Техническое задание
на разработку интернет-магазина книг

Структура базы данных

См. файл со списком и описаниями таблиц и полей — https://docs.google.com/document/d/1k5h92w20HbLSswTfft3gvU9wBGQn_idK2se-SZvo3f8/edit

Таблицы для хранения информации об оценках пользователями конкретных книг, а также о тэгах и привязке книг к тэгам, должны быть спроектированы студентом самостоятельно исходя из структуры соответствующих страниц, команд API и полученных знаний.

При создании отдельных таблиц в базе данных рекомендуется заполнять их тестовыми данными, используя сервис https://www.mockaroo.com/. При этом данных должно быть достаточно для тестирования всех функций сайта: например, если техническим заданием предусмотрено отображение на странице 20-ти книг, а также подгрузка остальных порциями по 20 при нажатии кнопки “Показать ещё”, рекомендуется сгенерировать данные хотя бы для 2-3-х подгрузок и, желательно, количество, не кратное 20-ти, чтобы протестировать скрытие кнопки “Показать ещё” после подгрузки всех книг.

Описания страниц

В адресах страниц заглавными буквами может быть указано слово SLUG. Оно означает, что в данном месте адреса находится мнемонический идентификатор сущности, к которой относится данная страница. Обозначение /books/SLUG, к примеру, может означать, что адреса таких страниц имеют вид: /books/pyat-sposobov-programmirovaniya, где “pyat-sposobov-programmirovaniya” — это мнемонический идентификатор конкретной книги.

Все параметры, указанные в описаниях страниц (например, количество элементов, которые выводятся изначально), должны быть вынесены в конфигурацию приложения.

Шапка

На всех страницах сайта шапка одинаковая и состоит из нескольких компонентов:

- Логотип. Ссылка на файл с изображением должна быть прописана в шаблоне. Сам логотип следует выбрать самостоятельно, например, на сайте бесплатных иконок — https://www.flaticon.com/
- Заголовок. Его необходимо устанавливать в шаблоне, беря из параметров конфигурации приложения. В параметрах нужно создать соответствующую переменную, содержащую строку.
- Главное меню. Все пункты меню должны вести на соответствующие разделы приложения: “Жанры”, “Новинки”, “Популярное”, “Авторы”. Это необходимо делать по мере реализации соответствующих разделов на сайте.
- Блоки с книгами, к которым имеет отношение текущий пользователь: купил их, отложил или положил в корзину (см. ниже “Принципы откладывания, перемещения в корзину и покупки книг”):
    - Блок "Мои книги" — отображается только у авторизованных пользователей, рядом с фразой “Мои книги” должно отображаться количество моих книг — книг, которые купил текущий пользователь, или 0, если он не купил ни одной книги, а сама фраза должна быть ссылкой на соответствующий раздел сайта.
    - Блок "Отложено" — отображается у всех пользователей, рядом со словом “Отложено” должно отображаться количество отложенных текущим пользователем книг или 0, если у него нет ни одной отложенной книги, а само слово должно быть ссылкой на соответствующий раздел сайта.
    - Блок "Корзина" — отображается у всех пользователей, рядом со словом “Корзина” должно отображаться количество книг, добавленных текущим пользователем в корзину, либо 0, если текущий пользователь не добавил ни одной книги. Само слово “Корзина” должно быть ссылкой на соответствующий раздел сайта.
- Поле поиска, при нажатии <Enter> в котором происходит переход на страницу с результатами поиска по введённому в поле запросу.
- Блок авторизации. В неавторизованном состоянии имеет вид кнопки “Войти”, в авторизованном — блока с именем текущего пользователя (ссылкой на его профиль) и балансом его личного счёта.

Футер

В левой части футера должен быть логотип и заголовок сайта, его краткое описание и блок с иконками-ссылками на страницы ресурса в социальных сетях (эти ссылки устанавливать не нужно, поскольку проект является учебным).

В центральной части футера должны быть частично продублированы ссылки на разделы сайта из шапки, но всё есть ряд отличий. Главное меню в футере состоит из следующих пунктов:

1. Главная (отображается только в неавторизованном состоянии)
2. Жанры
3. Новинки
4. Популярное
5. Авторы

Адреса соответствующих страниц вам необходимо прописать в шаблоне самостоятельно при реализации соответствующих разделов.

У авторизованных пользователей в центре футера должен также отображаться блок с четырьмя элементами:

- Блок "Мои книги" (полностью аналогичен такому же блоку в шапке)
- Блок "Корзина" (полностью аналогичен такому же блоку в шапке)
- Блок "Отложено" (полностью аналогичен такому же блоку в шапке)
- Ссылка “Мой профиль” (ссылка на профиль текущего пользователя)

В правой части футера должен быть блок со следующими ссылками:

- Ссылка "Войти" (ведёт на страницу с формой авторизации, отображается только у неавторизованных пользователей)
- Ссылка “Документы” (ссылка на перечень всех документов, имеющихся на сайте).
- Ссылка на раздел “О компании”
- Ссылка на раздел “Помощь”
- Ссылка на раздел “Контакты”.

Главная страница (адрес страницы — /)

На главной странице размещаются три блока с книгами: список рекомендуемых книг, список новинок и список популярных книг. И под этими тремя блоками располагается облако тэгов.

В списке рекомендуемых, новинок и популярных книги выводятся в виде ленты, которая прокручивается вправо и подгружается по мере прокрутки (см. “Список рекомендуемых книг”, “Список новинок” и “Список популярных книг” в документации по API). Также смотрите “Алгоритм выбора рекомендуемых книг” и “Алгоритм определения популярности книг”

Тэги в облаке тэгов должны выводиться в виде множества ссылок разного размера, в котором размер ссылки должен соответствовать частоте использования того или иного тэга. Алгоритм определения размера тэга должен быть разработан студентом самостоятельно.

