

**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ**

КОНКУРС «ИТ-СТАРТ – 2020»

**Техническая
документация
программный
продукт «ПО для
библиотеки»**

Школа: №2048

Капитан: Сайед Кирилл Русланович

Команда: Сапожков Егор
Александрович, Овсепян Андрей
Ервандович, Гайдарлы Роман
Николаевич, Ноянов Михаил
Юрьевич

Руководитель: Аминов Евгений Витальевич

Оглавление

Название и формулировка задания3

Развернутое рассмотрение возможных способов решения задачи, их анализ, плюсы и минусы,
возможные проблемы при реализации5

Выбранный участниками способ решения задачи, причины выбора именно этого способа7

Архитектура программного продукта.....7

Используемые технологии 10

Фотография продукта 11

Инструкция по эксплуатации программного продукта 13

Контрольный пример.....28

Название и формулировка задания

ПО для библиотеки

Техническое задание

В школьной библиотеке внедрена система электронного учета книг.

Основное назначение ПО автоматизации работы библиотеки – повысить производительность организации.

Данное ПО позволяет автоматизировать процесс обслуживания посетителей библиотеки (школьники), формирования документов, ведение и учет библиотечного фонда.

Состав ПО: база данных книг, личные кабинеты пользователей, интерфейс.

Каждому школьнику для удобства пользования библиотекой создан личный кабинет, где указаны дата выдачи и название книги, а также использование электронных ресурсов (книг).

Для библиотекаря создан личный кабинет администратора библиотеки с определенным набором функций:

- добавление новых книг
- списаний старых книг
- прием/выдача книг от школьников
- составление различных отчетов по выбранным временным отрезкам

Требования к программному продукту

1. Создание новой записи выдачи книг школьнику
2. Открытие существующих записей выданных книг с возможностью изменения статус на “книга возвращена” и фиксированием времени выдачи/приема.
3. Редактирование записи, документа, путем ввода, замены, удаления содержимого файла.
4. Сохранение записи в базу данных.
5. Формирование отчетов.
6. Разграничение доступа.

Требования к организации входных данных:

Входные данные вводятся в соответствующие поля ввода в программном продукте или загружаются из базы данных.

Требования к организации выходных данных:

Выходные данные выводятся в соответствующие поля вывода в

программном продукте или сохраняются в базу данных.

Из данных можно формировать отчеты, сохранять их в отдельном файле, печатать.

Требования к защите информации и программного продукта:

В разработанном программном продукте должно быть обеспечено разграничение прав доступа. Предусмотрено 3 роли:

Системный администратор

- Имеет доступ к визуальным настройкам системы
- Имеет доступ к настройкам безопасности системы
- Не имеет право изменять данные школьников, записей о выдаче. приеме книг от школьников, библиотечному фонду.

Библиотекарь:

- Имеет доступ к подсистеме библиотечного фонда
- Может разрабатывать документы, отчеты
- Имеет доступ к подсистеме учета приема. выдачи книг школьникам, подсистеме учета школьников

Школьник

- Имеет доступ к своему личному кабинету с возможностью изменения персональной информации и контролем взятых/сданных книг в библиотеке.

В базе данных должны быть определены:

- состав таблиц: по каждой таблице – поля, размерность полей, тип полей;
- взаимосвязь таблиц: ключевые атрибуты;
- структура: нарисовать структуру базы данных (рисунок в векторном редакторе).

Контрольный пример должен обеспечить проверку функционирования ПО, в том числе действий, выполняемых пользователями в процессе эксплуатации, и реакции ПО на действия пользователей.

Развернутое рассмотрение возможных способов решения задачи, их анализ, плюсы и минусы, возможные проблемы при реализации

Для решения данной задачи подходят несколько способов:

1. Написание C++ приложения на базе фреймворка Qt или с использованием windows forms;
2. Написание C# приложения с использованием windows forms;
3. Написание Python приложения на базе фреймворка Qt;
4. Написание web приложения на базе фреймворка Django.

Первый способ:

Плюсы:

1. Лучшая производительность;
2. Кроссплатформенность;
3. Гибкая кастомизация.

Минусы:

1. Написание приложения на данных технологиях требует большого количества времени;
2. Обучение команды для написания приложения на данных технологиях занимает много времени.

Второй способ:

Плюсы:

1. Хорошая производительность;
2. Гибкая кастомизация.

Минусы:

1. Нет кроссплатформенности;
2. Производительность ниже, чем при написании приложения на C++;
3. Нет поддержки старых версий windows.

Третий способ:

Плюсы:

1. Кроссплатформенность;
2. Гибкая кастомизация.

Минусы:

1. Производительность ниже, чем при написании приложения на C++.
2. Обучение команды для написания приложения на данных технологиях занимает много времени.

Четвертый способ:

Плюсы:

1. Кроссплатформенность;
2. Возможность запуска приложения на сервере и получения доступа к нему через локальную сеть;
3. Хорошая производительность;
4. Простота подключения базы данных;
5. Возможность написания интерфейса на HTML5 + CSS3;
6. Не требует много времени для изучения.

Минусы:

1. Производительность ниже, чем при написании приложения на C++;
2. Требуется постоянная поддержка системных администраторов.

Рассмотрев данные способы, наша команда выбрала четвертый.

Выбранный участниками способ решения задачи, причины выбора именно этого способа

Фреймворк Django используется для написания быстрых и защищенных сайтов. Данный фреймворк предоставляет защиту от таких атак, как SQL инъекция и других. Так же, Django имеет очень простой и минималистичный ORM, который позволяет максимально быстро построить базу данных. Наша команда использует SQLite, который Django использует по умолчанию. Однако, можно сменить СУБД на PostgreSQL, с которым данный фреймворк будет так же прекрасно работать. Далее, Django предоставляет огромное количество библиотек для аутентификации, например, Django-allauth. Также, Django по умолчанию включает в себя удобную админ-панель, которую можно легко кастомизировать.

Архитектура программного продукта



Весь код нашего проекта: https://github.com/PickBas/library_service. Каждый Django проект имеет папку с настройками проекта, в данном случае "library_service", папку "templates", где хранятся все html файлы, папки приложений, в нашем случае: "user_profile", "book", "core" – папки "static" и "media". В папке "static" хранятся js, css, некоторые png/jpeg файлы, такие как static/favicon.png. "media" используется для хранения более тяжелых файлов.

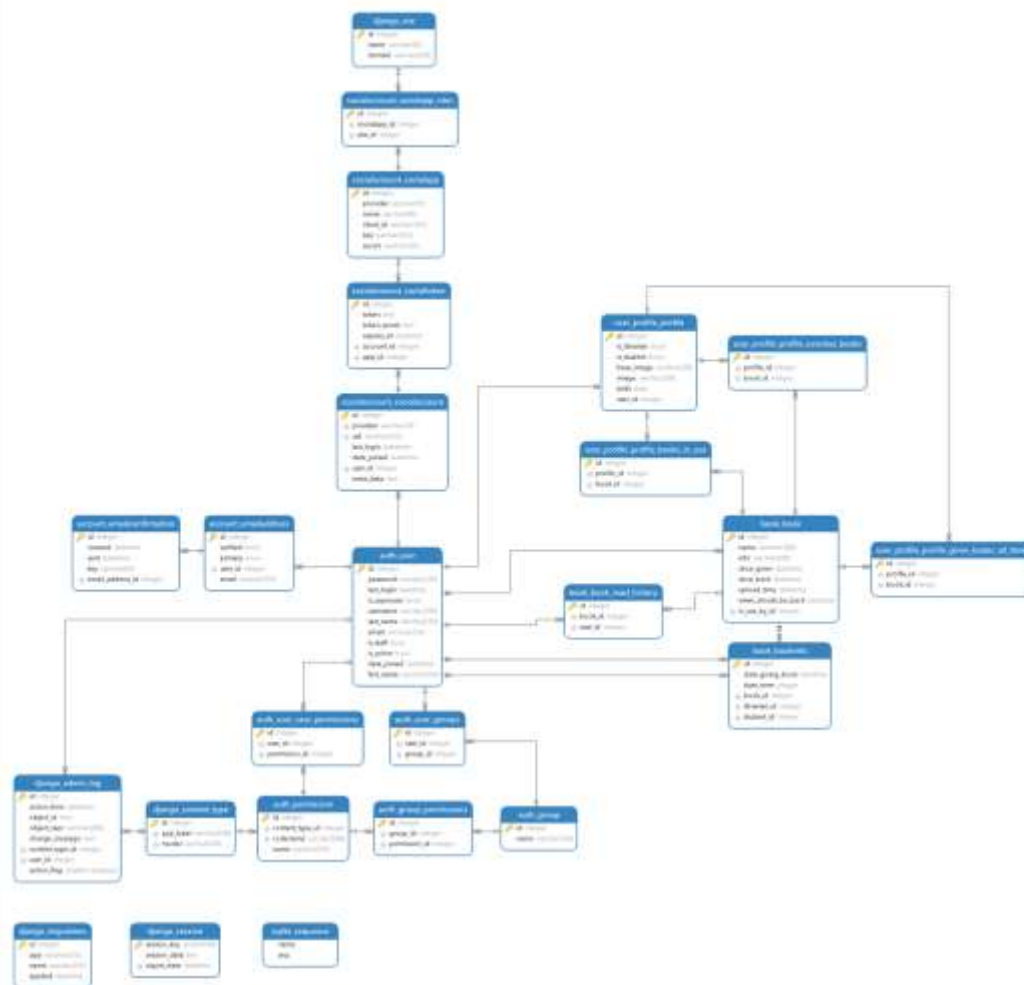
Папка "library_service" включает в себя файлы конфигурации проекта, в том числе "urls.py", где описаны все URLs проекта, и "settings.py", где описаны основные настройки проекта, такие как регион, способы

аутентификации, секретные ключи и прочее.

Папки приложений включают в себя папку migrations, в которой описываются все миграции базы данных; файл “admin.py”, где настраивается админ-панель для текущего приложения; “apps.py”, где описывается конфигурация приложения; файл “forms.py”, где находятся Django формы; файл “models.py”, где описаны модели для базы данных; “tests.py” используется для тестов текущего приложения; “urls.py”, в котором описаны URLs для текущего приложения, которые импортируются в файл “library_service/urls.py”; “views.py”, в котором описаны все View классы для текущего приложения.

В отличие от многих других web фреймворков, Django использует паттерн MVT (Model View Template), а не MVC (Model View Controller). В данном паттерне Views заменяют Controllers, поэтому вся логика для возвращения ответа клиенту описана в файлах приложений “views.py”.

