.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.slicknav.js')}}"></script>

      <script src="{{url\_for('static',filename='js/mixitup.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/main.js')}}"></script>

      <script type="text/javascript" src="http://bseth99.github.io/projects/jquery-ui/ratingbar.js"></script>

   </body>

</html>

**Е.14 Листинг html-файла shoping-cart.html**

<!DOCTYPE html>

<html lang="zxx">

   <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <meta http-equiv="X-UA-Compatible" content="ie=edge">

      <title>Booker</title>

      <!-- Google Font -->

      <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;900&display=swap" rel="stylesheet">

      <!-- Css Styles -->

      <link rel="stylesheet" href="{{url\_for('static',filename='css/bootstrap.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/font-awesome.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/elegant-icons.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/nice-select.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/jquery-ui.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/owl.carousel.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/slicknav.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/style.css')}}" type="text/css">

      <style type="text/css">

         .pro-qty-cart {

         width: 140px;

         height: 50px;

         display: inline-block;

         position: relative;

         text-align: center;

         background: #f5f5f5;

         margin-bottom: 5px;

         }

         .pro-qty-cart input {

         height: 100%;

         width: 100%;

         font-size: 16px;

         color: #6f6f6f;

         width: 50px;

         border: none;

         background: #f5f5f5;

         text-align: center;

         }

         .pro-qty-cart .qtybtn {

         width: 35px;

         font-size: 16px;

         color: #6f6f6f;

         cursor: pointer;

         display: inline-block;

         }

      </style>

   </head>

   <body>

      <!-- Page Preloder -->

      <div id="preloder">

         <div class="loader"></div>

      </div>

      {% include 'base/humberger.html' %}

      {% include 'base/header.html' %}

      {% include 'base/hero.html' %}

      <!-- Shoping Cart Section Begin -->

      <section class="shoping-cart spad">

         <div class="container">

            <div class="row">

               <div class="col-lg-12">

                  <div class="shoping\_\_cart\_\_table">

                     <table>

                        <thead>

                           <tr>

                              <th class="shoping\_\_product">Книги</th>

                              <th>Цена</th>

                              <th>Количество</th>

                              <th>Итого</th>

                              <th></th>

                           </tr>

                        </thead>

                        <tbody>

                           {%if not error%}

                           {% if productsInCart|length == 0 %}

                           <tr>

                              <td>Пусто(</td>

                           </tr>

                           {% else %}

                           {% for product in productsInCart %}

                           <tr id={{ product.id }}>

                              <td class="shoping\_\_cart\_\_item">

                                 <img style="max-width:25%;" src="{{ url\_for('static',filename=product.imageLink) }}" alt="">

                                 <h5><a style="text-decoration: none;" href="{{ url\_for('showDetailsOfBook',productID=product.id)}}">{{ product.title}}</a></h5>

                              </td>

                              <td class="shoping\_\_cart\_\_price">

                                 {{ product.cost }}P

                              </td>

                              <td class="shoping\_\_cart\_\_quantity">

                                 <div class="quantity">

                                    <div class="pro-qty-cart" value='{{ product.id }}' id='{{ product.cost }}'>

                                       <span class="dec qtybtn">-</span><input type="text" value="{{product.count}}" disabled><span class="inc qtybtn">+</span>

                                    </div>

                                 </div>

                              </td>

                              <td class="shoping\_\_cart\_\_total" id="{{product.id}}">

                                 {{product.total}}P

                              </td>

                              <td class="shoping\_\_cart\_\_item\_\_close">

                                 <span class="icon\_close"></span>

                              </td>

                           </tr>

                           {% endfor %}

                           {%endif%}

                           {% else %}

                           <tr>

                              <td>{{error}}</td>

                           </tr>

                           {% endif %}

                        </tbody>

                     </table>

                  </div>

               </div>

            </div>

            <div class="row">

               <div class="col-lg-12">

                  <div class="shoping\_\_cart\_\_btns">

                     <a href="{{ url\_for('showBooks',page=1) }}" class="primary-btn cart-btn">ПРОДОЛЖИТЬ ПОКУПКИ</a>