Списки книг

Во всех списках (на всех страницах) книги выводятся в одинаковом формате и содержат следующие элементы:

- Обложка
- Название
- Имя и фамилию первого автора и фразу “и другие”, если у книги несколько авторов
- Метку “бестселлер” поверх обложки, если книга является бестселлером
- Метку со скидкой, если на данную книгу установлена скидка
- Метку состояния для текущего пользователя, если книга отложена, помещена в корзину или куплена
- Цену обычную и цену со скидкой, если на книгу есть скидка. В случае, если на книгу есть скидка, обычная цена отображается зачёркнутой.

В случае, если какой-либо список пуст (например, корзина или список рекомендуемых), должно отображаться соответствующее сообщение, например “Корзина пуста” или “Рекомендуемых книг нет”.

На главной странице в рекомендациях, новинках и популярном, а также в соответствующих разделах на сайте не должны выводиться книги, которые куплены, добавлены в корзину или отложены текущим пользователем

Жанры (адрес страницы — /genres)

На странице “Жанры” должно размещаться дерево ссылок-жанров. Вложенность может быть разного уровня, но на последнем уровне жанры должны располагаться в одной строке (не списком, в котором каждая ссылка в отдельной строке).

В скобках около каждой ссылки должно быть указано количество книг, относящихся к данному жанру, а сами ссылки на всех уровнях должны быть отсортированы по убыванию количества привязанных к ним книг.

Ссылки должны вести на страницы со списками книг по соответствующему жанру.

Книги по жанру (адреса страниц — /genres/SLUG)

В верхней части страницы должны отображаться “хлебные крошки” — последовательность ссылок всех вышестоящих жанров. Последним пунктом в этой последовательности должно быть название текущего жанра (не ссылка), а первыми двумя — слово “Книги”, являющееся ссылкой на страницу с жанрами (/genres), а также “Главная” — ссылка на главную страницу.

Основная часть страницы — список книг по жанру в виде “плитки”. На странице должны отображаться первые 20 книг и кнопка “Показать ещё”, по нажатию на которую должны подгружаться следующие 20 книг. См. “Список книг по жанру” в документации по API.

Если подгружены все книги, кнопка “Показать ещё” на frontend скрывается автоматически.

Новинки (адрес страницы — /books/recent)

В верхней части страницы должны располагаться “хлебные крошки” из треёх пунктов — ссылки “Главная”, ведущей на главную страницу, ссылки “Книги”, ведущей на страницу с жанрами (/genres) и слова “Новинки”, не являющимся ссылкой.

Ниже должен располагаться календарь, в котором по умолчанию выбран только период “с” с датой ровно месяц назад относительно текущей даты. При выборе дат должны подгружаться книги, опубликованные за указанный период (см. “Список новинок” в документации по API).

На странице должны отображаться первые 20 книг (последние опубликованные книги в порядке убывания даты публикации — от самой новой до самой старой) и кнопка “Показать ещё”, по нажатию на которую должны подгружаться следующие 20 книг.

Популярное (адрес страницы — /books/popular)

В верхней части страницы должны располагаться “хлебные крошки” из треёх пунктов — ссылки “Главная”, ведущей на главную страницу, ссылки “Книги”, ведущей на страницу с жанрами (/genres) и слова “Популярное”, не являющимся ссылкой.

Внешний вид страницы полностью аналогичен внешнему виду страницы “Новинки” за исключением того, что на данной странице не должно быть календаря.

На странице должны отображаться первые 20 самых популярных книг по убыванию популярности и кнопка “Показать ещё”, по нажатию на которую должны подгружаться следующие 20 книг. См. “Список популярных книг” в документации по API и “Алгоритм определения популярности книг”.

Авторы (адрес страницы — /authors)

На странице с авторами должен размещаться список ссылок на всех авторов, отсортированный по алфавиту, и авторы должны быть сгруппированы по буквам алфавита. Каждая ссылка должна вести на страницу соответствующего автора. В каждой группе (для каждой буквы алфавита) ссылки должны размещаться в три столбца, при этом они должны быть отсортированы сначала в каждом столбце, а не в строке:

Данилов Сергей	Демишев Пётр	Димкин Христофор
Данилов Тимофей	Деникин Владимир	Донской Фаддей
Данчиков Филимон	Дейнего Аристарх	Дунаева Степанида

Алфавитный указатель в верхней части страницы — перечень якорных ссылок, прокручивающих страницу вниз до соответствующей буквы.

Страница автора (адрес страницы — /authors/SLUG)

В верхней части страницы автора должны размещаться “хлебные крошки” из трёх пунктов — ссылки “Главная”, ведущей на главную страницу, ссылки “Авторы”, ведущей на страницу со списком авторов, а также имени текущего автора, не являющегося ссылкой.

На странице автора должно также размещаться имя автора, его фотография или стандартная заглушка вместо фотографии, если она не указана или файл фотографии не существует, а также биография автора. Если биография автора слишком длинная, она должна быть свёрнута, и должна отображаться кнопка “Показать полностью”, по нажатию на которую биография должна разворачиваться.

В нижней части страницы должен быть подзаголовок “Книги автора” и один ряд книг данного автора. Ниже должна быть ссылка “Все книги автора” с количеством всех книг данного автора, ведущая на страницу книг данного автора (см. ниже).

Книги автора (адрес страницы — /books/author/SLUG)

В верхней части страницы должны располагаться “хлебные крошки” из трёх пунктов — ссылки “Главная”, ведущей на главную страницу, ссылки “Книги”, ведущей на страницу с жанрами (/genres), а также имени текущего автора, не являющегося ссылкой.

На странице должны отображаться первые 20 книг (в порядке убывания даты публикации — от самой новой до самой старой) и кнопка “Показать ещё”,

по нажатию на которую должны подгружаться следующие 20 книг данного автора. См. “Список книг автора” в документации по API.

Вход (адрес страницы — /signin)

На странице располагается radio-кнопки выбора варианта входа — по e-mail или по телефону. По умолчанию отмечен вход по телефону.

После ввода e-mail или телефона должна отобразиться новая форма с текстом о том, что на указанный e-mail (или телефон) отправлен код подтверждения, и полем ввода такого кода.