Схема БД (Так же можно посмотреть [здесь](#)):



Используемые технологии

Для написания данного веб-сайта нами были применены данные технологии:

- Python 3
- Django
- HTML5
- CSS3
- JavaScript
- Bootstrap 4
- SQLite 3

- Python 3

Python 3 – это интерпретируемый, высокоуровневый язык программирования, обладающий огромным количеством библиотек, одна из которых – Django.

- Django

Django – это web фреймворк с открытым исходным кодом. Он был написан на языке Python. Используемый паттерн – MVT (Model View Template).

- Html5

Html5 - язык верстки, используемый для верстки страниц и для отображения контента на них.

- CSS

CSS - Для создания приятного интерфейса нами был использован язык отвечающий за визуальное представление документов CSS.

- JavaScript

JavaScript – это высокоуровневый язык программирования, имеющий огромное количество библиотек и фреймворков. Он был использован для создания формы загрузки фотографий и для отправки POST запросов на сервер через Ajax.

- Bootstrap 4

Bootstrap 4 – CSS фреймворк, который расширяет нам набор инструментов для создания нашего веб-сайта, так же содержит шаблоны оформления для типографии HTML и CSS.

- SQLite 3

SQLite – это СУБД, которая по умолчанию используется в Django.

Фотография продукта

В нашем проекте мы применили простой и строгий дизайн. Мы использовали резко контрастирующие друг с другом цвета и оттенки (рис.1), что позволило нам подчеркнуть важность отдельных элементов или информации.

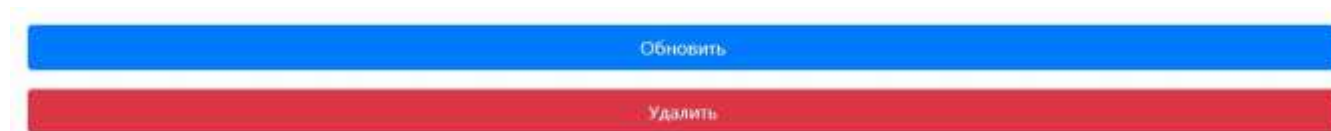


Рисунок 1. Контрастирующие элементы сайта.

В качестве шапки для таблицы с книгами был использован черный цвет (рис.2). Он помогает заострить внимание посетителя сайта на данном элементе. Также черный цвет создает ощущение элегантности и минимализма, что хорошо подходит к нашему проекту.

Книги

Добавить книгу			
№	Название	Статус	Добавлено
1	Л.Н. Толстой "Война и Мир"	Не доступна	19:12, 12/10/2020
2	Пушкин Евгений Онегин	Доступна	19:12, 12/10/2020

Рисунок 2. Таблица

Для кнопок сайта были выбраны 2 контрастирующих между собой цвета – голубой и серый. Голубой цвет в данном случае показывает важность именно этой кнопки. Серый же цвет отодвигает кнопку на второй план, показывая тем самым посетителю, что кнопка серого цвета является второстепенной относительно голубой. На рис.3 мы можем увидеть наглядный пример. Кнопка “Обновить фото” в данном случае для посетителя не так важна, как кнопка редактировать профиль. Сперва нужно указать ФИО и дату рождения, чтобы

библиотекарю было удобнее выдавать ученикам книги. А фото не является обязательным условием для выдачи книги. Именно поэтому кнопки разных цветов.

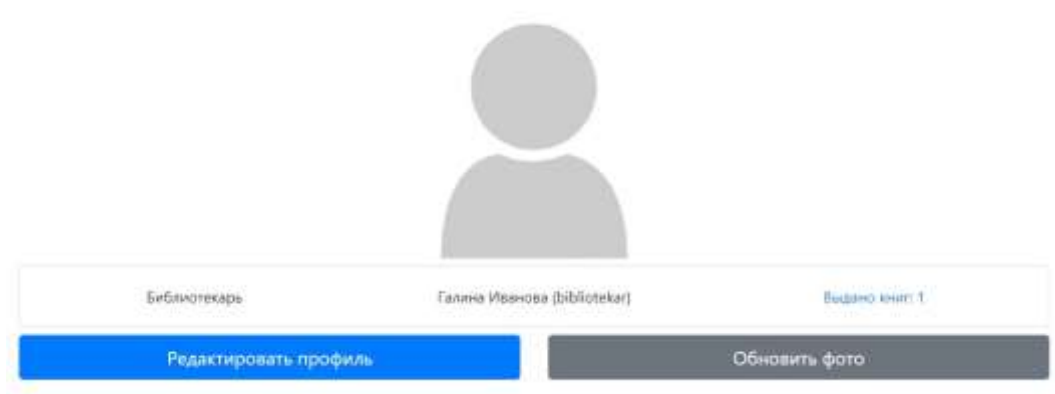


Рисунок 3. Контрастирующие кнопки.

Фон для сайта был выбран белый (рис.4). На белом фоне отчетливо видны все элементы сайта. Также светлые и пастельные тона легче воспринимаются человеческим глазом, и текст становится более читабельным. Белый цвет не отвлекает внимание пользователя и позволяет сосредоточить его поведение на более важных вещах – к призывам к действию, важной информации, в нашем случае – к выдаче или получению книг.

A registration form titled 'Регистрация' (Registration). It contains several input fields: 'E-mail*' (with a sub-label 'E-mail адрес'), 'Имя пользователя*' (with a sub-label 'Имя пользователя'), 'Имя*' (with a sub-label 'First name'), 'Фамилия*' (with a sub-label 'Last name'), 'Пароль*' (with a sub-label 'Пароль'), 'Пароль (еще раз)*' (with a sub-label 'Пароль (еще раз)'), and 'Секретный ключ'. At the bottom is a blue button with the text 'Зарегистрироваться' (Register). Below the button is a small link that says 'Еще есть аккаунт? Войти' (Still have an account? Log in).

Рисунок 4. Белый фон.

Инструкция по эксплуатации программного продукта

Первым делом, конечно же, стоит зарегистрироваться. Необходимо вписать свои данные в соответствующие окна: почту, имя пользователя, имя, фамилию, пароль, повтор пароля и секретный ключ (если он известен). Рис1 Обязательные поля для заполнения помечены звездочкой *. Секретный ключ не обязателен, он необходим для регистрации пользователя- библиотекарь, как в нашем примере.

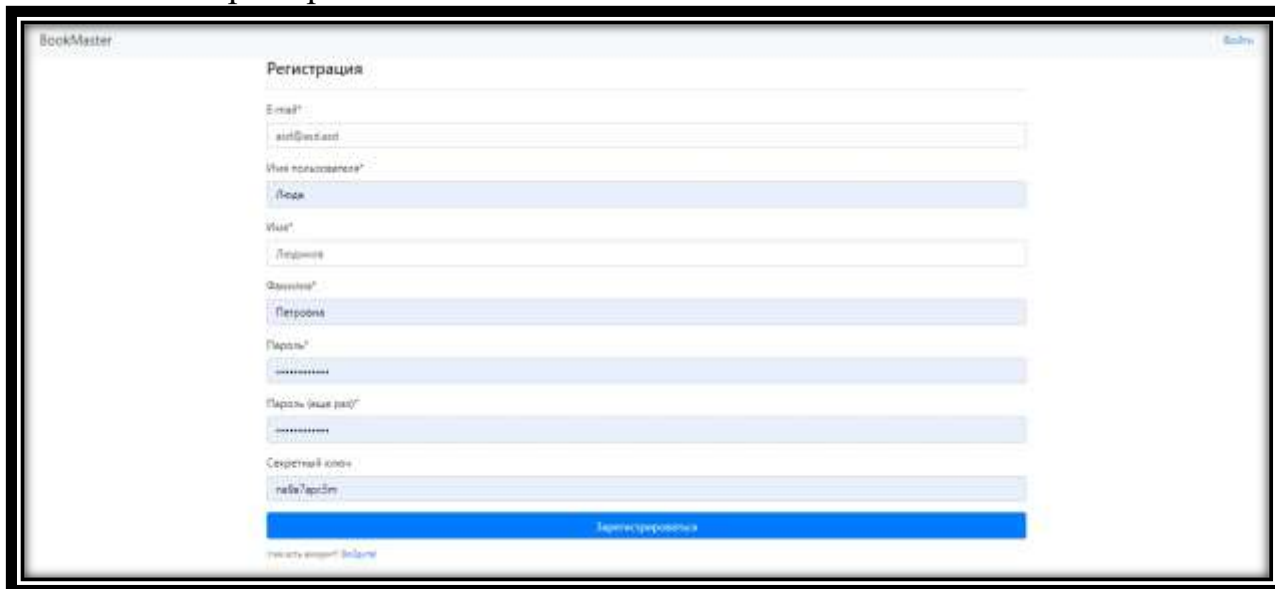
The screenshot shows the 'Регистрация' (Registration) page of the 'BookMaster' application. The page has a light blue header with the 'BookMaster' logo on the left and a 'Войти' (Login) link on the right. The main content area contains several input fields: 'Email*' (with 'mail@bookland' entered), 'Имя пользователя*' (with 'Людя' entered), 'Имя*' (with 'Петровна' entered), 'Фамилия*' (with 'Петровна' entered), 'Пароль*' (with masked characters), 'Пароль (повтор)' (with masked characters), and 'Секретный ключ' (with 'lib@bookland' entered). A blue 'Зарегистрироваться' (Register) button is at the bottom. At the very bottom, there is a small link: 'уже есть аккаунт? Войти'.

Рис1.

