**Министерство науки и высшего образования Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”  
(УНИВЕРСИТЕТ ИТМО)**

**ЦЕНТР АВТОРИЗОВАННОГО ОБУЧЕНИЯ ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ**

**ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА**

**Социальная сеть SocFilms в виде веб-приложения**

Автор **Ильюшин Никита Александрович** \_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия Имя Отчество) (Подпись)

**Центр авторизованного обучения информационным технологиям**

Наименование программы **«Python-разработчик»**

Руководитель **Кузьмин Константин Михайлович** \_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

**К защите допустить**

|  |  |  |
| --- | --- | --- |
| *Заместитель директора ЦАО ИТ,*  *к.т.н.* |  | / *Т.В. Зудилова*/ |

Санкт-Петербург, 2023г.

Обучающийся Ильюшин Никита Александрович Группа 124/29

(Фамилия, И. О.)

Работа принята «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Работа выполнена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023г.

Секретарь ИАК \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Фамилия, И. О.) (подпись)

Листов хранения \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Демонстрационных материалов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ОГЛАВЛЕНИЕ**

[Введение 4](#_Toc130771979)

[Цель 4](#_Toc130771980)

[Описание системы 4](#_Toc130771981)

[Интерфейс системы 6](#_Toc130771982)

[Функции системы 8](#_Toc130771983)

[Структура данных 9](#_Toc130771984)

[Разворачивание ПО 9](#_Toc130771985)

[Тестирование 10](#_Toc130771986)

[Заключение 10](#_Toc130771987)

[Литература 11](#_Toc130771988)

# Введение

Наименование программы – “SocFilms”.

Веб-приложение предназначено для упрощения процессов поиска, сбора и обмена информации пользователями.

Актуальность задачи обусловлена несколькими факторами:

* Поиск новых фильмов для просмотра по различным источникам занимает много времени;
* Опрос друзей об интересных фильмах занимает много времени;
* Отсутствие прямых конкурентов в области социальных сетей, позволяющих сделать акцент на обмене информации о кино.

Целевая аудитория ограничена базовыми навыками владения сетью Интернет и ПК.

# Цель

* Изучить принципы организации веб-приложения
* Изучить функциональные требования к разрабатываемому приложению.
* Разработать веб-приложение, отвечающее функциональным требованиям.

# Описание системы

Технические требования:

* Веб-приложение с пользовательским интерфейсом в виде упорядоченных html страниц.
* Многопользовательность

Технические средства:

* Python 3.11 язык программирования
* Flask 2.2.3 веб-фреймворк
* Pycharm Community Edition 2022.3.3 среда разработки
* MySQL Community 8.0.32.0 серверная БД

Данный стэк технологий был выбран для реализации данного приложения, потому что полностью удовлетворяет заданные технические требования.

В качестве альтернативы можно выбрать для веб-фреймворка - Django для упрощения и скорости создания приложения, а для БД – любую другую серверную, например PostgreSQL, если важно использование независимого ПО.

Для пользователей, которым нужно хранить и обмениваться информацией о кинофильмах, данная программа “SocFilms” является информационной системой, которая обеспечит легкость поиска и обмена данными между пользователями. Система будет сохранять и воспроизводить данные о кинофильмах, а также отношениях между пользователями, равно как и предоставлять доступ к соответствующему функционалу.

Пользовательские истории:

* Как анонимный пользователь, я хочу иметь возможность выполнить поиск фильма, чтобы понять, насколько большая база фильмов и в каком виде выводится результат.
* Как анонимный пользователь, я хочу иметь возможность выполнить регистрацию для использования полного функционала приложения.
* Как зарегистрированный пользователь, я хочу иметь возможность поиска друзей и отправки запроса дружбы, чтобы просматривать избранное друга.
* Как зарегистрированный пользователь, я хочу иметь возможность принять или отклонить запрос дружбы, чтобы только настоящие друзья могли просматривать мое избранное.