После ввода неверного кода подтверждения на этой же странице ввода кода должна отображаться ошибка: “Код подтверждения введён неверно. У вас осталось N попыток”. Если количество попыток исчерпано, должна снова отобразиться страница входа с текстом “Количество попыток входа по e-mail исчерпано, попробуйте войти по телефону или повторить вход по e-mail через 5 минут” либо “Количество попыток входа по телефону исчерпано, попробуйте войти по e-mail или повторить вход по телефону через 5 минут”.

После ввода верного кода подтверждения пользователь автоматически авторизуется и возвращается на ту страницу, с которой перешёл на страницу авторизации.

Внизу страницы входа должна располагаться ссылка “Зарегистрироваться”, ведущая на страницу регистрации.

Страница регистрации (адрес страницы — /signup)

Форма регистрации содержит три основых поля — “Имя”, “Телефон” и “E-mail”. Рядом с полями “Телефон” и “E-mail” должны располагаться кнопки “Подтвердить”, которые активируются при заполнении соответствующих полей. При нажатии на них на сервер отправляется значение соответствующего поля (см. “Запрос подтверждения контакта” в документации по API), на странице рядом с этим полем отображается поле ввода кода подтверждения, а под полем — сообщение о том, что код подтверждения отправлен по данному e-mail или телефону.

После ввода кода подтверждения заданной длины (длина по умолчанию должна быть установлена в параметрах конфигурации приложения), код должен отправляться на сервер через команду “Отправка кода подтверждения” (см. документацию по API).

В случае если код подтверждения введён неверно, отображаются сообщения, аналогичные сообщениям в форме авторизации (см. выше). При исчерпании попыток входа отображается сообщение “Число попыток подтверждения превышено, повторите попытку через 5 минут” и вместо поля ввода кода подтверждения снова отображается кнопка “Подтвердить”.

Если код подтверждения введён верно, то вместо поля ввода кода подтверждения и кнопки “Подтвердить” отображается зелёным цветом сообщение “Подтверждён”.

После ввода всех сведений форма регистрации должна отправляться на ту же страницу методом POST. Если все данные введены верно, должно отобразиться сообщение об успешной регистрации и кнопка “Продолжить”, переадресующая пользователя на ту страницу, с которой он перешёл на страницу регистрации.

Корзина (адрес страницы — /cart)

На странице выводится список всех книг, добавленных текущим пользователем в корзину (значение “CART” в соответствующей таблице в базе данных). Для каждой книги должны выводиться:

- Изображение с меткой “бестселлер”, если книга является бестселлером
- Перечень авторов книги, каждый из которых является ссылкой на страницу соответствующего автора
- Название книги
- Рейтинг книги
- Кнопка “Отложить”, по нажатию на которую книга должна удаляться из корзины и перемещаться в отложенные. Для этого делается запрос на сервер с новым статусом “KEPT” (см. “Изменение статуса книги” в документации по API).
- Кнопка “Удалить”, по нажатию на которую книга должна удалять из корзины и отвязываться от текущего пользователя. Для этого делается запрос на сервер с параметром “status”, равным “UNLINK” (см. “Изменение статуса книги” в документации по API).
- Если на книгу имеется скидка, то зачёркнутая исходная и актуальная (с учётом скидки) цена на книгу, а также размер скидки. Если скидки на книгу нет, то просто исходная цена на книгу.

См. подробнее “Принципы откладывания, перемещения в корзину и покупки книг”.

Под списком книг должна отображаться общая цена на все книги в корзине. Она должна асинхронно (без перезагрузки страницы) пересчитываться, если книга удаляется из корзины или перемещается в отложенные.

Под общей ценой должна размещаться кнопка “Купить”, по нажатию на которую:

- Если пользователь не авторизован, должно происходить перенаправление на страницу входа.

- Если пользователь авторизован, должен происходить переход к платёжной системе по адресу: /payment/?sum=SUM&user=HASH, где SUM — сумма, на которую требуется пополнить счёт, а HASH — хэш пользователя, который пополняет себе счёт (поле user.hash).

После пополнения счёта платёжная система вернёт пользователя на эту же страницу методом GET или POST с параметром result, который может быть равен “true” или “false”, а также, если “result” равен “false”, с параметром “error”, который содержит текст ошибки. В случае, если передана ошибка, на данной странице на кнопкой “Купить” должно появиться сообщение об ошибке. Если же оплата произошла без ошибок, должно отображаться сообщение об успешности оплаты (вместо формы), а статус всех книг, которые были в корзине, должен в корзине поменяться на “PAID”.

Подробнее смотрите “Алгоритм работы платёжной системы”.

Отложенное (адрес страницы — /postponed)

На данной странице должен отображаться список всех книг, которые у данного пользователя находятся в статусе отложенных (значение “KEPT” в таблице в базе данных). Интерфейс пользователя должен быть идентичен таковому в корзине (см. описание выше), за исключением кнопки “Отложить”, которая на данной странице должна быть заменена кнопкой “Купить”, по нажатию на которую книга удаляется из отложенных и перемещается в корзину отправкой команды смены статуса на “CART” для соответствующей книги (см. “Изменение статуса книги” в документации по API).

Под списком книг должна располагаться кнопка “Купить все”, по нажатию на которую на сервер со статусом “CART” должен быть отправлен массив идентификаторов книг (см. “Изменение статуса книги” в документации по API).

Мой профиль (адрес страницы — /profile)

В верхней части страницы должны быть ссылки на страницу с историей транзакций и на страницу пополнения счёта.

На странице должна быть форма из трёх основных полей: “Имя”, “E-mail” и “Телефон”. Под полями “E-mail” и “Телефон” должно быть одно из трёх:

- Сообщение “Подтверждён”, если указанный в поле e-mail или телефон подтверждён

- Кнопка “Подтвердить”, если введено новое значение, которое необходимо подтвердить. При нажатии на эту кнопку на сервер должен отправляться запрос (см. “Запрос подтверждения контакта” в документации по API).
- Поле ввода кода подтверждения. При вводе в него кода подтверждения нужной длины (длина должна устанавливаться в параметрах конфигурации приложения), должен происходит запрос на сервер (см. “Отправка кода подтверждения” в документации по API).