Затем необходимо авторизоваться (войти). Нужно ввести поля соответствующие данные. Рис2

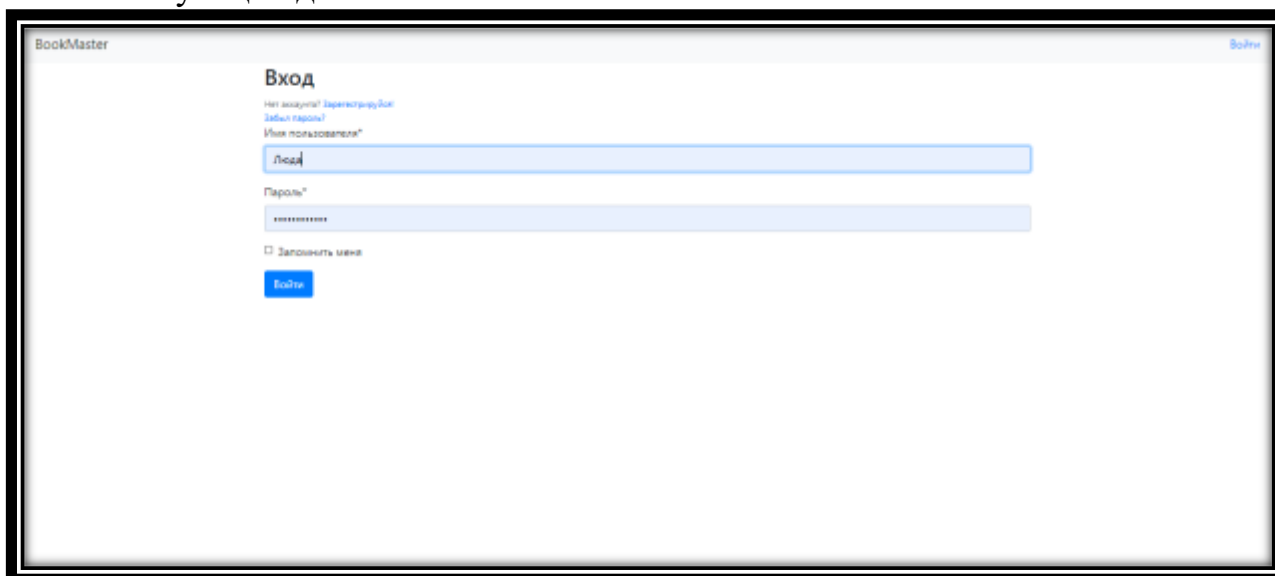
The screenshot shows the 'Вход' (Login) page of the 'BookMaster' application. The page has a light blue header with the 'BookMaster' logo on the left and a 'Войти' (Login) link on the right. The main content area contains the title 'Вход' and two links: 'Нет аккаунта? Зарегистрируйтесь' and 'Забыл пароль?'. Below these are input fields for 'Имя пользователя*' (with 'Людя' entered) and 'Пароль*' (with masked characters). There is a checkbox labeled 'Запомнить меня' (Remember me) and a blue 'Войти' (Login) button at the bottom.

Рис2.

Если вдруг забыли пароль, то его всегда можно восстановить, нажав на «Забыл пароль?» Тогда появится окно (Рис3), введя туда свою почту, вы получите письмо и сможете создать новый пароль.



Рис3.

Войдя на сайт, вы попадаете на главную страницу, где указаны все книги библиотеки и их информация. Рис4. В шапке страницы есть кнопки «Ученики» (пока что их нет) и имя пользователя. Нажмем на имя пользователя и на кнопку профиль.



Рис4.

У нас есть возможность отредактировать свой профиль и добавить фотографию. Также представлена информация о статусе пользователя и количестве выданных книг (пока что ноль). Рис5. Нажмем на «Редактировать профиль»

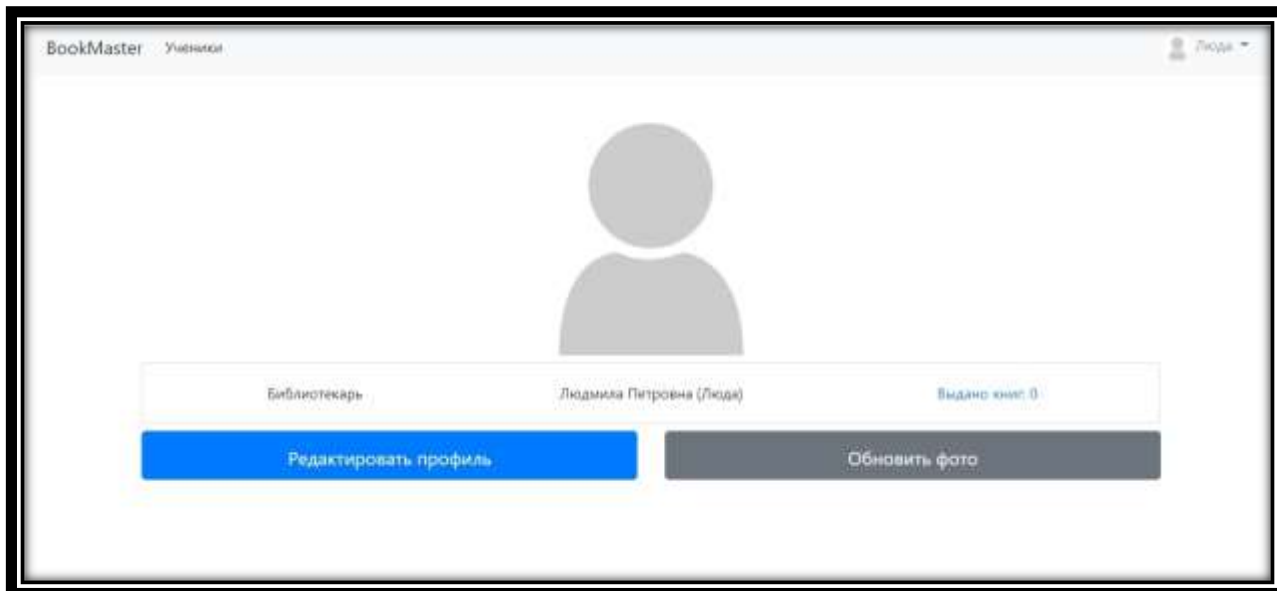


Рис5.

Тут можно поменять имя, фамилию, добавить дату рождения и поменять пароль. Рис6.

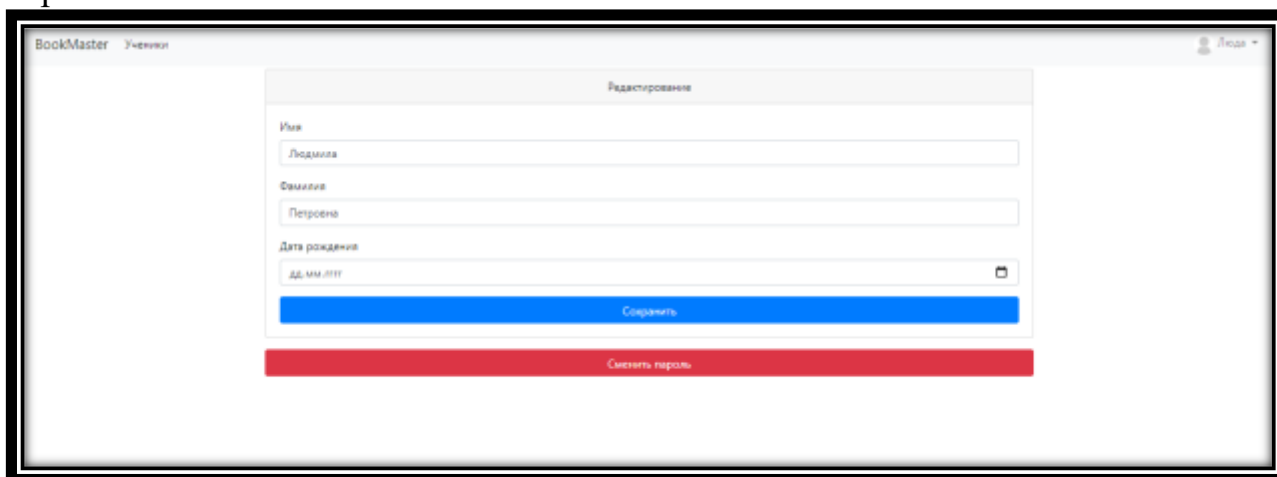


Рис6.

Нажав на «Сменить пароль» вы сможете изменить свой пароль, введя старый. (Рис7)



Рис7.

Вернувшись назад (Рис5) нажмем на обновить фото. Нам предоставляется возможность добавить фотографию и обрезать ее по необходимости. Рис8.

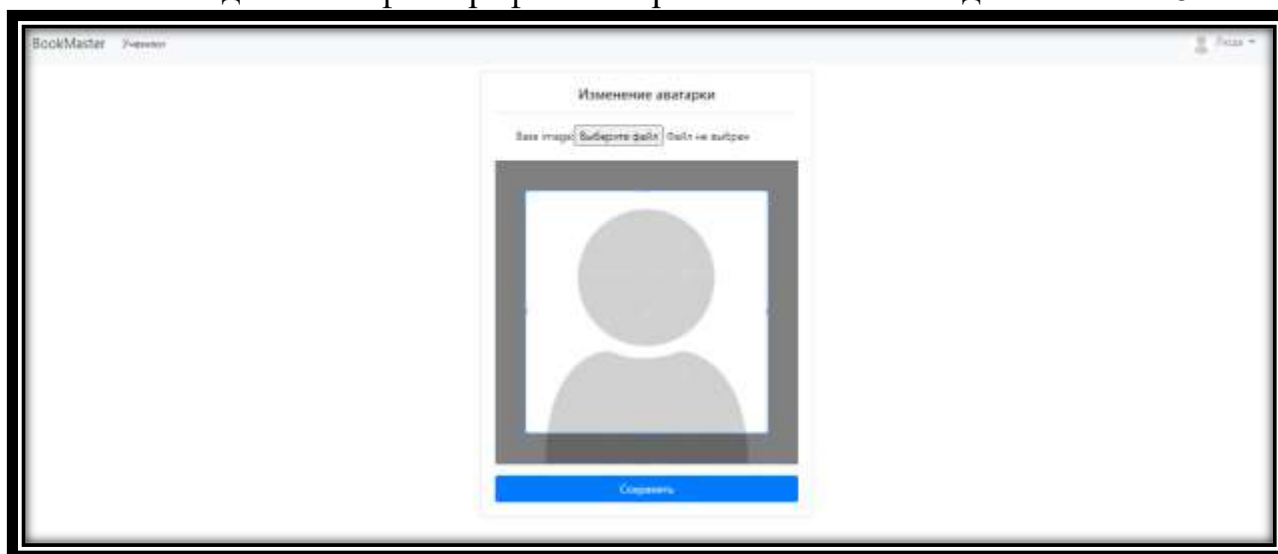


Рис8.

Вернемся на главную страницу (Рис4). Нажмем на существующую книгу. Можно увидеть информацию о книге: доступность, описание, количество прочитавших. Эту книгу можно выписать ученику, отредактировать. Рис9.