                  </div>

               </div>

               <div class="col-lg-6">

                  <div class="shoping\_\_checkout">

                     <h5>Итого в корзине</h5>

                     <ul>

                        <li>Итого <span>{{  user.getTotalCostOfCart()  }}P</span></li>

                     </ul>

                     {% if not error %}

                     {% if not productsInCart|length == 0 %}

                     <a href="/checkout" class="primary-btn" id="primary-btn-checkout">ПЕРЕЙТИ К ОФОРМЛЕНИЮ</a>

                     {% endif %}

                     {% else %}

                     <h3>{{error}}</h3>

                     {% endif %}

                  </div>

               </div>

            </div>

         </div>

      </section>

      <!-- Shoping Cart Section End -->

      {% include 'base/footer.html' %}

      <!-- Js Plugins -->

      <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/bootstrap.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.nice-select.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery-ui.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.slicknav.js')}}"></script>

      <script src="{{url\_for('static',filename='js/mixitup.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/main.js')}}"></script>

   </body>

</html>

**Е.15 Листинг html-файла base/admn-table.html**

{% if message %}

    <h3>{{message}}</h3>

    {% else %}

    {% if resultOfResponse %}

        <table>

            <caption>Result</caption>

            <tr>

                <th>Count</th>

                {% for key in resultOfResponse['keys'] %}

                    <th>{{key}}</th>

                {% endfor %}

            </tr>

            {% for row in resultOfResponse['data']%}

            <tr>

                <th>{{ loop.index }}</th>

                {% for key in resultOfResponse['keys'] %}

                <th>{{row[key]}}</th>

                {% endfor %}

            </tr>

            {% endfor %}

        </table>

    {% endif %}

    {% endif %}

**Е.16 Листинг html-файла base/footer.html**

<footer class="footer spad">

    <div class="container">

      <div class="row">

      <div class="col-lg-3 col-md-6 col-sm-6">

      <div class="footer\_\_about">

       <div class="footer\_\_about\_\_logo">

      <a href="{{ url\_for('showBooks',page = 1)}}"><img width=20% height=20% src="{{ url\_for('static',filename='img/logo.png') }}" alt=""></a>

      </div>

      <ul>

      <li>12 линия ВО,51(Общежитие СПБГУТ "Василеостровское")</li>

      <li>Телефон: 8(800)-555-35-35</li>

      </ul>

      </div>

      </div>

      </div>

      <div class="row">

      <div class="col-lg-12">

      <div class="footer\_\_copyright">

      <div class="footer\_\_copyright\_\_text"><p>

      Copyright &copy;<script data-cfasync="false" src="/cdn-cgi/scripts/5c5dd728/cloudflare-static/email-decode.min.js"></script><script type="baea7c064f6a573d1522ee04-text/javascript">document.write(new Date().getFullYear());</script> All rights reserved

      </p></div>

      <div class="footer\_\_copyright\_\_payment"><img src="img/payment-item.png" alt=""></div>

      </div>

      </div>

      </div>

      </div>

  </footer>

**Е.17 Листинг html-файла base/header.html**

<header class="header">

    <div class="header\_\_top">

      <div class="container">

      <div class="row">

      <div class="col-lg-6 col-md-6">

      </div>

      <div class="col-lg-6 col-md-6">

      <div class="header\_\_top\_\_right">

        {% if user.is\_authenticated %}

        <div class="header\_\_top\_\_right\_\_auth">

          <nav class="header\_\_menu">

            <ul>

                <li><a href="#"><span><i class="fa fa-user"></i> Профиль</span></a>

                    <ul class="header\_\_menu\_\_dropdown">

                      <li><a href="{{ url\_for('userInfo')}}">Информаци</a></li>

                      <li><a href="{{ url\_for('showOrders')}}">Заказы</a></li>

                      <li><a href="{{ url\_for('logoutUser')}}">Выход</a></li>

                    </ul>

                </li>

            </ul>

        </nav>

      </div>

        {% else %}

      <div class="header\_\_top\_\_right\_\_auth">

      <a href="{{url\_for('loginUser')}}"><i class="fa fa-user"></i>Вход</a>

      </div>

      {% endif %}

      </div>

      </div>

      </div>

      </div>

      </div>

      <div class="container">

      <div class="row">

      <div class="col-lg-3">

      <div class="header\_\_logo">

      <a href="{{ url\_for('showBooks',page=1) }}"><img width="30%" height="30%" src="{{ url\_for('static',filename = 'img/logo.png') }}" alt=""></a>