Функциональные требования:

* Возможность регистрации и входа по никнейму
* Возможность выполнить поиск фильма анонимному пользователю
* Возможность выполнить поиск и добавить в избранное фильм зарегистрированному пользователю
* Возможность выполнить поиск пользователя по стране и отправить запрос на дружбу зарегистрированному пользователю
* Если зарегистрированный пользователь получил запрос на дружбу, имеется возможность принять или отклонить такой запрос.
* Если у зарегистрированного пользователя есть друг, есть возможность просмотреть избранное друга.

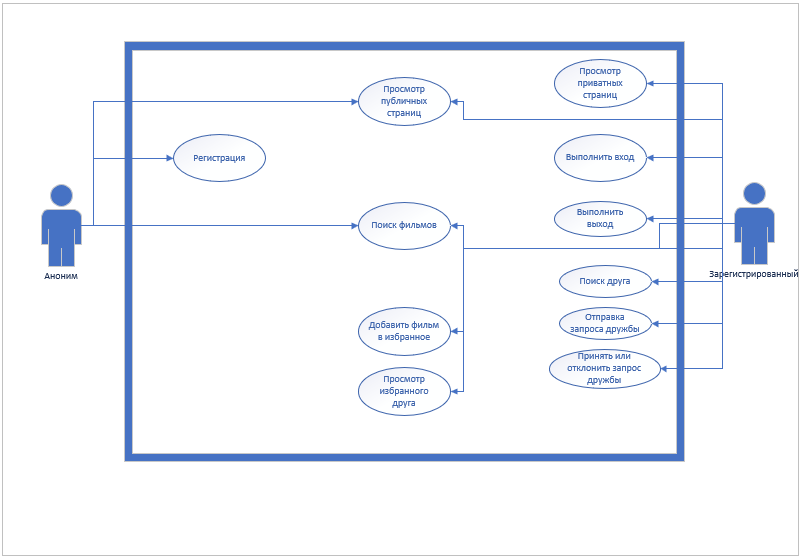


Рис.1 Варианты использования

# Интерфейс системы

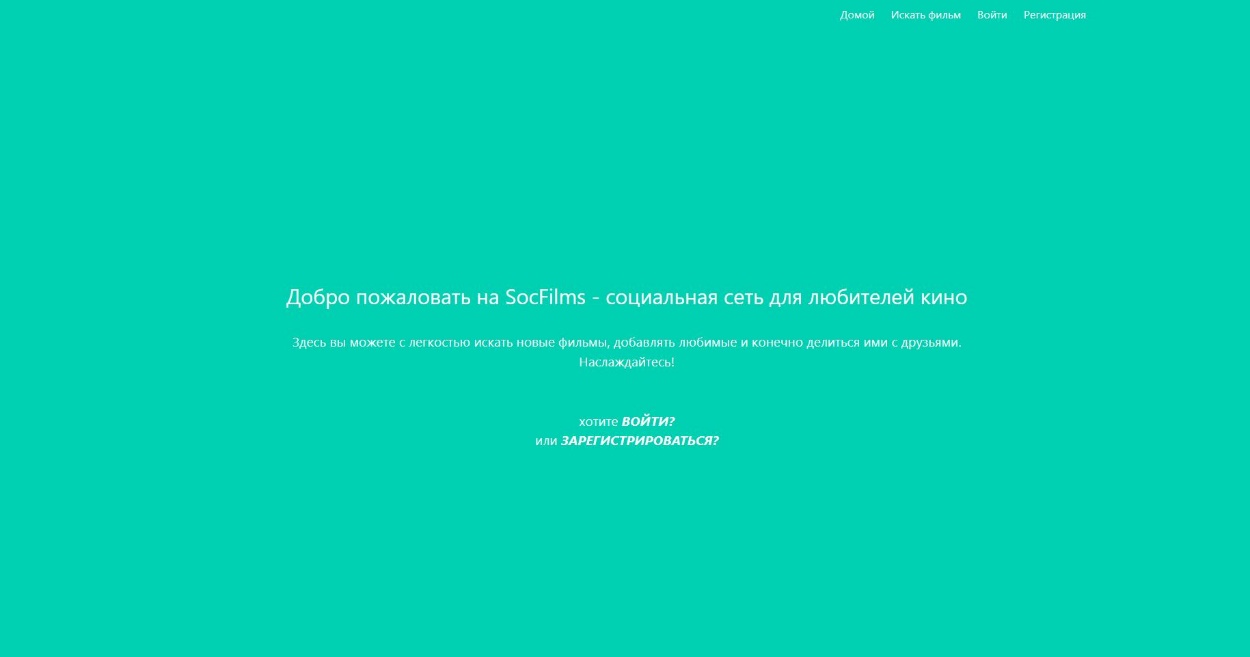


Рис.2 Домашняя страница

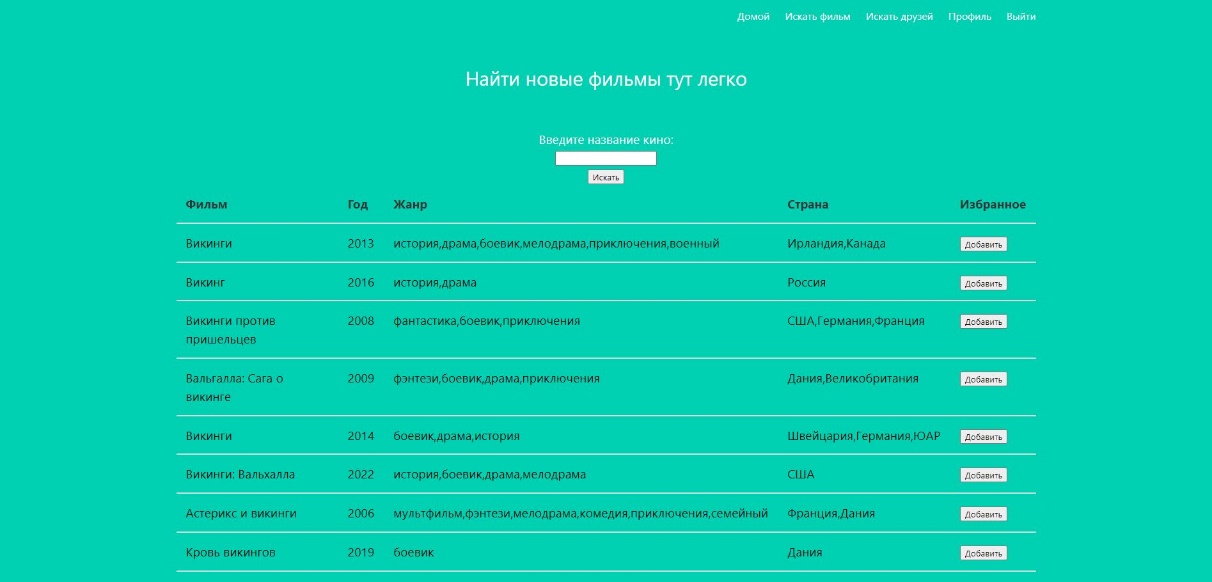


Рис.3 Поиск фильмов зарегистрированным пользователем (анонимный не может добавлять)