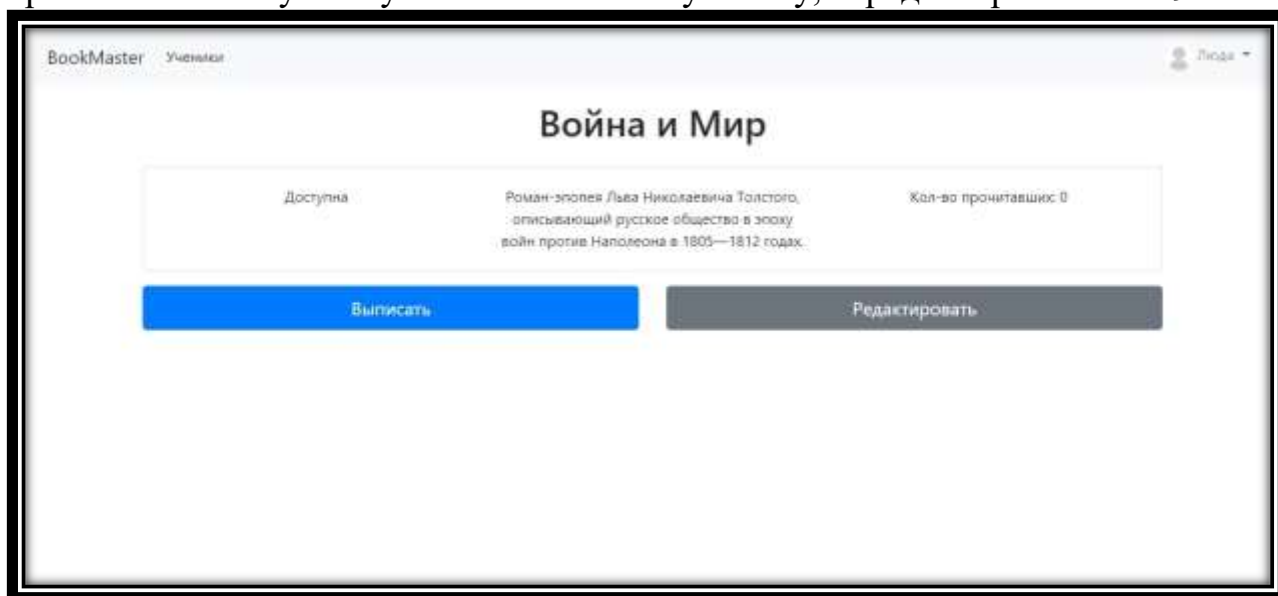


Рис9.

Нажав на «Редактировать» мы можем поменять имя книги и ее описание. Также мы можем её удалить. Рис10.

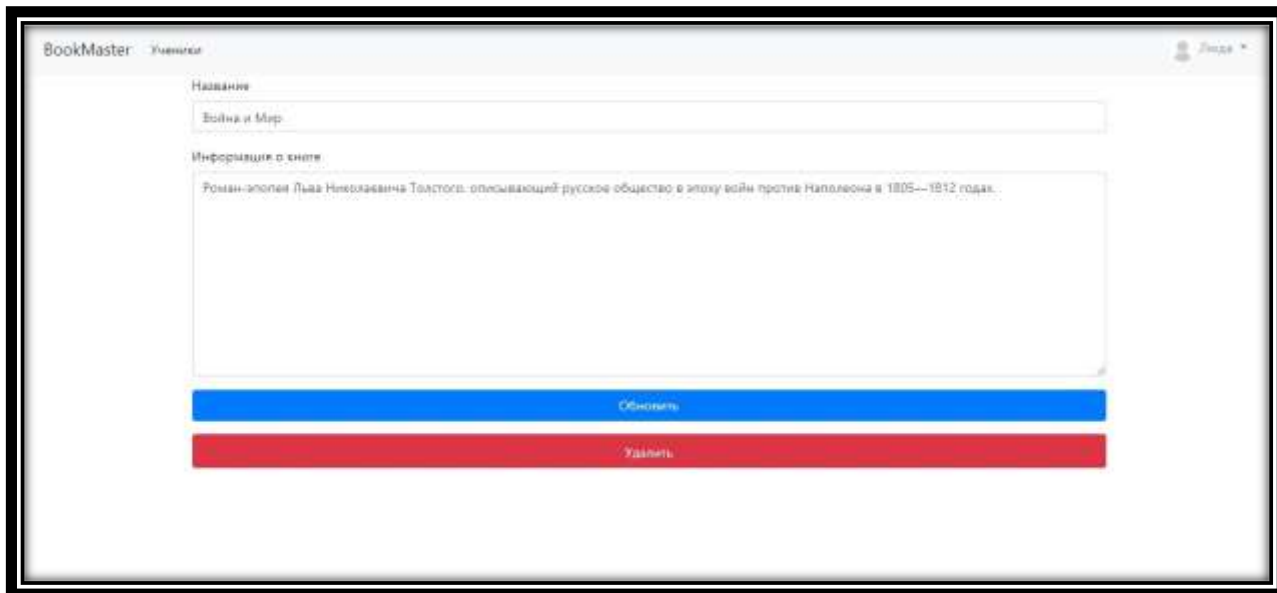


Рис10.

Выйдем на главную страницу и добавим новую книгу нажав на кнопку «Добавить книгу». Рис4. Перед нами представляются несколько полей для заполнения информации новой книги. Рис11.

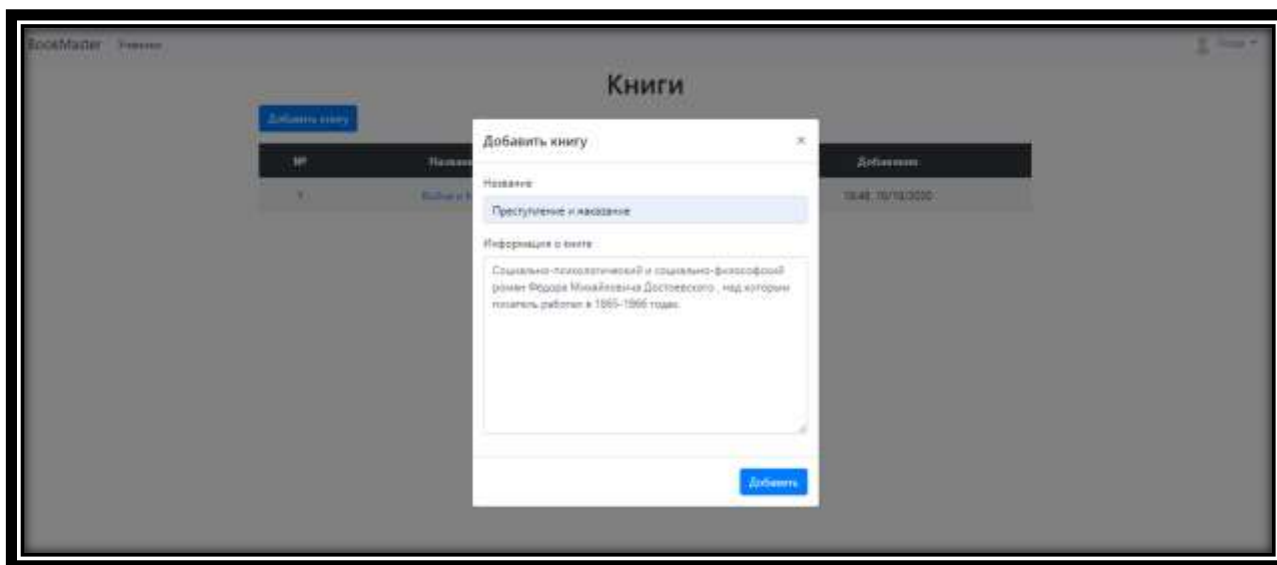


Рис11.

Заполнив информацию новой книги и сохранив ее, мы видим, что она добавилась на главную страницу. Рис12.

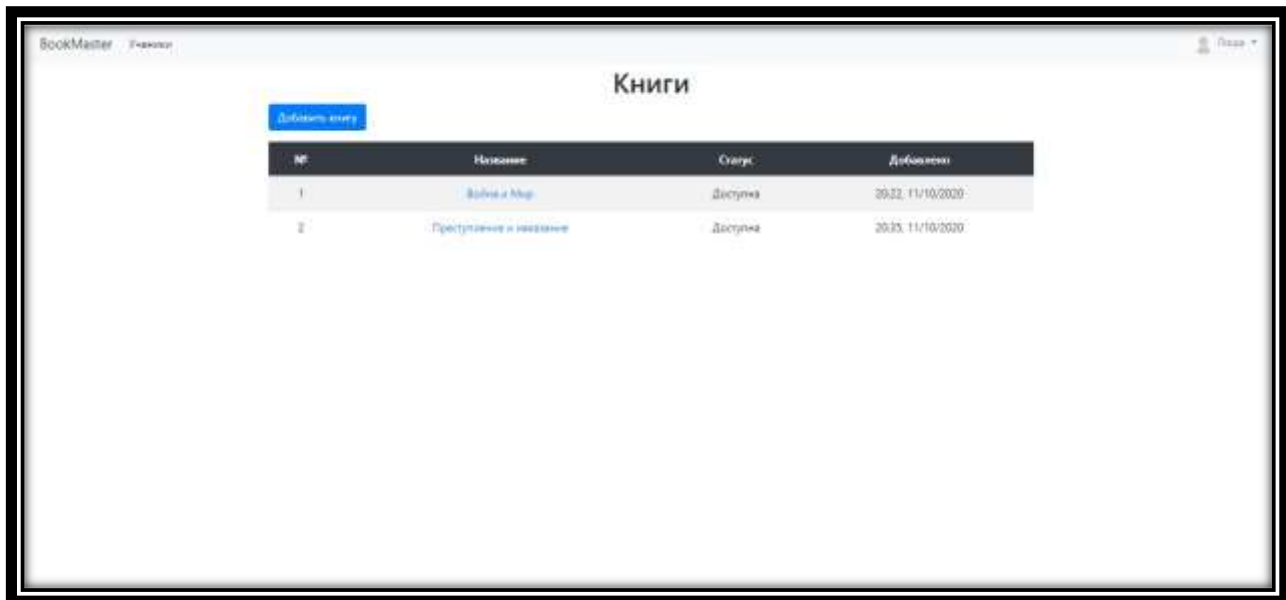


Рис12.

Давайте выйдем с аккаунта библиотекаря и создадим аккаунт ученика. Для этого надо нажать на свое имя и кнопку выйти. Нас спросят, действительно ли мы хотим выйти, и мы выбираем «Выйти». Рис13.