После изменения данных, если все введённые контакты подтверждены, пользователь может нажать на кнопку “Сохранить”. При нажатии на кнопку “Отменить” все неподтверждённые контакты должны удаляться из базы.

История транзакций (адрес страницы — /transactions)

На данной странице отображается список транзакций в порядке убывания даты и времени, при этом в списке есть возможность обратной сортировки — в этом случае транзакции должны подгрузиться с сервера (см. “Список транзакций” в документации по API).

На страницу при загрузке должны выводиться первые 50 транзакций. Если ещё есть транзакции, должна отображаться кнопка “Показать ещё”, по нажатию на которую подгружаются следующие 50 транзакций до тех пор, пока не подгрузятся все транзакции. После подгрузки всех транзакций кнопка “Показать ещё” скрывается.

Пополнение счёта (адрес страницы — /payment)

На странице должен отображаться текст, поле ввода суммы и кнопка “Пополнить”, по нажатию на которую должен происходить переход к платёжной системе по адресу: /payment/?sum=SUM&user=HASH, где SUM — сумма, на которую требуется пополнить счёт, а HASH — хэш пользователя, который пополняет себе счёт (поле user.hash).

После пополнения счёта платёжная система вернёт пользователя на эту же страницу методом GET или POST с параметром result, который может быть равен “true” или “false”, а также, если “result” равен “false”, с параметром “error”, который содержит текст ошибки. В случае, если передана ошибка, на данной странице должна также находиться форма оплаты, а над ней должно красным цветом отображаться сообщение об ошибке. В случае, если оплата произошла без ошибок, должно отображаться сообщение об успешности оплаты (вместо формы).

Подробнее смотрите “Алгоритм работы платёжной системы”.

Мои книги (адрес страницы — /my)

В верхней части страницы располагается меню из двух пунктов — “Непрочитанные” и “Архив”. Страница, которая открывается по ссылке “Архив” (адрес страницы — /my/archive) полностью идентична странице “Непрочитанные” (/my).

На странице размещаются все книги, находящиеся у данного пользователя в статусе “PAID”. На странице “Архив” — все книги, находящиеся у данного пользователя в статусе “ARCHIVED”.

Все книги размещаются в виде плитки (см. раздел “Списки книг”). При нажатии на любую книгу происходит переход на страницу данной книги (см. ниже). См. также “Принципы откладывания, перемещения в корзину и покупки книг”.

Результаты поиска (адрес страницы — /search/QUERY, где QUERY — поисковый запрос)

В верхней части страницы должно быть размещено поле поиска, в которое введён поисковый запрос.

В случае, если задан пустой поисковый запрос, под полем поиска должно быть сообщение “Поисковый запрос не задан”. Если по заданному запросу не найдено ни одной книги, должно быть сообщение “По вашему запросу книги не найдены”. Если же книги найдены, должно отображаться сообщение “Найдено N книг” (или “книги”, в соответствии с правилами русского языка), а ниже должен быть выведен список книг в соответствии с алгоритмом поиска книг (см. ниже описание данного алгоритма).

Если книг больше 20-ти, должны быть выведены только первые 20 книг, а также книга “Показать ещё”, подгружающая следующие 20 книг. Если подгружены все книги, то кнопка “Показать ещё” отображаться не должна. Загрузка книг на данную страницу должна происходить через команду “Поиск книг” (см. документацию по API).

Книги по тэгу (адрес страницы — /tags/SLUG)

В верхней части страницы должен отображаться тэг, по которому произведён переход. Рядом с ним правее должны быть ссылка “Все тэги”, раскрывающая облако тэгов на данной странице. Облако тэгов должно выглядеть точно также, как и на главной странице.

Ниже должен размещаться список книг, привязанных к данному тэгу, в произвольном порядке. На странице должны отображаться первые 20 книг и кнопка “Показать ещё”, по нажатию на которую должны подгружаться следующие 20 книг. См. “Список книг по тэгу” в документации по API.

Если подгружены все книги, кнопка “Показать ещё” на frontend скрывается автоматически.

Страница книги (адрес страницы — /books/SLUG)

В верхней части страницы должны размещаться “хлебные крошки” с названием текущей книги, пунктом “Книги”, ведущий на страницу с жанрами, и пунктом “Главная”, ведущим на главную страницу.

На странице должно быть размещено изображение обложки книги, справа от которого должны также располагаться:

- Имена и фамилии всех авторов книги (ссылки на их страницы авторов)
- Название книги
- Рейтинг книги, представляющий собой 5 звёздочек, которые закрашены в соответствии с рейтингом, см. “Алгоритм расчёта рейтинга книг”. По нажатию на рейтинг должно всплывать окошко с информацией о том, какое количество людей оценило книгу в 1, 2, 3, 4 или 5 звёзд, а также общее количество проголосовавших.
- Блок оценки книги. Если текущий пользователь не оценивал данную книгу, должны отображаться 5 звёзд таким образом, чтобы при наведении на любую из них выделялась она и те звёзды, которые находятся левее. При клике на конкретную звезду оценка пользователем книги должна отправиться на сервер (команду API необходимо спроектировать и разработать самостоятельно).
- Тэги, к которым привязана данная книга. Каждый тэг должен быть ссылкой на страницу с перечнем книг по данному тэгу.
- Цена книги. Если на книгу имеется скидка, то основная цена должна быть зачёркнута, и должна отображаться также цена с учётом скидки.
- Описание книги.
- Кнопки, меняющие статус данной книги для текущего пользователя (см. также “Принципы откладывания, перемещения в корзину и покупки книг”):

Статус книги	Кнопки	Действия по нажатию
Книга не связана с пользователем	Отложить	Перемещение книги в отложенные
	Купить	Перемещение книги в корзину
Книга отложена	Отложена	Удаление из отложенных
	Купить	Перемещение книги в корзину
Книга в корзине	Отложить	Перемещение в отложенные

	В корзине	Переход в корзину
Книга куплена	Скачать	Всплывающее окно с вариантами скачивания (см. ниже “Скачивание книги”)
	В архив	Перемещение книги в архив
Книга куплена и в архиве	Вернуть из архива	Перемещение книги из архива в раздел “Непрочитанное”

- Перечень отзывов на книгу в порядке убывания их рейтинга (см. “Алгоритм расчёта рейтинга отзывов”). У каждого отзыва должна быть возможность поставить как лайк, так и дизлайк, и у отзывов должны отображаться количества и тех, и других. Команду API для сохранения лайков и дизлайков необходимо разработать самостоятельно.
- Если текущий пользователь авторизован — форма отправки отзыва на книгу (см. команду “Отправка отзыва на книгу” в документации по API). Если текущий пользователь не авторизован, вместо формы должно отображаться сообщение: “Отзывы могут оставлять только авторизованные пользователи” и две кнопки: “Войти” и “Зарегистрироваться”, ведущие на страницу входа и страницу регистрации, соответственно.