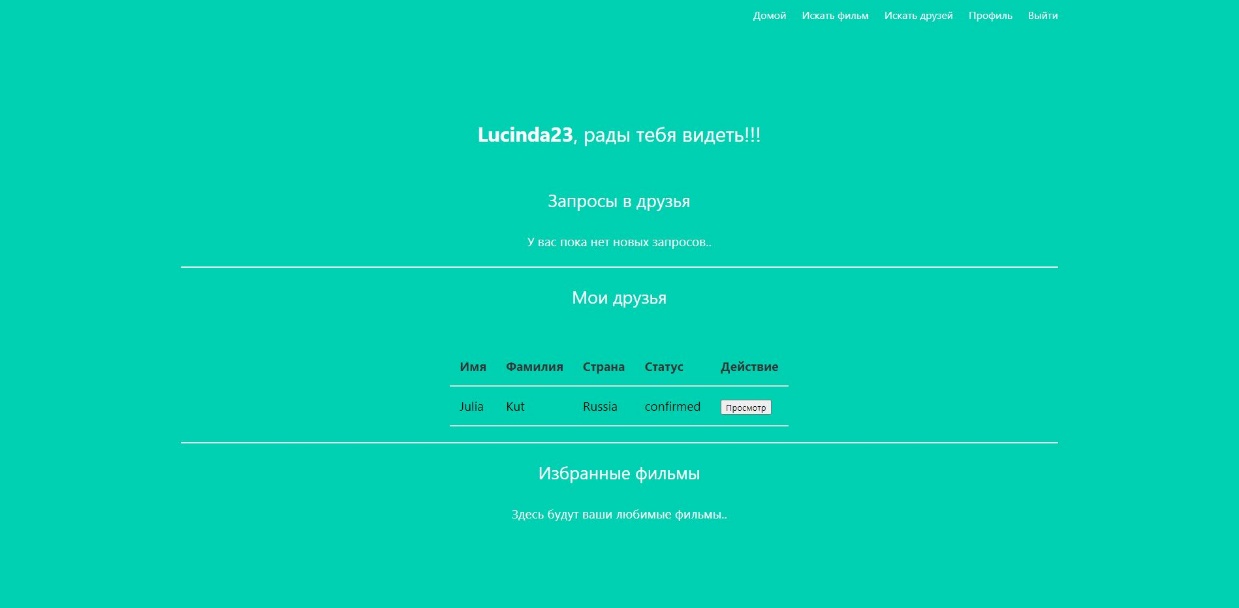


Рис.4 Профиль зарегистрированного пользователя

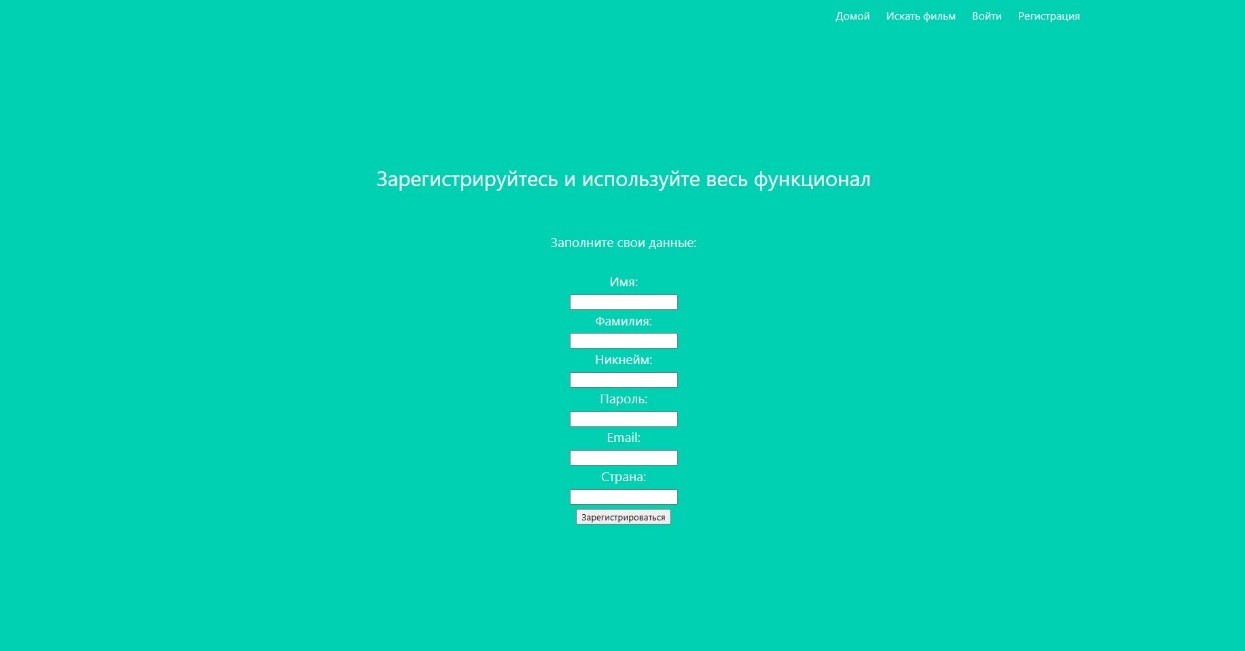


Рис.5 Регистрация пользователя

Структура системы:

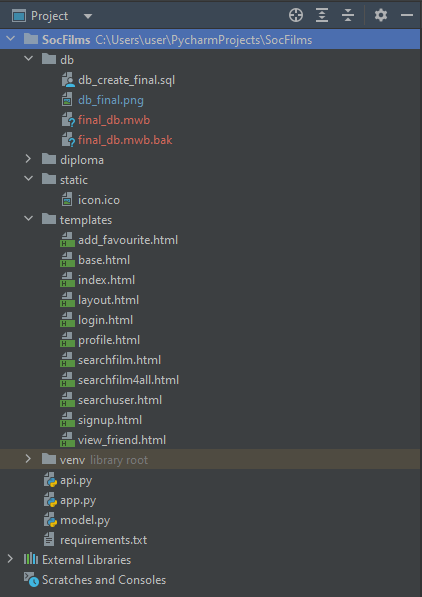


Рис.6. Структура системы

Система состоит из нескольких разработанных модулей, файлов шаблонов html,

статического содержимого, данных для развертки БД, требований к импортированным внешним библиотекам. В качестве источника данных о кино используется стороннее API. При выполнении поиска фильма, система обращается к API, затем проверяет есть ли в своей БД такие записи и, если есть разница, обновляет свои данные, затем выдает результат пользователю.

При добавлении фильма предусмотрена проверка информации в своей БД о жанрах и странах для исключения записи повторяющихся значений.

# Функции системы

1. Регистрация
2. Вход
3. Выход
4. Поиск фильма
   1. Обращение к API
   2. Обращение к БД
   3. Идентификация новых данных и внесение их в БД
   4. Вывод информации пользователю
   5. Добавление в избранное
5. Поиск друга

5.1 Отправка запроса на дружбу

6. Принятие или отклонение поступившего запроса дружбы

7. Просмотр избранного подтвержденного друга

# Структура данных

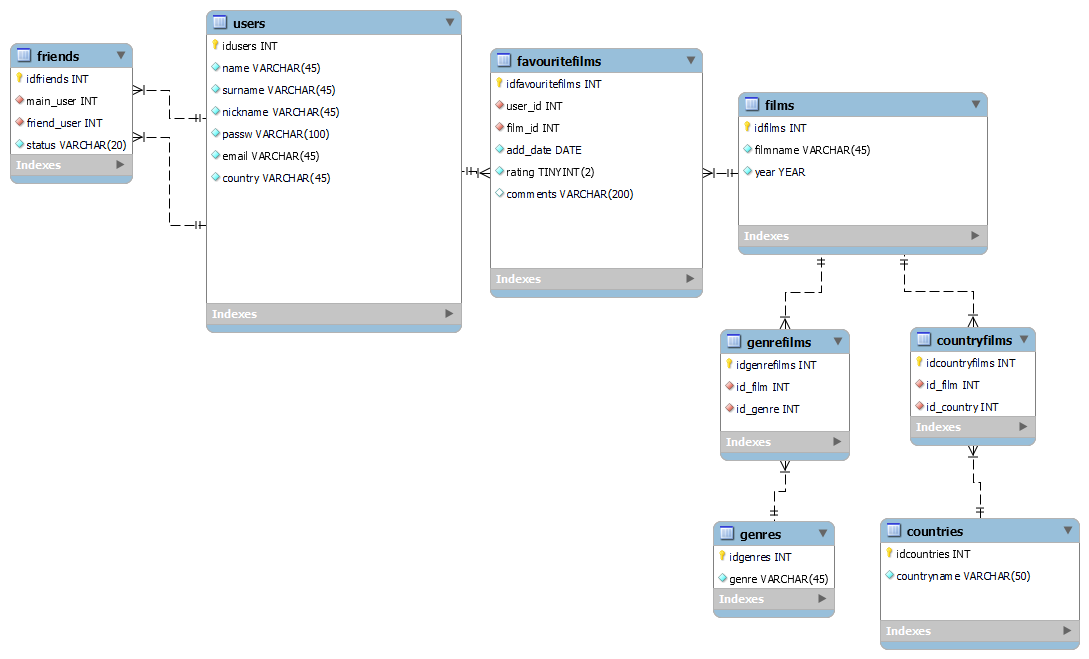


Рис.7 Схема модели базы данных

# Разворачивание ПО

Для функционирования веб-приложения требуется веб-сервер, совместимый с Python/MySQL.