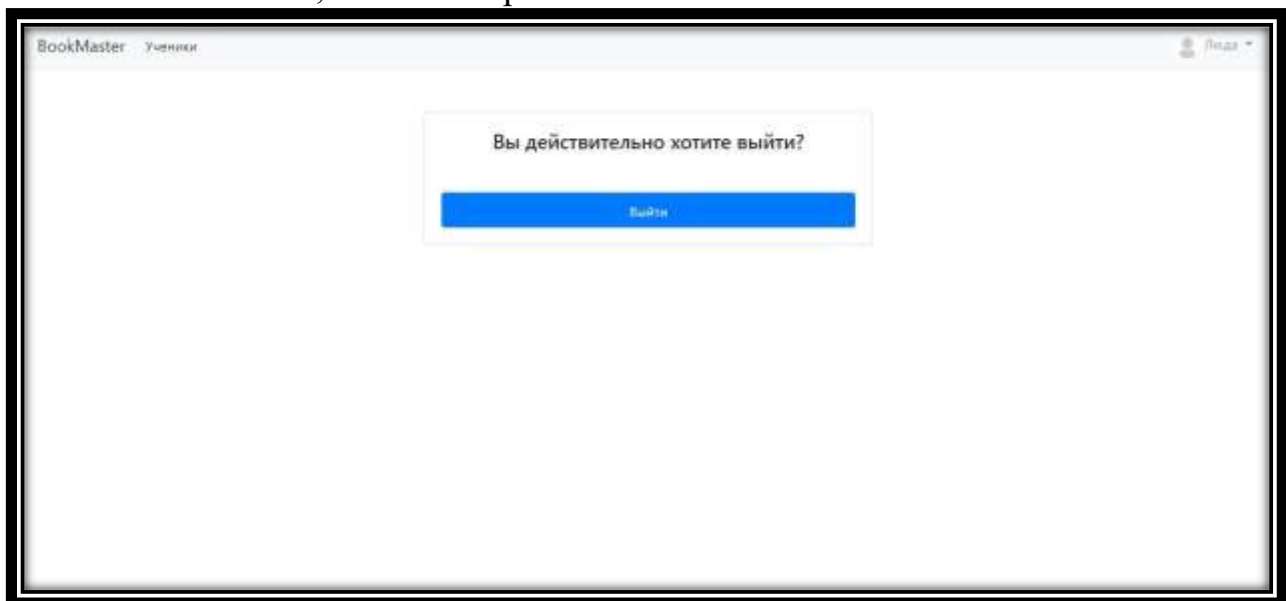


Рис13.

Зарегистрировав нового пользователя-ученика и войдя под его данными, мы попадаем на главную страницу. Рис14. Ученик может посмотреть какие книги в наличии, какие нет и их информацию, нажав на них (Рис9).

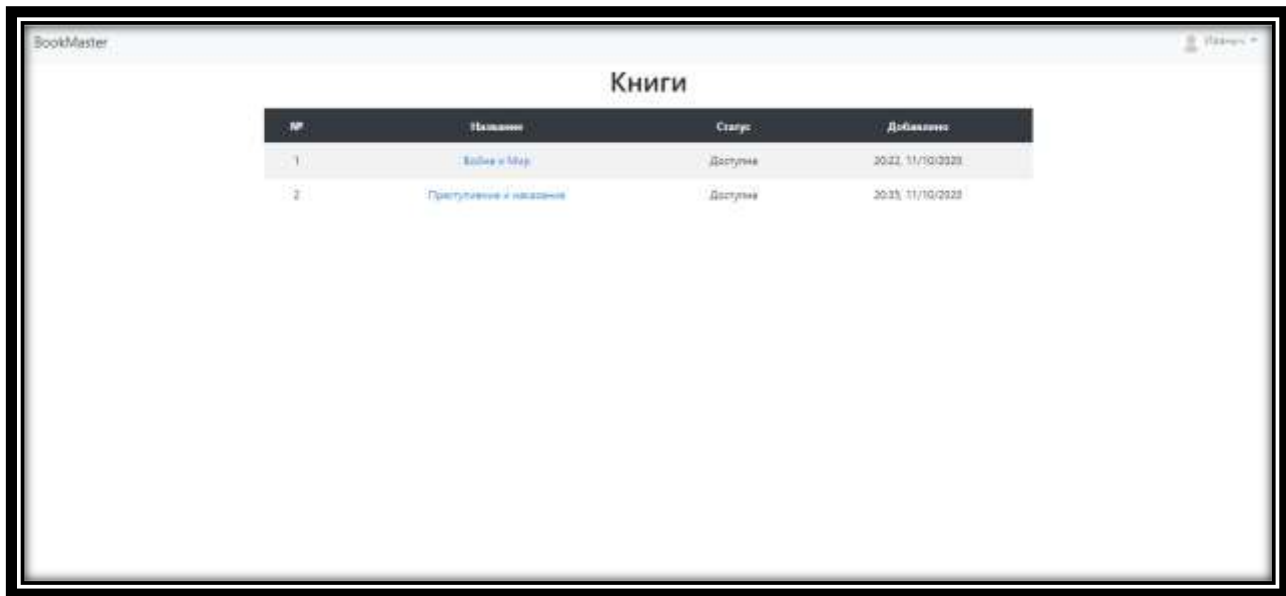


Рис14.

Зайдя в профиль ученика, мы видим его информацию, кнопку «Редактирование профиля» (при нажатии аналогично Рис6) и фотографии (при нажатии аналогично Рис8). Указано количество книг (сейчас 0). Рис15.

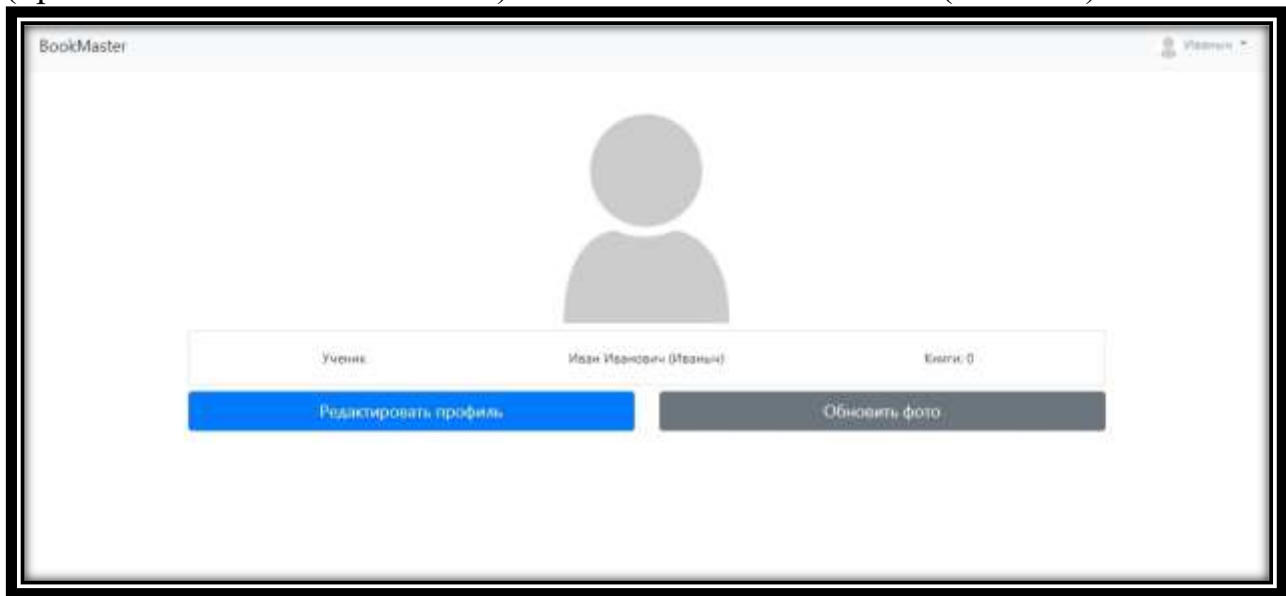


Рис15.

Вернувшись в профиль библиотекаря и нажав на кнопку, «Ученики» мы увидим нового ученика и даже можем посмотреть его данные (Рис17). Рис16.



Рис16.

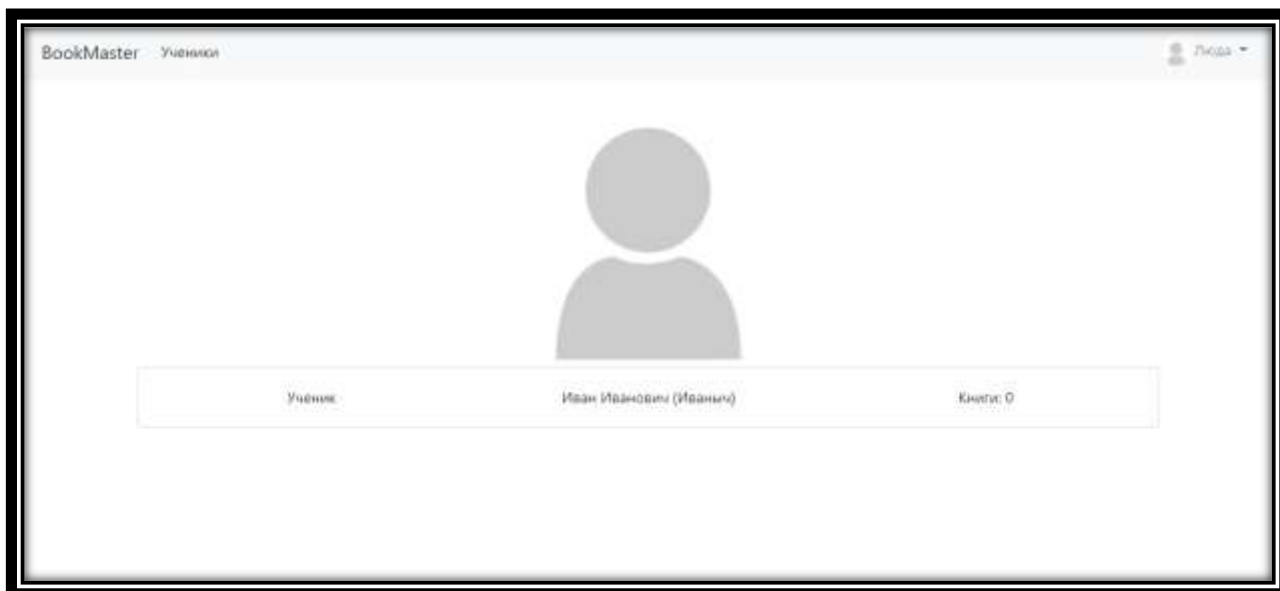


Рис17.

Выдадим этому пользователю книгу. Для этого на главной странице (Рис12) нужно нажать на выдаваемую книгу и затем нажать на кнопку «Выписать».

Рис18.

