МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное учреждение высшего образования
«Чувашский государственный университет И.Н. Ульянова»
Факультет информатики и вычислительной техники
Кафедра вычислительной техники

Кросс-платформенные средства разработки программного обеспечения
Лабораторная работа 6
«Использование веб-фреймворка FLASK»

Выполнил:

Студент группы ИВТ-41-20
Галкин Д.С.

Проверил:

Ковалев С.В.

Чебоксары, 2023

Цель работы:

1. Ознакомление с понятием RESTful API
2. Ознакомление с вэб-фреймворком FLASK
3. Получение практических навыков в создании веб-сервисом и развертывании веб-приложений в облачных платформах

Задание для выполнения

В шестой лабораторной работе необходимо реализовать web-приложение, которое позволяет найти пример употребления какого-либо слова на иностранном языке в реальной художественной литературе. Демонстрационный пример какое-то время будет доступен по следующей ссылке
HTTPS://KVTIVTEXAMPLE.PYTHONANYWHERE.COM/

К программе предъявляются следующие требования:

1. Web-приложение состоит, как минимум, из четырех базовых конечных точек (enpoints), т.е существует четыре url, на которые можно отправлять запросы
2. Первая конечная точка («/») является стартовой, предназначена для удобства и просто перенаправляет запрос на url «/search», используя функцию redirect.
3. Вторая конечная точка («/search») содержит форму, позволяющую указать слово для поиска и количество примеров, которые необходимо получить. После заполнения параметров поиска и нажатия на кнопку, происходит переход к третьей конечной точке «/result»
4. Третья конечная точка («/result») обрабатывает полученный запрос, то есть ищет необходимо слово в файле достаточно большого размера и формирует результат поиска в виде отдельной html-страницы
5. Четвертая конечная точка («/request») принимает запрос в виде json-данных, ищет указанные примеры и формирует ответ, который также является json-данными. Для проверки работоспособности данной конечной точки требуется использоваться приложение Postman. Json- запросы и json-ответы должен содержать, как минимум, поля, показанные на рисунке, приведенном на следующей странице
6. При работе с формами поиска Web-приложение должно сохранять cookie и при повторном посещении страницы в форме ввода параметров должны содержаться ранее указанные данные на случай, если пользователь хочет вспомнить пример употребления ранее выбранного слова
7. Web-приложение должно быть размещено на сайте HTTPS://WWW.PYTHONANYWHERE.COM

⚠ **Внимание!**

При работе с файлами следует учесть, что кодировка по умолчанию на вашей рабочей машине, и на сервере pythonanywhere может различаться, поэтому рекомендуется заранее узнать кодировку файла, который вы собираетесь использовать в качестве примера и указывать её явно при открытии файла

Полный текст программы:

forms.py

```
from typing import List


class Document():
    def __init__(self, book: str, words: List[str], min_sentence: int, max_sentence:
int, count: int):
        self.book = book
        self.words = words
        self.min_sentence = min_sentence
        self.max_sentence = max_sentence
        self.count = count
```