Скачивание книги (всплывающее окно)

В окне должен быть выведен список всех файлов данной книги из таблицы “book_file” с сортировкой по типу файла. У каждого файла должен быть указан формат (из поля book_file_type.name), описание формата (из поля book_file_type.description), размер в человекочитаемом формате (килобайтах или мегабайтах), а также кнопка “Скачать”, ведущая на скрипт скачивания файла. См. подробнее в разделе “Алгоритм скачивания файла”.

В случае, если пользователь превысил лимит скачивания данной книги (таблица “file_download” в базе данных), то вместо списка форматов файлов, доступных для скачивания, должно отображаться сообщение об ошибке: “Вы превысили лимит скачивания данной книги. Обратитесь в техподдержку, если хотите скачать книгу снова.”, при этом слова “Обратитесь в техподдержку” должны быть ссылкой на страницу “Контакты”.

Документы (адрес страницы — /documents)

На данной странице должен отображаться перечень всех документов в порядке сортировки (document.sort_index) — их изображений, заголовков,

начала текста, обрезанного по первому предложению или абзацу таким образом, чтобы высота блока с текстом была примерно равна высоте изображения. В качестве изображений для документов необходимо самостоятельно выбрать любое из Интернета — одинаковое для всех документов.

Изображения, заголовки и начала текстов документов должны быть ссылками, ведущими на страницы отдельных документов.

Документ (адрес страницы — /documents/SLUG)

На странице документа должен отображаться его заголовок и текст. Текст документа в базе данных хранится в формате HTML, это должно быть учтено при его выводе на страницу.

О компании (адрес страницы — /about)

Страница с заголовком “О компании” и произвольным текстом в формате HTML, содержащим 1-2 изображения и 5-10 абзацев текста.

Помощь (адрес страницы — /faq)

На странице должны быть выведены все вопросы из таблицы faq в порядке сортировки (по возрастанию faq.sort_index), ответы на них должны быть выведены в скрытых блоках ниже (кроме первого вопроса — ответ на него должен быть раскрыт при загрузке страницы), и при нажатии пользователя на ссылку “Ответ”, соответствующий ответ должен раскрываться.

Под перечнем вопросов с ответами должна быть фраза “Или свяжитесь с нами”, в которой слова “свяжитесь с нами” должны быть ссылкой, ведущей на страницу “Контакты”.

Контакты (адрес страницы — /contacts)

На странице должны выводиться контактные данные и форма обратной связи с полями “Имя”, “E-mail”, “Тема” и “Сообщение”. В случае, если пользователь авторизован, поля “Имя” и “E-mail” отображаться не должны.

Сообщения должны отправляться перезагрузкой данной страницы (методом POST) и должны сохраняться в таблице “message” (см. структуру базы данных). В случае, если какие-то поля были заполнены неверно, должно отображаться соответствующее сообщение об ошибке, при этом данные в полях должны быть теми же, которые были введены пользователем, чтобы он не вводил их снова и мог просто исправить допущенную ошибку.

Документация по API

Слово “ID”, указанное в адресах, означает, что на его месте должен быть числовой идентификатор запрашиваемой сущности, например: /api/authors/647.

Слово “***BOOK***”, указанное в JSON в качестве значений ключей, означает, что на его месте должен быть объект книги следующего формата:

```
{
    'id': 390,
    'slug': "priklyucheniya_programmistov",
    'image': "/img/34/54/12/2323.jpg",
    'authors': "Сергей Дмитриев и другие",
    'title': "Восстание Емельяна Пугачёва",
    'discount': 30,
    'isBestseller': true,
    'rating': 3,
    'status': 'KEPT',
    'price': 250,
    'discountPrice': 175
}
```

Путь к изображению обложки книги должен быть указан относительно корня сайта (начинаться со слэша ‘/’).

Если у книги несколько авторов, то первым должен идти тот, у которого минимальное значение book2author.sort_index, а вместо остальных должно быть написано “и другие”.

Если рейтинга у книги нет (см. “Алгоритм расчёта рейтинга книги”), то значением поля “rating” должно быть “false”.

В поле “status” указан статус книги по отношению к текущему пользователю: KEPT (отложена), CART (в корзине) и PAID (куплена, если она куплена или в архиве). Если книга не привязана, ключ “status” должен быть равен “false”.

В поле “price” должна быть указана цена книги в рублях. Если на книгу имеется скидка, то должно быть задано поле “discountPrice”, и оно должно содержать цену с учётом скидки.

Список рекомендуемых книг — GET /api/books/recommended

Метод выводит книги, которые рекомендуются данному пользователю или вообще, если о пользователе ничего неизвестно. См. “Алгоритм выбора рекомендуемых книг”.

В значении “count” должно выводиться общее количество книг, рекомендуемых данному пользователю.

Параметры:

- offset — сдвиг от 0 для постраничного вывода
- limit — количество книг, которое надо вывести

Формат ответа:

```
{
     'count': 390,
     'books': [
          BOOK,
          …
     ]
}
```

Список новинок — GET /api/books/recent

Метод выводит книги, отсортированные по убыванию даты публикации — от самой новой до самой старой, в пределах дат, указанных в параметрах.