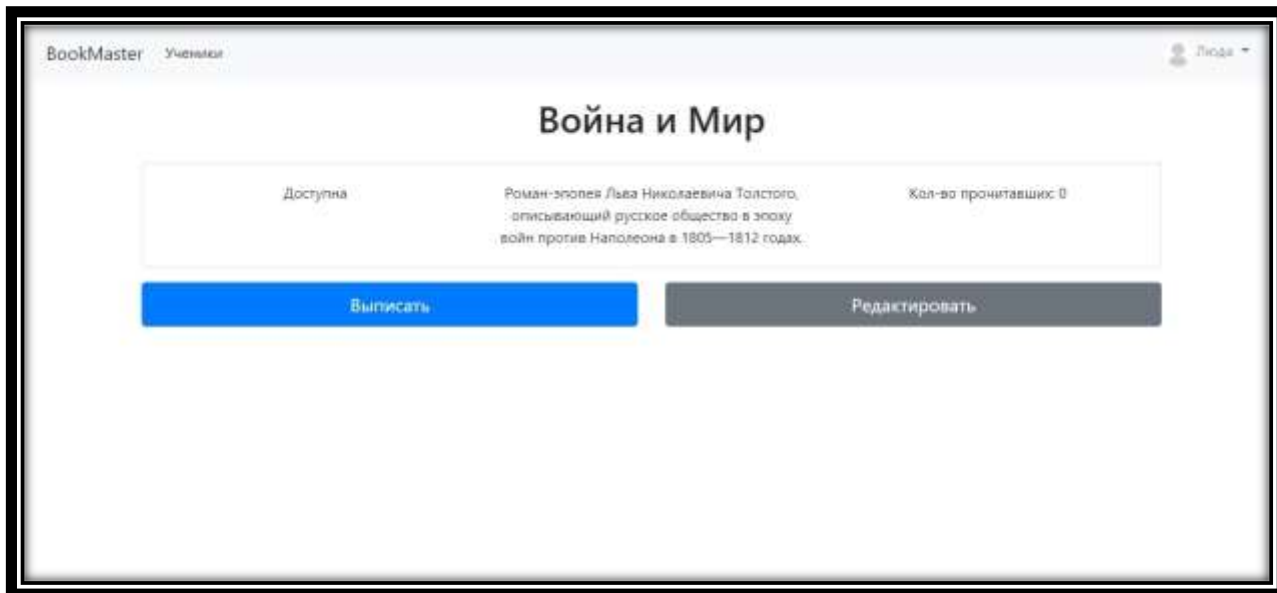


Рис18.

Выбираем пользователя, которому собираемся выдать книгу, выбираем дату, до которой выдаем пользователю книгу и нажимаем кнопку «Выписать»

Рис19.

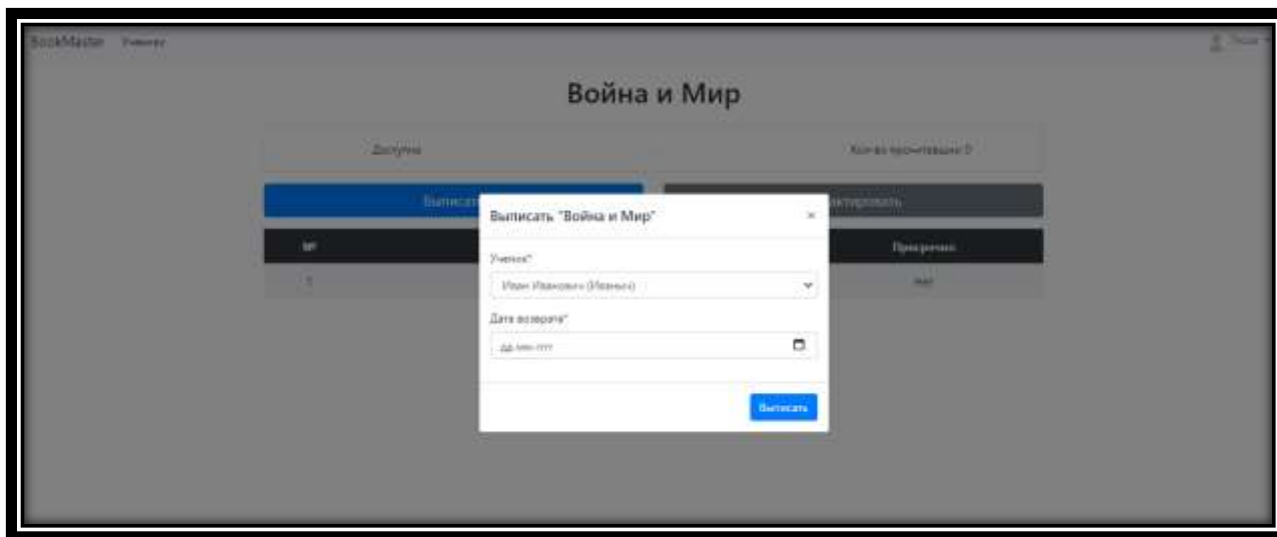


Рис19.

После этого можно увидеть, что эта книга уже не доступна. Рис20.

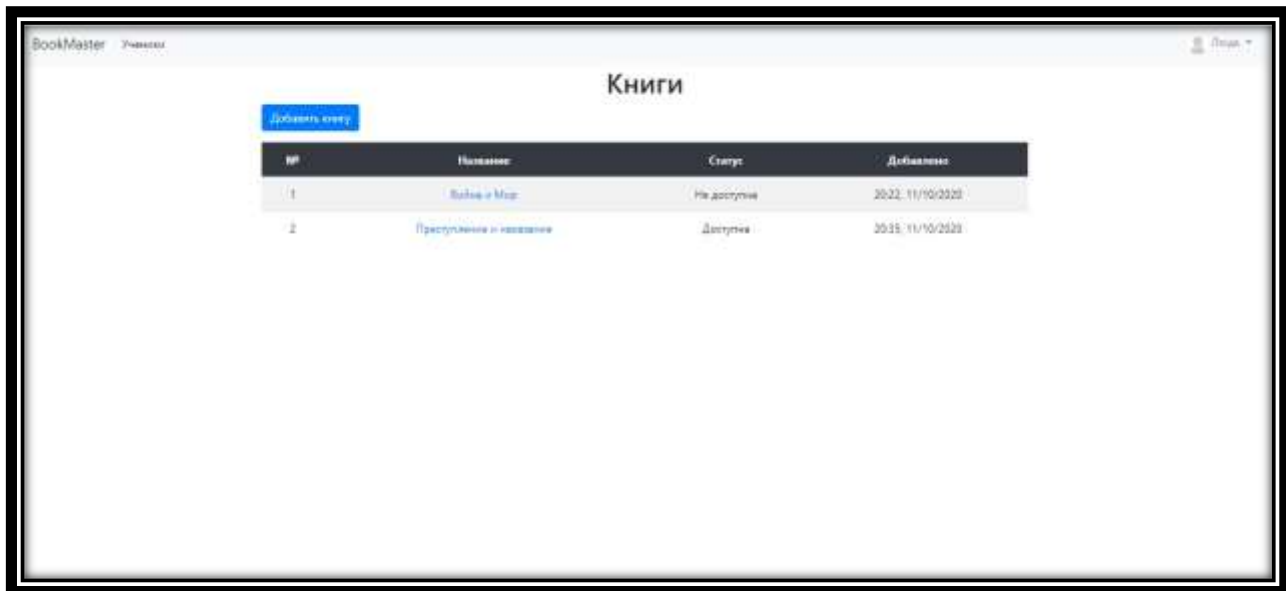


Рис20.

На странице «ученики» у библиотекаря отмечается, что у него добавилась книга. Рис21.



Рис 21.

Тем временем в профиле этого пользователя появляется выданная книга и информация, до какого числа надо вернуть. Рис22.

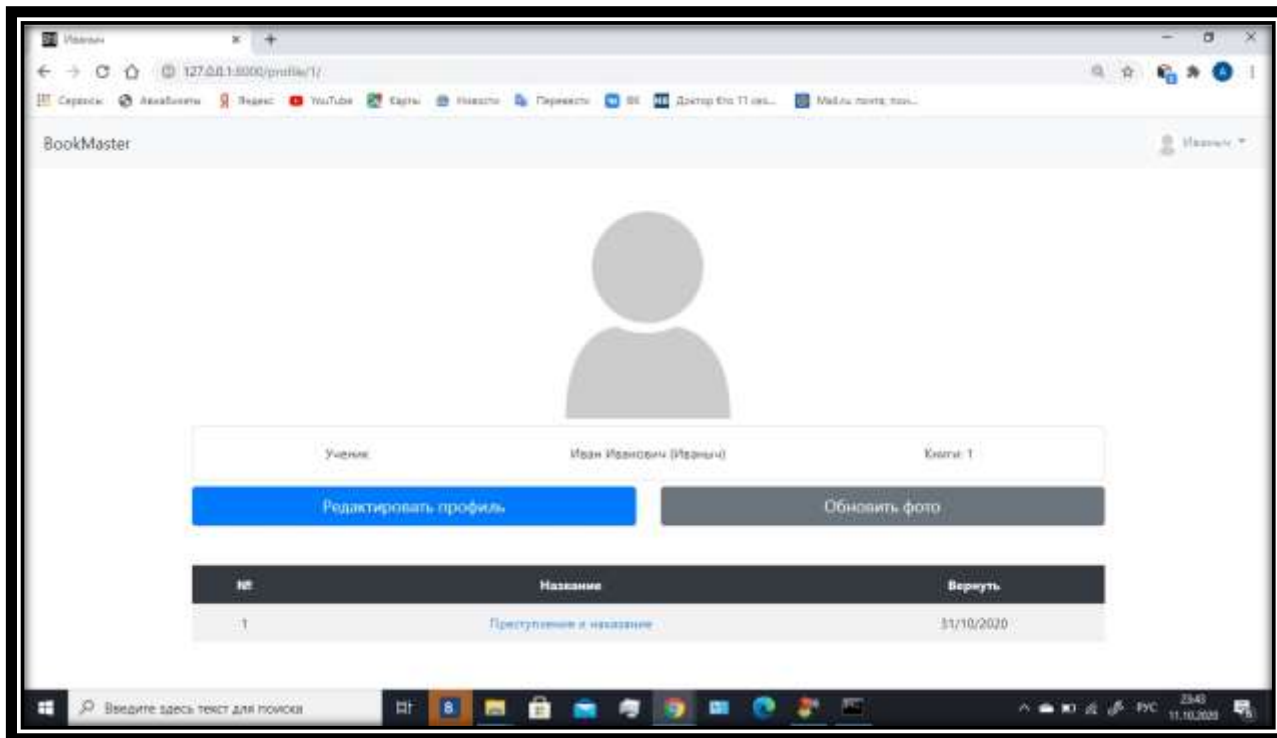


Рис22

Вернем книгу обратно. Для этого снова нажимаем на эту книгу. Видим информацию о текущем читателе и информацию, просрочил ли он. Нажимаем на зеленую кнопку «Вернуть». Рис23.

