МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(МОСКОВСКИЙ ПОЛИТЕХ)**

Факультет информационных технологий

Кафедра «Инфокогнитивные технологии»

КУРСОВОЙ ПРОЕКТ

на тему: «[Инструмент исследования данных из файлов с книгами»](https://docs.google.com/document/u/0/d/1MsRGsYIjr5tt6F78gV0o7QPw0jTChqkLM2v3KcLbJAk/edit)

Направление подготовки 09.03.03 «Прикладная информатика»

Профиль «Корпоративные информационные системы»

**Выполнил:**

студент группы 211-361

Сыров Максим Евгеньевич

|  |  |
| --- | --- |
| 27.06.2023 |  |
|  | (подпись) |

Москва 2023

**Описание**

При установке всего проекта, пользователю необходимо добавить свою папку с книгами в корневую папку проекта.

Проверить установлены ли следующие модуля для python:

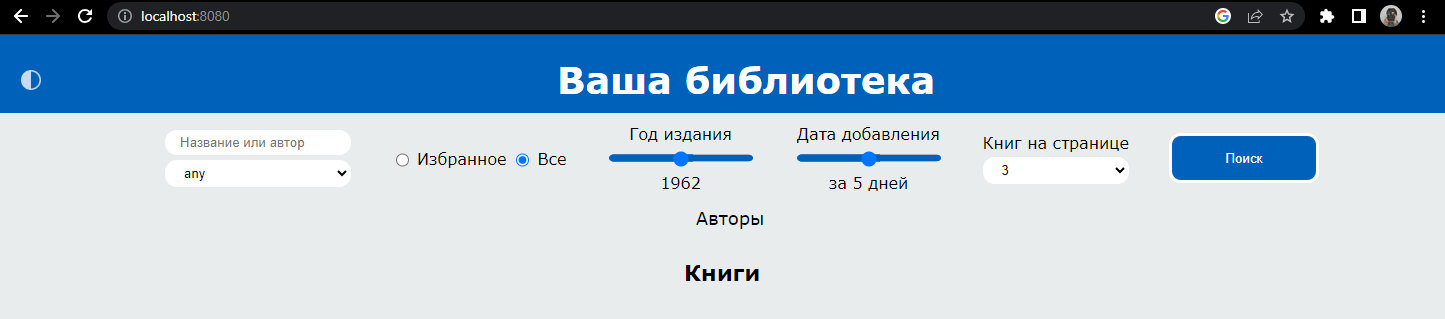
* Flask — фреймворк для работы с html и GET запросами;
* base2048/base64 — конвертация изображения в BLOB формат хранения картинок;
* pysqlite3 — подключение к базе данных;
* datetime2 — конвертация строки в формат даты;
* math22 — математические вычисления
* waitress — производственная версия запуска приложения в flask фреймворке;
* collections2 — вспомогательный модуль для counter-a;
* colorama — окрашивание текста в консоли;
* threading2 — последовательный запуск частей кода, прерывание потоков;
* argparse3 — парсер входных данных, с помощью которого возможен запуск проекта через консоль
* chardet2 — автоматическое определение кодировки книги;
* micropython-hashlib5 — хэш данных книги;
* Pillow — сохранение изображений и их открытие;
* PyMuPDF (fitz) — вспомогательный модуль работы с pdf файлами;
* EbookLib — вспомогательный модуль работы с epub файлами;
* Pypdf — вспомогательный модуль работы с pdf файлами;
* bs4 — BeautifulSoup, вспомогательный модуль работы с fb2 и прочими файлами;
* ebookmeta — модуль, позволяющий достать метаданные книг в формате epub;

Алгоритм работы программы:

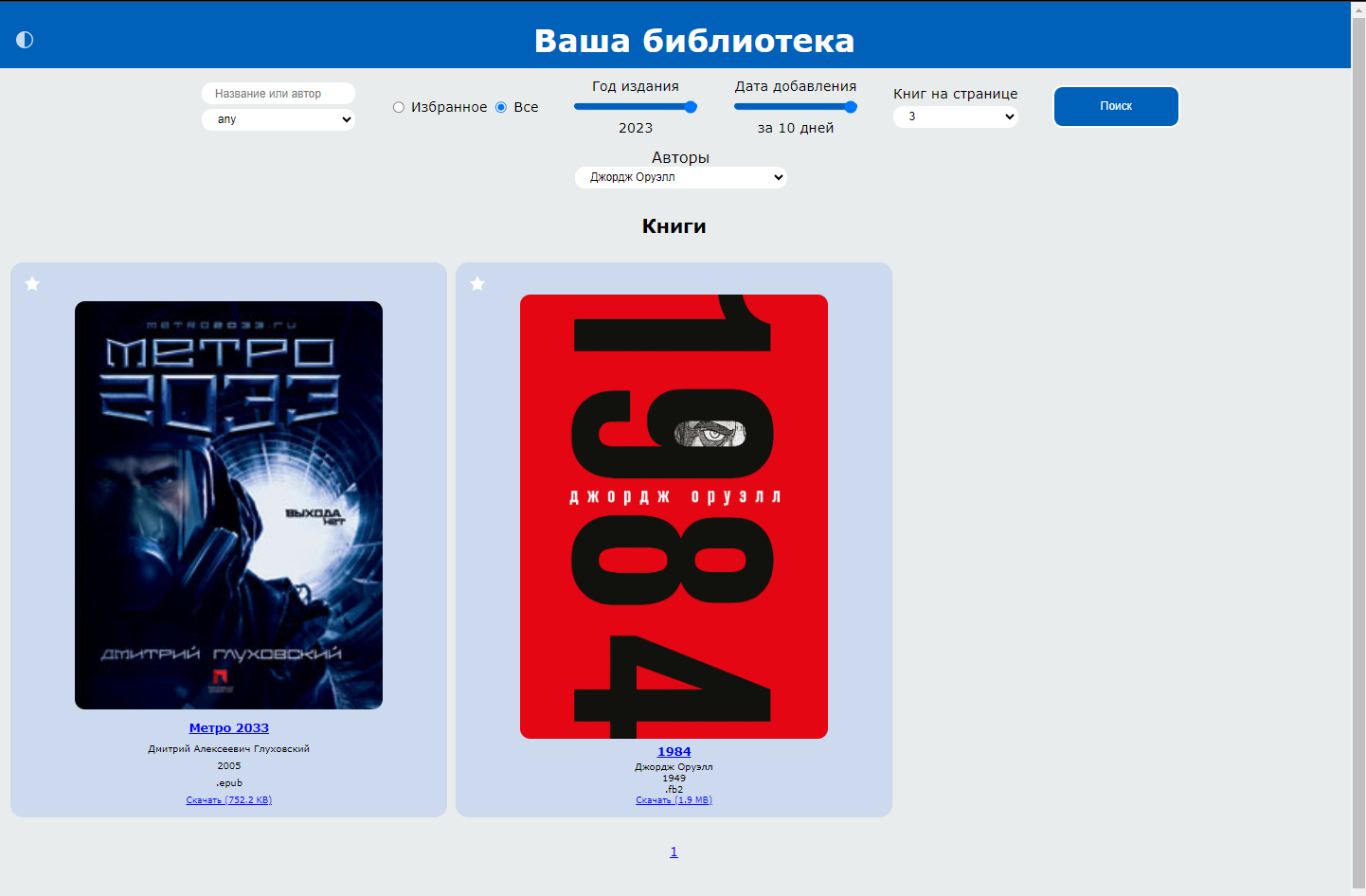
Сначала работает первый алгоритм, который находит в корневой папке проекта файлы с расширением книг, определяя корневую папку, где хранятся все книги.

После чего, функция check\_repeated\_books. Рекурсивно проходит по всей папке с книгами. Из каждой книги получает её название и сохраняет в массив элемент, в котором хранится название книги и путь к книге. После проходимся снова по каждой книге. Находим сколько раз наша книга повторяется в всех папках, выводим информацию о ней.

Функция process\_folder. Цикл идет по каждой книге в каталоге, определяет расширение книги, от этого вызывается необходимый класс из другого модуля version1\_0.py. Там есть 4 класса, каждый из которых представляет из себя каждое расширение книги. Из каждой книги вытаскивается автор, название, год публикации, текст, путь к книге, количество страниц, обложка (при наличии), вычисляется хэш книги. После чего каждая книга добавляется в базу данных или, если книга уже была добавлена в базу данных (сверяется хэш книги), другие данные обновляются.

Функция start\_web\_server запускает web-server по адресу localhost:8008/. Она запускает файл index.html с формой фильтрации данных среди книги

Существуют поля с поиском книги по названию или автору, выбрать формат книги, определить максимальный год издания книги, максимальную дату добавления книги в базу данных, а также количество книг на странице.

****В левой части также можно включить темную тему.

****

Помимо всего прочего, существует функционал добавления книги в список избранных (белая звездочка)

Все параметры формы передаются через GET запрос на страницу /submit. В модуле forms.py с помощью фреймворка flusk определяется путь /submit в котором получаются параметры из url запроса.

После чего из базы данных парсится массив с данными о книгах в виде двумерного массива с примененными фильтрами, после чего происходит новый рендер страницы. Jinja2 позволяет получать доступ к переменным, переданные из python файла в самом html файле через синтаксис {%%}. Так, выводятся книги через цикл в общем формате, а каждая книга представляет собой массив многих данных.

Также, каждое название книги представляет собой гиперссылкой на фрагмент книги. Из базы данных применяется выборка по контенту, после чего массив обрезается в 15000 символов.

Также, каждую книгу можно скачать через путь /download, который использует специальный метод у фреймворка flusk send\_file. Нужно лишь передать путь к файлу, а его мы взяли из базы данных.

Сервис поддерживает некотрое количество страниц отображения книг. Так, передается в url строку номер страницы, после чего общий массив данных обрезается в зависимости от количества страниц и страницы на которой находится пользователь.

Для работы темной темы, отображения чисел под ползунками и авто заполнением полей после отправки формы, используются скрипты на языке JavaScript.

**Заключение**

1. Ссылка на репозиторий:

<https://github.com/moximilian/Course-project-4sem/tree/master>

**Инструкция к применению**

https://github.com/moximilian/Course-project-4sem/tree/master#readme