В случае, если дата “from” не указана, должны выводиться все книги, опубликованные до даты “to”. Если дата “to” не указана — все книги, опубликованные с даты “from” по настоящее время. Если не указаны обе даты — все книги вообще.

Параметры:

- from — дата “от” в формате “20-05-2020”
- to — дата “до” в формате “20-05-2020”
- offset — сдвиг от 0 для постраничного вывода
- limit — количество книг, которое надо вывести

Формат ответа:

```
{
     'count': 390,
```

```
    'books': [
        BOOK,
        …
    ]
}
```

Список популярных книг — GET /api/books/popular

Метод выводит книги по убыванию популярности — от самой большой популярности до самой маленькой. См. “Алгоритм определения популярности книг”. Книги, у которых популярность равна 0, можно выводить в произвольном порядке без какой-либо специальной сортировки.

Параметры:

- offset — сдвиг от 0 для постраничного вывода
- limit — количество книг, которое надо вывести

Формат ответа:

```
{
    'count': 390,
    'books': [
        BOOK,
        …
    ]
}
```

Список книг по жанру — GET /api/books/genre/ID

Метод выводит книги, привязанные к жанру, чей ID передан в запросе, в порядке убывания даты публикации — от самой новой к самой старой.

Параметры:

- offset — сдвиг от 0 для постраничного вывода
- limit — количество книг, которое надо вывести

Формат ответа:

```
{
      'count': 390,
      'books': [
            BOOK,
            …
      ]
}
```

Список книг автора — GET /api/books/author/ID

Метод выводит книги, привязанные к автору, чей ID был передан в запросе, в порядке убывания даты публикации — от самой новой до самой старой.

Параметры:

- offset — сдвиг от 0 для постраничного вывода
- limit — количество книг, которое надо вывести

Формат ответа:

```
{
      'count': 390,
      'books': [
            BOOK,
            …
      ]
}
```

Запрос подтверждения контакта — POST /api/requestContactConfirmation

Метод проверяет корректность переданного контакта, — e-mail или телефона, — а затем, если это первый запрос кода для этого контакта, генерирует и отправляет на данный контакт код подтверждения. Также в базе данных отмечается дата и время его отправки, а количество попыток выставляется равным 0.

Если запрос кода подтверждения повторный, то, в случае если срок действия кода подтверждения не истёк и количество попыток его ввода не исчерпано:

- Если превышен таймаут повторной отправки кода, на переданный контакт отправляется текущий код подтверждения.

- Если таймаут повторной отправки кода не превышен, возвращается ошибка: “Повторно запросить код можно через N минут”, где N — время в минутах, после которого можно запросить код подтверждения повторно.

Если исчерпано количество попыток ввода кода, то проверяется, прошло ли достаточно времени с момента генерации кода подтверждения. Если прошло достаточно, генерируется новый код. Если же нет — возвращается сообщение об ошибке: “Исчерпано количество попыток ввода кода подтверждения. Запросите новый код через N минут”, где N — время в минутах, оставшееся до момента, когда можно будет запросить новый код.

Если же истёк срок действия кода подтверждения, а количество попыток ввода не превышает лимит, генерируется и отправляется новый код.

Предлагается установить в конфигурации приложения следующие параметры:

- Максимальное количество попыток ввода кода — 3
- Срок действия кода подтверждения — 10 минут
- Таймаут, после которого код можно запросить повторно, — 5 минут

Параметры:

- contact — e-mail или телефон, на который нужно отправить код подтверждения

Формат успешного ответа:

```
{
     ‘result’: true
}
```

Формат ответа в случае ошибки:

```
{
     ‘result’: false,
     ‘error’: “E-mail указан неверно”
}
```

Отправка кода подтверждения — POST /api/approveContact

Метод проверяет, код подтверждения контакта. Код подтверждения может быть просрочен, может быть исчерпано количество попыток его ввода, а

также код может быть введён неверно: в этом случае, возвращается ошибка. В первых двух случаях также должен отправлять параметр “return”, равный “true”, который означает, что необходимо предоставить пользователю возможность запросить новый код. В противном случае контакт активируется (user_contact.approved становится равным 1), а метод возвращает значение “result”, равное “true”.

Параметры:

- contact — e-mail или телефон
- code — код подтверждения e-mail’а или телефона

Формат успешного ответа:

```
{
     ‘result’: true
}
```

Формат ответа в случае ошибки:

```
{
     ‘result’: false,
     ‘return’: true,
     ‘error’: “Код подтверждения устарел. Запросите новый”
}
```

Отправка отзыва на книгу — POST /api/bookReview

Метод отправляет отзыв на книгу авторизованным пользователем.

Параметры:

- bookId — ID книги, на которую отправляется отзыв
- text — текст отзыва

Формат ответа:

```
{
     ‘result’: true
}
```

Формат ответа в случае ошибки:

```
{
     'result': false,
     'error': "Отзыв слишком короткий. Напишите, пожалуйста,
более развёрнутый отзыв"
}
```

Список транзакций — GET /api/transactions

Метод возвращает список транзакций текущего пользователя.

Параметры:

- sort — сортировка транзакций, может быть “asc” или “desc” (по умолчанию, “asc”, если значение не указано)
- offset — сдвиг от 0 для постраничного вывода
- limit — количество книг, которое надо вывести

Формат ответа:

```
{
     'count': 390,
     'transactions': [
          {
               'time': "1595942297",
               'value': -300,
               'description': "Покупка книги "Интервью Петра I
          телеканалу RTVI""
          },
          …
     ]
}
```

Список книг по тэгу — GET /api/books/tag/ID

Метод выводит книги, привязанные к тэгу, чей ID указан в запросе, отсортированные по убыванию даты публикации — от самых новых до самых старых.

Параметры:

- offset — сдвиг от 0 для постраничного вывода
- limit — количество книг, которое надо вывести

Формат ответа:

```
{
      'count': 390,
      'books': [
            BOOK,
            …
      ]
}
```

Изменение статуса книги — POST /api/changeBookStatus

Метод изменяет статус одной или нескольких книг, ID которых переданы в массиве bookIds, на тот, который также указан в запросе. Изменение статуса происходит только в соответствии с “Принципами откладывания, перемещения в корзину и покупки книг” (см. ниже).