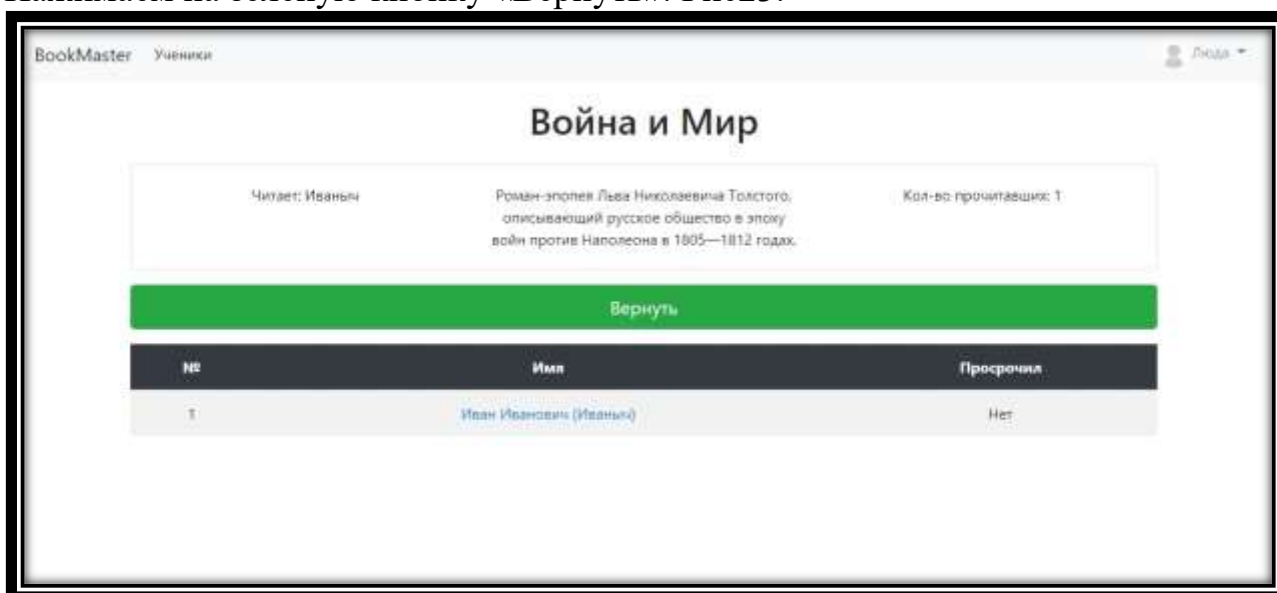


Рис23.

Как только возвращается книга, у пользователя в профиле она пропадает (Рис15), а на основной странице она становится доступной (Рис12).

Зайдя на профиль библиотекаря и нажав на выданные книги, мы увидим статистику: какую книгу брали, кто и на какой срок. Можно выбрать период статистики, чтобы смотреть любой промежуток времени. Рис24.

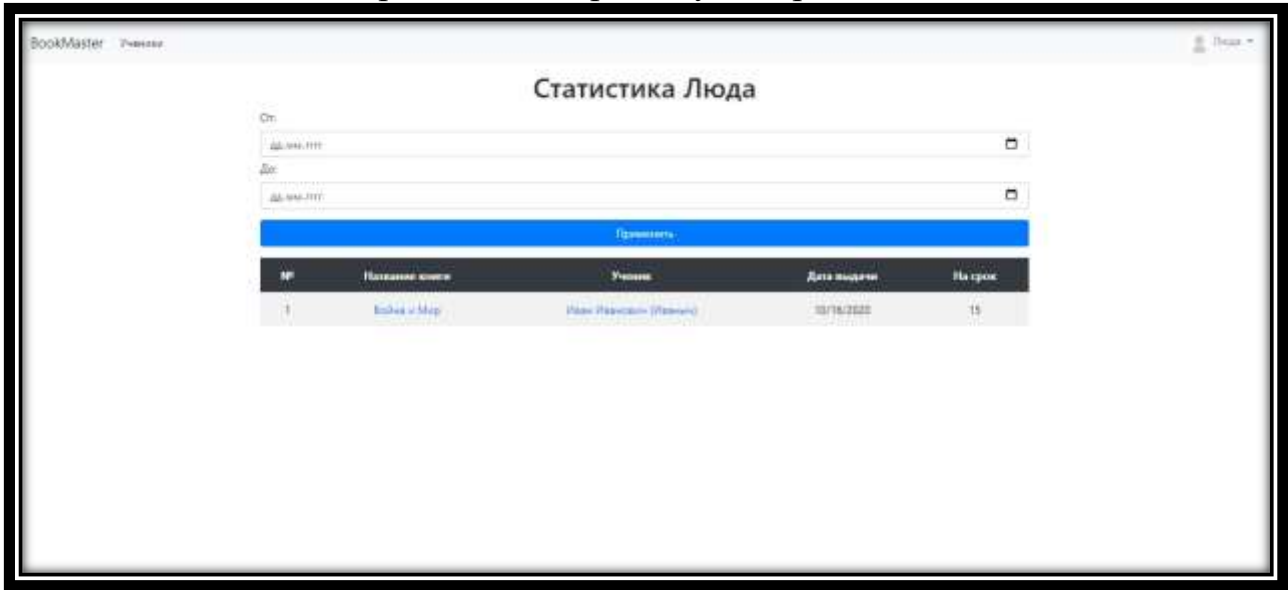


Рис24.

Чтобы попасть на страницу администратора, нужно ввести определенный адрес. Вводим логин и пароль и заходим на страницу администрации. Рис25.

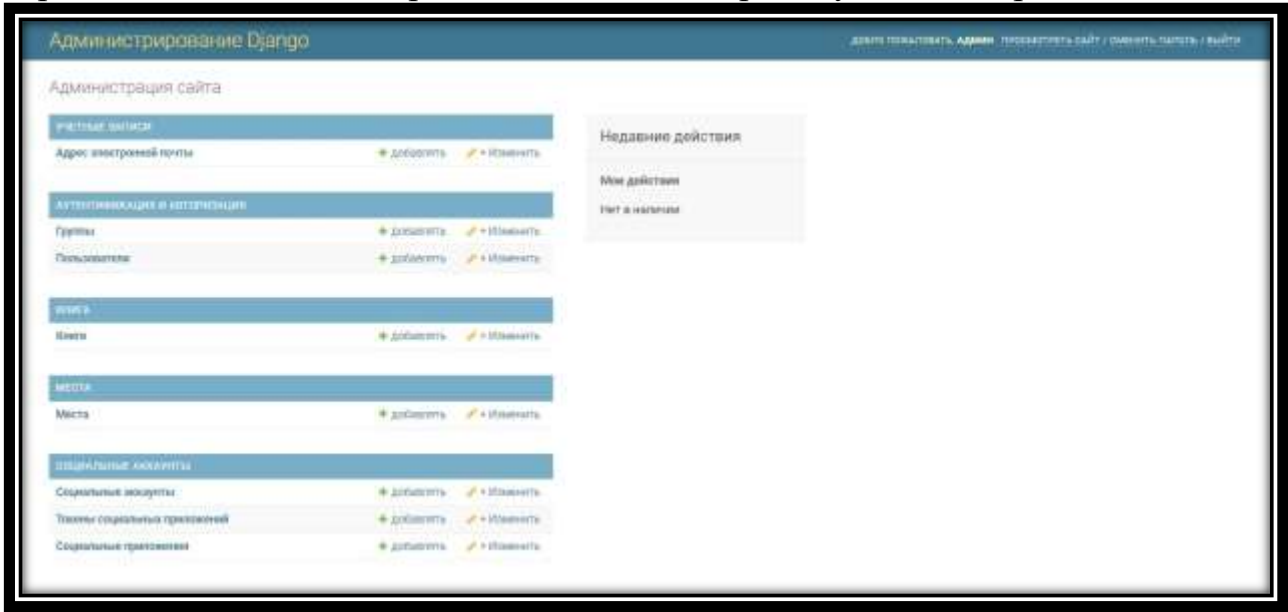


Рис 25.

Если выбрать «добавить» у строчки адреса электронной почты, то можно будет добавить электронную почту. Рис26.

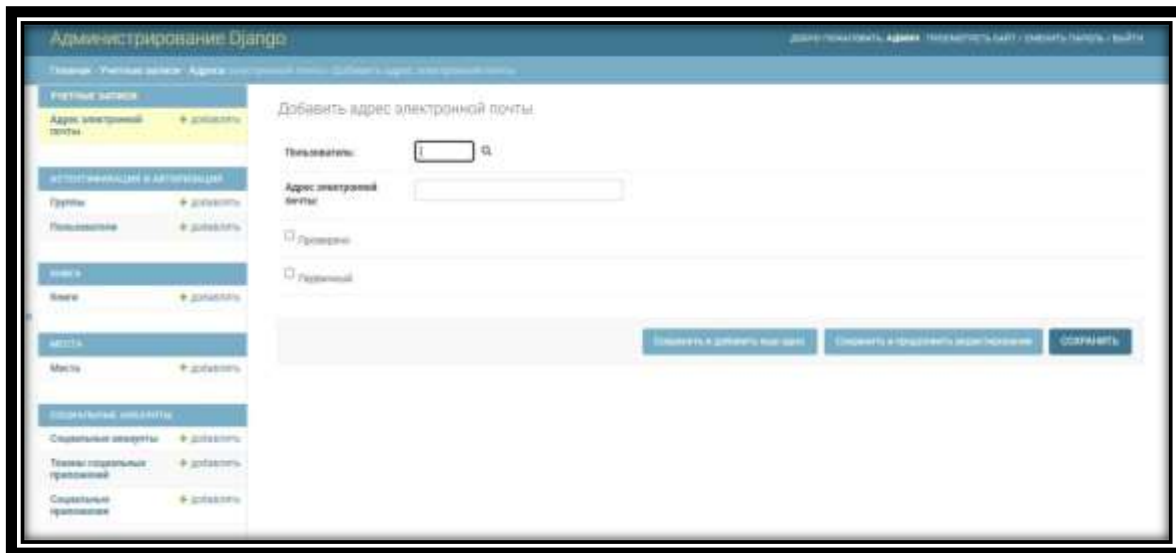


Рис26.

Если нажать на «Изменить», то мы сможем просмотреть и изменить некоторые данные пользователей. Рис27.