      </div>

      </div>

      <div class="col-lg-3">

        <div class="header\_\_logo hidden">

        <a href="./index.html"><img src="img/logo.png" alt=""></a>

        </div>

        </div>

        <div class="col-lg-3">

          <div class="header\_\_logo hidden">

          <a href="./index.html"><img src="img/logo.png" alt=""></a>

          </div>

          </div>

      <div class="col-lg-3">

      <div class="header\_\_cart">

      <ul>

        {% if user.is\_authenticated%}

      <li><a href="/cart"><i class="fa fa-shopping-bag"></i> <span id="fa\_\_fa-shopping-bag\_\_indicator\_\_quantity">{{user.getCountOfItemsInCart()}}</span></a></li>

        {% else %}

      <li><a href="{{url\_for('loginUser')}}"><i class="fa fa-shopping-bag"></i> <span>0</span></a></li>

        {% endif %}

      </ul>

        {% if user.is\_authenticated %}

      <div class="header\_\_cart\_\_price">В корзине:<span id="header\_\_cart\_\_price\_\_totalCost">{{user.getTotalCostOfCart()}}P</span></div>

        {% else %}

      <div class="header\_\_cart\_\_price">В корзине:<span>0.0P</span></div>

        {% endif %}

      </div>

      </div>

      </div>

      <div class="humberger\_\_open">

      <i class="fa fa-bars"></i>

      </div>

      </div>

    </header>

**Е.18 Листинг html-файла base/hero.html**

<section class="hero">

    <div class="container">

      <div class="row">

      <div class="col-lg-12">

      <div class="hero\_\_search">

      <div class="hero\_\_search\_\_form">

      <form action="{{ url\_for('showBooks',page=1) }}" method="GET">

        {% if q%}

      <input name="q" type="text" placeholder="Введите запрос" value = "{{ q }}">

        {%else %}

      <input name="q" type="text" placeholder="Введите запрос">

        {% endif %}

      <button type="submit" class="site-btn">ПОИСК</button>

      </form>

      </div>

      <a class="site-btn" href="{{ url\_for('searchAdvanced')}}">РАСШИРЕННЫЙ ПОИСК</a>

      <div class="hero\_\_search\_\_phone">

      <div class="hero\_\_search\_\_phone\_\_icon">

      <i class="fa fa-phone"></i>

      </div>

      <div class="hero\_\_search\_\_phone\_\_text">

      <h5>8(800)555-35-35</h5>

      <span>Поддержка 24/7</span>

      </div>

      </div>

      </div>

      </div>

      </div>

      </div>

  </section>

**Е.19 Листинг html-файла base/humberger.html**

<!-- Humberger Begin -->

<div class="humberger\_\_menu\_\_overlay"></div>

<div class="humberger\_\_menu\_\_wrapper">

    <div class="humberger\_\_menu\_\_logo">

    <a href="{{ url\_for('showBooks',page = 1)}}"><img src="{{ url\_for('static',filename='img/logo.png') }}" alt=""></a>

    </div>

    <div class="humberger\_\_menu\_\_cart">

    <ul>

    <li>

      <a href="/cart">

      <i class="fa fa-shopping-bag"></i>

      {% if user.is\_authenticated %}

       <span id="humberg\_\_fa\_\_fa-shopping-bag\_\_indicator\_\_quantity">{{user.getCountOfItemsInCart()}}</span>

       {% else %}

       <span id="fa\_\_fa-shopping-bag\_\_indicator\_\_quantity">0</span>

       {% endif %}

    </a>

    </li>

    </ul>

    {% if user.is\_authenticated %}

    <div class="header\_\_cart\_\_price">В корзине: <span id="humberg\_\_header\_\_cart\_\_price\_\_totalCost">{{user.getTotalCostOfCart()}}P</span></div>