В соответствии с инструкции deployment.txt в целях предварительного конфигурирования и развертки рабочей БД системному администратору необходимо назначить пользователя с правами DBA в СУБД MySQL.

Логин и пароль вписать в конфигурационные переменные словарей config\_deploy и config, проверить доступность порта, затем модуль model.py как основной. В случае успеха на консоль выводится сообщение об успешной отработке скрипта по созданию структуры БД.

В дальнейшем используется app.py со всеми требуемыми зависимостями, указанными в requirements.txt.

1) Create or choose account with DBA permissions on your MySQL server  
2) Insert DB account`s name and password in config\_deploy and config dicts in model.py in order  
to operate correctly.  
3) To deploy DB you need to run model.py as main  
4) Install all required packages via pip install -r requirements.txt  
5) run app.py as web application instance on a server

Рис.8 Инструкция по разворачиванию(деплойменту)

# Тестирование

В процессе разработки необходимо проверять корректность входных и выходных данных функций и модуля в целом, чтобы получать ожидаемый результат работы программы. Также необходимо выполнить ручное тестирование веб-интерфейса с помощью различных тест-кейсов, чтобы устранить возможное нежелательное поведение программы. Подобным мы всесторонне имитируем поведение пользователя, передающего разнообразные данные через веб-интерфейс.

В процессе разработки и тестирования были найдены и решены следующие проблемы:

* при развертке БД пользователь должен иметь права на все операции, указанные в SQL скрипте
* невозможность отправить запрос дружбы самому себе
* при подтверждении запроса дружбы только один пользователь видел запись о дружбе
* невозможно создать пользователя с уникальным никнеймом

# Заключение

Данная работа позволила на практике закрепить и расширить полученные знания в области проектирования и разработки веб-приложений. В соответствии с задокументированными техническими и функциональными требованиями задачи были решены.

В части сложностей, было необходимо решить несколько возникших проблем с отображением подтверждённых друзей друг у друга, а также с исключением многократных повторов записей в таблицах жанры и страны. Данные сложности были устранены и отлажены программно на уровне СУБД и использования алгоритма проверки.

Данное приложение можно усовершенствовать:

* возможностью самостоятельного добавления фильмов напрямую в БД
* возможностью наполнения личных сведений в профиле
* возможностью просмотра трейлеров путем добавления плеера
* возможность поиска фильмов по различным атрибутам
* возможность ручного добавления фильмов
* возможность поиска пользователей по различным атрибутам

# Литература

1. Swaroop C H A Byte of Python, Sphinx, 2020. - 164 с.
2. Лутц Марк Изучаем Python, том 1, 5-е изд, СПб.: ООО “Диалектика”, 2019. — 832 с.
3. Лутц Марк Изучаем Python, том 2, 5-е изд, СПб. : ООО “Диалектика”, 2020.— 720 с.
4. Вигерс Карл, Битти Джой Разработка требований к программному обеспечению. 3-е изд., дополненное, БХВ-Петербург, 2014. - 736 стр.
5. Flask’s documentation: сайт. – URL: https://flask.palletsprojects.com/en/2.2.x/(дата обращения 24.03.2023). – Текст: электронный.
6. MySQL Connector/Python Developer Guide: сайт. – URL:
7. <https://dev.mysql.com/doc/connector-python/en/>(дата обращения 20.03.2023). – Текст: электронный.
8. [Abdelhadi Dyouri](https://www.digitalocean.com/community/users/adyouri), How To Make a Web Application Using Flask in Python 3, 2020: сайт. – URL: <https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3>(дата обращения 24.03.2023). – Текст: электронный.