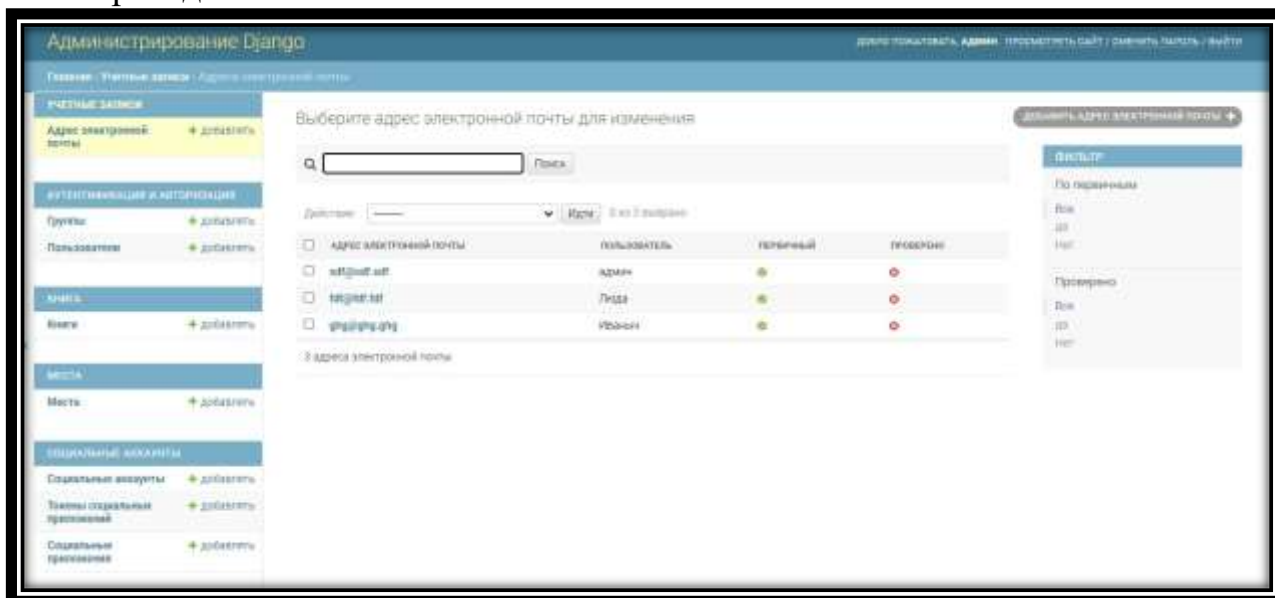


Рис27.

Нажав на «добавить» у пользователей, мы можем создать нового пользователя. Рис28.

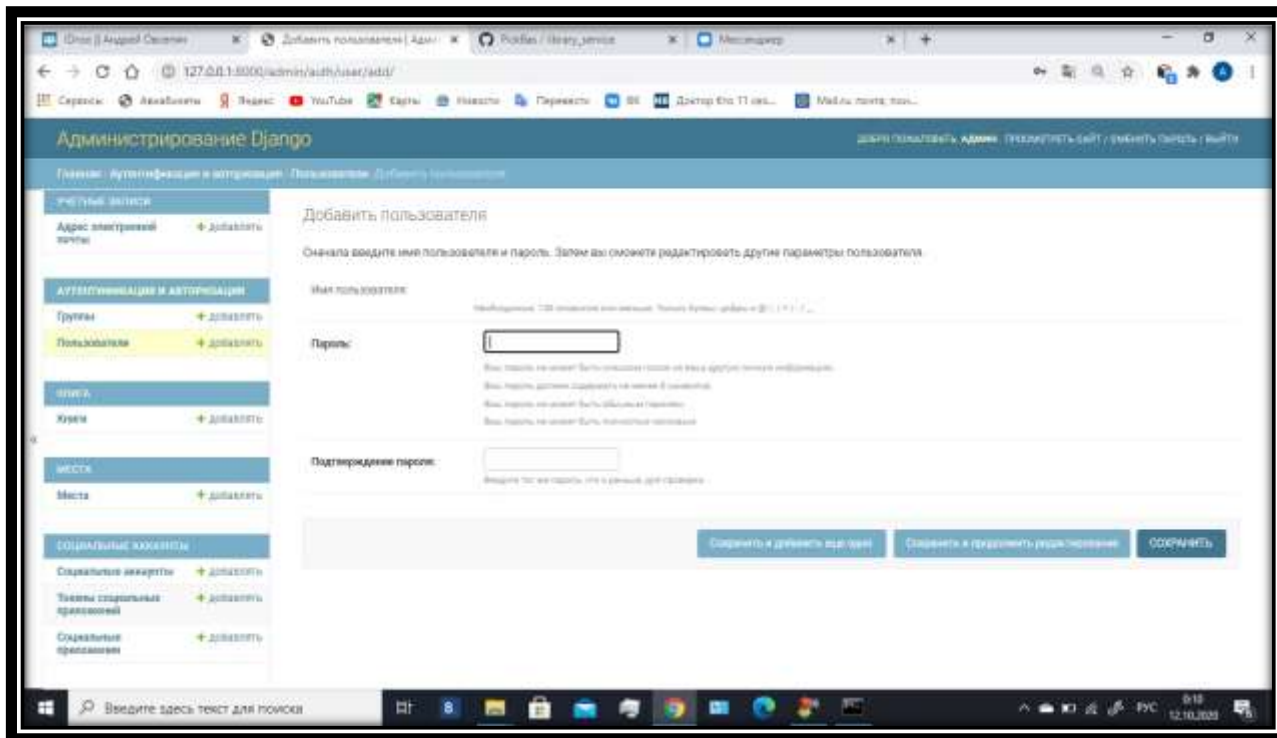


Рис28.

Если нажмем на «изменить», то сможем редактировать некоторые данные пользователей. Рис29.

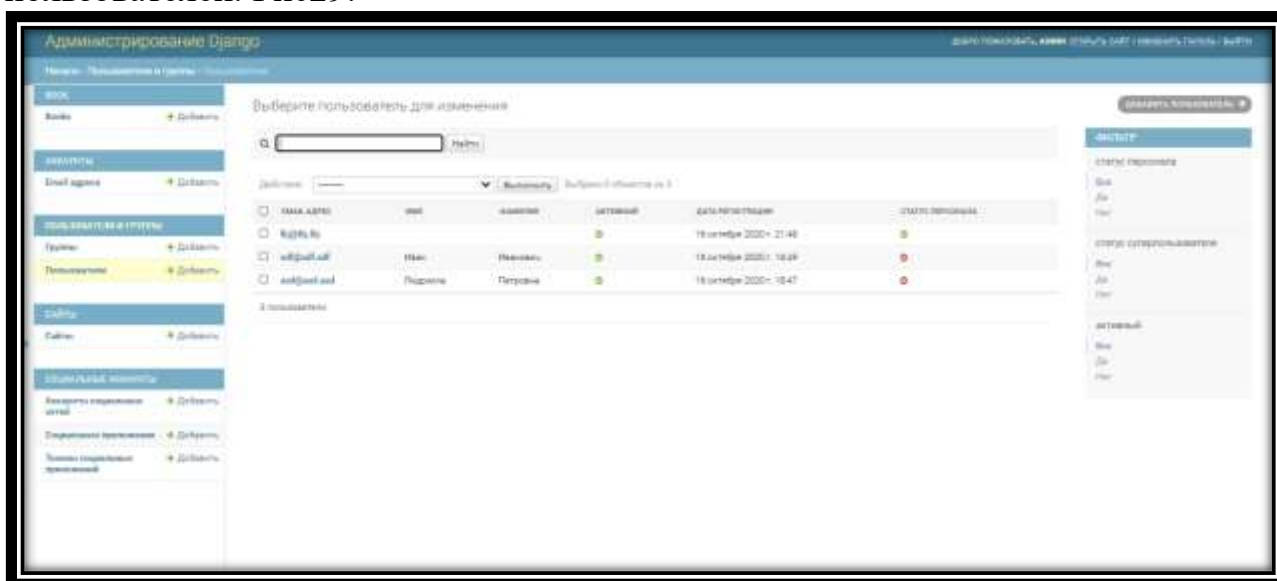


Рис29.

Администратор также может добавить книги и информацию для нее. Рис30.


```
$ python manage.py createsuperuser_profile
Enter username: admin

Enter password: admin_password
Please repeat the password: admin_password

Enter email: admin@email.com

Superuser created successfully!
```

Рис32.

Контрольный пример

№ п/п	Входные данные	Реакция системы ((выходные данные))	Описание проверяемой функциональности системы
1	Вход в учетную запись ученика. Поле "Имя пользователя": student1 Поле "Пароль": school202	Ошибка. Имя пользователя и/или пароль не верны.	Попытка войти в аккаунт ученика с неверным паролем
2	Вход в учетную запись ученика. Поле "Имя пользователя": student1 Поле "Пароль": school2020	Успешно. Ученик вошел в свою учетную запись	Вход в аккаунт ученика
3	Добавление книги в библиотеку. Поле "Название книги": Л.Н. Толстой "Война и Мир". Поле "Информация о книге":	Ошибка. Поле "Информация о книге" не заполнено	Попытка добавить книгу в библиотеку, когда поле "Информация о книге" не заполнено
4	Добавление книги в библиотеку. Поле "Название книги": Л.Н. Толстой "Война и Мир". Поле "Информация о книге": книга вышла в свет в 1863 году...	Успешно. Книга добавлена в библиотеку	Добавление книги в библиотеку
5	Выдача книги ученику. Поле "Ученик": Egor Sapozhkov (student 1) Поле "Дата возврата": 06.02.2020	Ошибка. Некорректная дата возврата книги	Попытка выписать книгу ученику с некорректной датой возврата
6	Выписывание книги ученику. Поле "Ученик": Egor Sapozhkov (student 1) Поле "Дата возврата": 06.02.2021	Успешно. Книга выписана ученику	Выписывание книги ученику

7	<p>Регистрация пользователя (ученика). Поле "E-mail": sapozhkov.egor@mail Поле "Имя пользователя": Egor2003 Поле "Имя": Egor Поле "Фамилия": Sapozhkov Поле "Пароль": egor2003 Поле "Пароль (ещё раз)": egor2003 Поле "Секретный ключ":</p>	<p>Ошибка. Введён неправильный адрес электронной почты Введённый пароль слишком похож на имя пользователя</p>	<p>Попытка регистрации с неверным адресом электронной почты и похожим на имя пользователя паролем</p>
8	<p><i>Регистрация пользователя (ученика).</i> Поле "E-mail": sapozhkov.egor@mail.ru Поле "Имя пользователя": Egor2003 Поле "Имя": Egor Поле "Фамилия": Sapozhkov Поле "Пароль": thebestschool26 Поле "Пароль (ещё раз)": thebestschool26 Поле "Секретный ключ":</p>	<p>Успешно. Пользователь зарегистрирован в системе</p>	<p>Добавление нового пользователя в систему</p>