Параметры:

- bookIds — массив идентификаторов книг, статус которых необходимо изменить по отношению к текущему пользователю
- status — новый статус книги, варианты:
    - UNLINK — удаление связки книги с пользователем
    - KEPT — изменение статуса на “Отложена”
    - CART — изменение статуса на “В корзине”
    - ARCHIVED — изменение статуса на “В архиве”

Формат успешного ответа:

```
{
      'result': true
}
```

Формат ответа в случае ошибки:

```
{
```

```
    'result': false,
    'error': "Нельзя отложить купленную книгу"
}
```

Лайк или дизлайк отзыва на книгу — POST /api/rateBookReview

Метод добавляет лайк или дизлайк отзыву книги. В случае успешного лайка или дизлайка возвращается result, равный true. В противнов случае — равный false

Параметры:

- reviewId — ID отзыва
- value — оценка: лайк (1) или дизлайк (-1)

Формат ответа:

```
{
    'result': true
}
```

Оценка книги — POST /api/rateBook

Метод добавляет оценку книги. В случае успешного добавления оценки возвращается result, равный true. В противнов случае — равный false

Параметры:

- bookId — ID отзыва
- value — оценка от 1 до 5

Формат ответа:

```
{
    'result': true
}
```

Поиск книг — GET /api/search

Метод выводит книги, найденные по переданному поисковому запросу. В случае, если поисковый запрос не задан, в count необходимо выводить число -1, а массив books должен быть пустым.

Параметры:

- query — поисковый запрос, по которому выводятся книги
- offset - сдвиг от 0 для постраничного вывода
- limit - количество книг, которое надо вывести

Формат ответа:

```
{
     'count': 390,
     'books': [
          BOOK,
          …
     ]
}
```

Получение информации об оплате — POST /api/payment

Метод сохраняет информацию о зачислении, произведённом через платёжную систему, в базе данных в таблице с транзакциями.

Параметры:

- hash — хэш пользователя, на счёт которого произошло зачисление
- sum — зачисленная сумма
- time — дата и время зачисления (в формате timestamp, временная зона — UTC).

Формат ответа:

```
{
     'result': true
}
```

Ответы с ошибками и их обработка

Любой метод API может возвращать ошибку, если она произошла. Такие ответы должны быть пустыми и должны сопровождаться соответствующими статус-кодами. Желательно ограничиться использованием кодов 400, 401, 403, 404, 405 и 500 при возникновении соответствующих им типов ошибок. Frontend непредвиденную ошибку должен отображать во всплывающем сообщении (“alert”).

Принципы и алгоритмы

Алгоритм выбора рекомендуемых книг

Алгоритм должен быть спроектирован и разработан студентом самостоятельно на основании следующих принципов:

- Если пользователь не авторизован и не добавлял никакие книги в корзину или в отложенные, то рекомендации должны строиться на основе тех книг, которые имеют наивысший рейтинг на сайте (изначально нужно для 50% книг сгенерировать данные рейтинга), а также недавно появились.

- Если пользователь авторизован, покупал какие-то книги либо добавлял книги в корзину или в отложенные, то рекомендации должны строиться на основе этих добавлений. В этом случае рекомендуемые книги должны подбираться по тэгам, жанрам и авторам книг, к которым пользователь имеет какое-либо отношение, а также по их новизне. Например, пользователю должна отображаться в рекомендациях новая недавно вышедшая книга автора, книги которого он уже покупал.

Алгоритм расчета рейтинга книг

Рейтинг книги представляет собой число от 1 до 5, которое рассчитывается как среднее значение всех оценок пользователей данной книги, которые тоже могут быть равны от 1 до 5. Среднее значение округляется и получается рейтинг книги.

Примеры:

- Оценки пользователей: 5, 3, 2, 5, 1, 1, 5, 4. Среднее значение равно 26/8 = 3,25, то есть рейтинг равен 3.
- Оценки пользователей: 5, 2. Среднее значение 7/2 = 3,5, то есть рейтинг равен 4.

В случае, если ни один пользователь не оценил книгу, её рейтинг равен 0. Студенту необходимо самостоятельно спроектировать и реализовать таблицу в базе данных для хранения рейтинга книг.

Алгоритм расчёта рейтинга отзывов

Рейтинг отзыва рассчитывается как разница между количеством лайков и дизлайков отзыва. Если нет ни одного лайка и ни одного дизлайка, рейтинг равен 0.

Алгоритм определения популярности книг

Популярность книги представляет собой неотрицательное число, которое можно рассчитать по следующей формуле:

$$P = B + 0,7*C + 0,4*K,$$

где B — количество пользователей, купивших книгу, C — количество пользователей, у которых книга находится в корзине, а K — количество пользователей, у которых книга отложена.

Алгоритм работы платёжной системы

Платёжная система — внешний сервис, который принимает платежи, зачисляет их на банковский счёт компании, а параллельно сообщает сайту, на котором происходит покупка (или пополнение счёта), что платёж принят от клиента.

При нажатии на сайте кнопки “Пополнить счёт” или “Оплатить” должен происходить переход к платёжной системе по адресу: /payment/?sum=SUM&user=HASH, где SUM — сумма, на которую требуется пополнить счёт, а HASH — хэш пользователя, который пополняет себе счёт (поле user.hash).

После пополнения счёта платёжная система возвращает пользователя на ту же страницу, с которой пользователь пришёл, методом GET или POST с параметром result, который может быть равен “true” или “false”, а также, если “result” равен “false”, с параметром “error”, который содержит текст ошибки.

В случае успешной оплаты платёжная система также отправляет запрос к API (см. “Получение информации об оплате” в документации по API).

См. также описания страниц “Пополнение счёта” и “Корзина”.

В качестве первой реализации необходимо использовать тестовую платёжную систему. В качестве последующей — настоящую в тестовом режим. Для настоящей платёжной системы студентам будет необходимо внести изменения в приложение (в зависимости от требований платёжной системы).