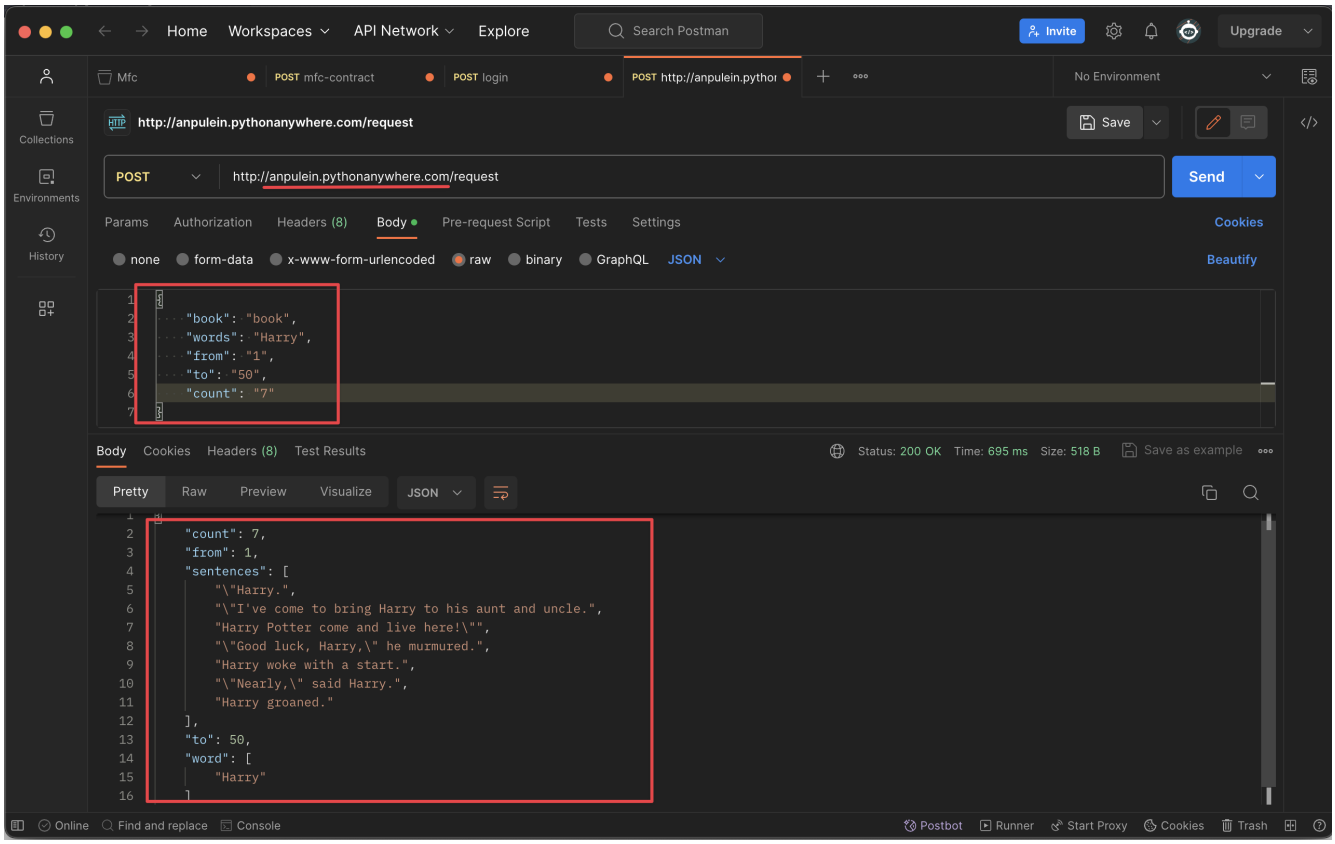
mixin.py

```
from forms import *
from typing import List, Dict
from nltk import sent_tokenize


def get_document(data) -> Document:
    return Document(
        book=data['book'],
        words=data['words'].split(',') if ',' in data['words'] else [data['words']],
        min_sentence=int(data['from']) if 'from' in data else 1,
        max_sentence=int(data['to']) if 'to' in data else 50,
        count=int(data['count'])
    )


def is_sentence(sentence: str, document: Document) -> bool:
    if (document.min_sentence < len(sentence) < document.max_sentence) and all(
            word.lower() in sentence.lower() for word in document.words):
        return True
    return False

def search_in_doc(document: Document) -> Dict[str, List[str]]:
    sentences = []

    with open(document.book + '.txt') as file:
        text = sent_tokenize(file.read())
        for sentence in text:
            if (is_sentence(sentence, document)) and len(sentences) < document.count:
                sentences.append(sentence)

        return {'sentences': sentences }
```

main.py

```
from forms import *
from mixin import *
from flask import Flask, render_template, request, redirect, url_for, jsonify

app = Flask(__name__)


@app.route('/')
def index():
    return redirect(url_for('search'))


@app.route('/search', methods=['GET', 'POST'])
def search():
    return render_template('search.html')


@app.route('/result', methods=["GET", "POST"])
def result():
    # Обработка полей формы
    document = get_document(request.form)

    # Выполнение поиска
    result = search_in_doc(document)
    return render_template('result.html', result=result, count=document.count)


@app.route('/request', methods=['POST'])
def json_request():
    # Обработка JSON-запроса и поиск примеров
    document = get_document(request.json)

    # Выполнение поиска
    result = search_in_doc(document)
    if result:
        response_data = {
            'sentences': result['sentences'],
            'word': document.words,
            'count': document.count,
            'from': document.min_sentence,
            'to': document.max_sentence
        }
        return jsonify(response_data)


if __name__ == '__main__':
    app.run()
```

Тест:



Задание доступно по ссылке

HTTP://ANPULEIN.PYTHONANYWHERE.COM