    {% else %}

    <div class="header\_\_cart\_\_price">В корзине: <span>0.0P</span></div>

    {% endif %}

    </div>

    <div class="humberger\_\_menu\_\_widget">

      {% if user.is\_authenticated %}

      <div class="header\_\_top\_\_right\_\_auth">

        <a href="{{url\_for('userInfo')}}"><i class="fa fa-user"></i>Профиль</a>

        </div>

      {% else %}

    <div class="header\_\_top\_\_right\_\_auth">

    <a href="{{url\_for('loginUser')}}"><i class="fa fa-user"></i>Вход</a>

    </div>

      {% endif %}

    </div>

    <div id="mobile-menu-wrap"></div>

    <div class="humberger\_\_menu\_\_contact">

    <ul>

    <li>Закажи, прочитай, поумней</li>

    </ul>

    </div>

    </div>

<!-- Humberger End -->

**Е.20 Листинг html-файла user-profie.html**

<!DOCTYPE html>

<html lang="zxx">

   <head>

      <meta charset="UTF-8">

      <meta name="viewport" content="width=device-width, initial-scale=1.0">

      <meta http-equiv="X-UA-Compatible" content="ie=edge">

      <title>Booker</title>

      <!-- Google Font -->

      <link href="https://fonts.googleapis.com/css2?family=Cairo:wght@200;300;400;600;900&display=swap" rel="stylesheet">

      <!-- Css Styles -->

      <link rel="stylesheet" href="{{url\_for('static',filename='css/bootstrap.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/font-awesome.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/elegant-icons.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/nice-select.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/jquery-ui.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/owl.carousel.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/slicknav.min.css')}}" type="text/css">

      <link rel="stylesheet" href="{{url\_for('static',filename='css/style.css')}}" type="text/css">

   </head>

   <body>

      <!-- Page Preloder -->

      <div id="preloder">

         <div class="loader"></div>

      </div>

      {% include 'base/humberger.html' %}

      {% include 'base/header.html' %}

      {% include 'base/hero.html' %}

      <!-- User Section Begin -->

      <section class="checkout spad">

         <div class="container">

            <div class="row">

               <div class="col-lg-6">

                  <h3>Профиль пользователя</h3>

                  <div class="row">

                     <div class="col-lg-6">

                        <div class="checkout\_\_input">

                           <p>Имя</p>

                           <input name="first\_name" type="text" value="{{ user.first\_name }}" disabled>

                        </div>

                     </div>

                     <div class="col-lg-6">

                        <div class="checkout\_\_input">

                           <p>Фамилия</p>

                           <input name="last\_name" type="text"  value="{{ user.last\_name }}" disabled>

                        </div>

                     </div>

                  </div>

                  <div class="row">

                     <!-- <div class="col-lg-6">

                        <div class="checkout\_\_input">

                            <p>Phone</p>

                            <input name="phone" type="text" required>

                        </div>

                        </div> -->

                     <div class="col-lg-6">

                        <div class="checkout\_\_input">

                           <p>Email</p>

                           <input name="email" type="text" value="{{ user.email }}" disabled>

                        </div>

                     </div>

                  </div>

               </div>

            </div>

         </div>

      </section>

      <!-- Checkout Section End -->

      {% include 'base/footer.html' %}

      <!-- Js Plugins -->

      <script src="{{url\_for('static',filename='js/jquery-3.3.1.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/bootstrap.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.nice-select.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery-ui.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/jquery.slicknav.js')}}"></script>

      <script src="{{url\_for('static',filename='js/mixitup.min.js')}}"></script>

      <script src="{{url\_for('static',filename='js/main.js')}}"></script>

   </body>

</html>

# ПРИЛОЖЕНИЕ И

**Разработка конфигурации проекта**

**И.1 Листинг файла конфигурации проекта config.py**

import os

from datetime import timedelta

class BaseConfig(object):

    UPLOAD\_FOLDER = "app\\static\\img\\book"

    PLACEHOLDER\_NAME = 'placeholder.jpg'

    SECRET\_KEY = os.environ.get("SECRET\_KEY") or os.urandom(50)

    REMEMBER\_COOKIE\_DURATION = timedelta(days=7)

    JSON\_AS\_ASCII = False