Принципы откладывания, перемещения в корзину и покупки книг

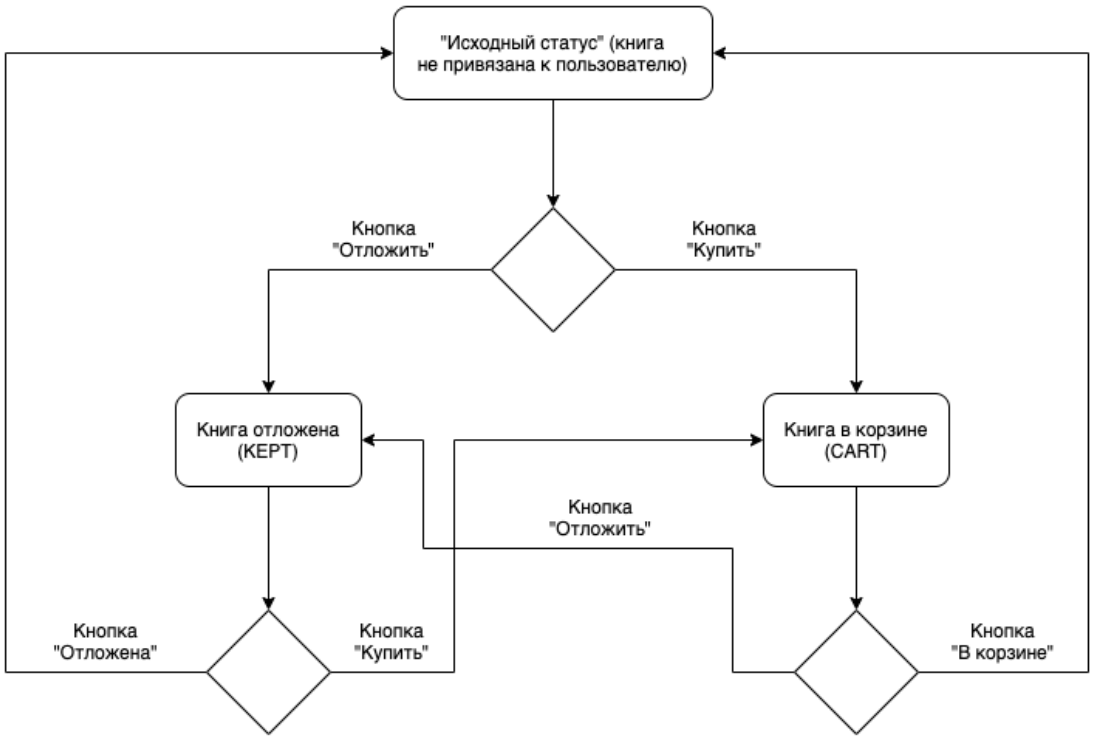
Пользователь сайта может быть авторизован, а может не быть авторизован. Если он не авторизован, он так же, как и авторизованный, может откладывать книги или добавлять их в корзину, но для него остаётся недоступным раздел “Мои книги”, и соответствующие пункты меню не отображаются.

Если пользователь не авторизован, создаётся “пустой” пользователь для данной сессии. Если в дальнейшем пользователь регистрируется, то данные регистрации (e-mail и телефон) привязываются к созданному пользователю. Если же пользователь авторизуется, то все данные “пустого” пользователя сливаются с данными того пользователя, под которым произошла авторизация.

“Пустые” пользователи должны удаляться из базы через 1 месяц (на основании даты регистрации — user.reg_time).

Книга может изменять свой статус в соответствии со следующей диаграммой переходов статусов:

У всех пользователей



Только у авторизованных пользователей

Алгоритм поиска книг

Для поиска книг необходимо самостоятельно установить, настроить и интегрировать с веб-приложением любой поисковый движок, например, Apache Solr (https://lucene.apache.org/solr/guide/7_0/installing-solr.html)
или свое решение. В поиске должны участвовать следующие свойства книг:

- Имена и фамилии всех авторов книги
- Название книги
- Описание книги
- Тэги, к которым привязана книга
- Жанры, к которым привязана книга

Алгоритм скачивания файла

Адрес скрипта скачивания файла — /download/HASH, где HASH — хэш файла, указанный в поле book_file.hash. Скачивание файла возможно только авторизованным пользователем, купившим книгу. В случае, если книга не куплена или пользователь не авторизован, скрипт должен возвращать ошибку 404 с пустым телом ответа.

При скачивании книги счётчик её скачиваний (file_download.count) должен увеличиваться на единицу. При первом скачивании, разумеется, в этой таблице должна появляться запись с количеством скачиваний, равным 1.

Дополнительные задачи “со звёздочкой” для самостоятельного выполнения

Недавно просмотренные книги

Спроектировать и разработать сохранение данных о просмотрах книг пользователем и использовать их в разделе рекомендуемых (на главной и на соответствующей странице) наравне с другими рекомендуемыми книгами, а также в разделе “Популярное” (учитывать в алгоритме расчёта популярности). Кроме того, можно создать страницу “Недавно просмотренные” и выводить эти книги туда.

Ограничение авторизации

Спроектировать и реализовать ограничение авторизация максимум на трёх устройствах. В случае авторизации на четвёртом, информирование об этом пользователя и спрос самой редко используемой авторизации. Также возможность просмотра в профиле пользователя перечня активных сессий с возможностью завершить отдельные или все, кроме текущей.

Телеграм-бот для выбора и покупки книг

Телеграм-бот должен поприветствовать пользователя и предложить ввести ему жанр, автора, название, тэг или свободный поисковый запрос, по

которому вывести книги в виде пронумерованного списка, чтобы пользователь мог добавить их в корзину. Когда корзина сформирована, пользователь может их купить путём перехода по ссылке для оплаты (бот должен её прислать). После оплаты бот выводит ссылки на скачивание книг. Все ссылки, ведущие на сайт, должны содержать хэш для автоматической авторизации данного пользователя. Хэш для каждой ссылки в целях безопасности должен быть разным. Алгоритм формирования и работы такого хэша студент должен разработать самостоятельно.