Другий етап курсової роботи

виконала Геворгян Софія

План:

1. Визначення функціональних та нефункціональних вимог до програми
2. Опис можливостей пакетів json, xml та html
3. Опис даних, які буде отримувати програма

Визначення функціональних та нефункціональних вимог до програми

|  |  |
| --- | --- |
| Functional requirements |  |
| Process-oriented | 1. Система повинна дозволяти користувачу вибирати, які дані він хоче отримати – базуючись на певній сторінці чи на проектах Wikipedia Foundation 2. Система має дозволяти користувачу вказувати посилання на Вікіпедію або назву сторінки Вікіпедії 3. Система повинна дозволяти користувачу вказувати кількість сторінок із Вікіпедії, короткі дані яких він хоче переглянути 4. Система повинна перевіряти коректність посилання чи вказаної назви 5. Система повинна надавати можливість вибирати портал з якого будуть показувати сторінки 6. Система повинна надавати дані, пов’язані з обраною сторінкою базуючись на даних про кількість переглядів із Вікіпедії |
| Information-oriented | 1. Система повинна містити дані з різних сторінок розділу Вікіпедії за вказаною сторінкою/назвою 2. Система повинна надавати зображення і посилання до кожного короткого опису |

|  |  |
| --- | --- |
| Nonfunctional requirements |  |
| Operational | 1. Система повинна працювати в будь-якому веб-браузері |
| Performance | 1. Взаємодія між програмою і користувачем не повинна перевищувати 10 секунд 2. Система повинна бути доступною з будь-якого пристрою з підключенням до Інтернету |
| Security | 1. Користувачі можуть бачити лише результати пошуку і задавати свої параметри для роботи 2. Обмежений доступ до змін у функціонуванні системи |
| Cultural and Political | 1. Система доступна для користування лише однією мовою (англійською) |

Опис можливостей пакету json

Json is a lightweight data interchange format.

*json.dump(obj, fp, skipkeys=False, ensure\_ascii=True, check\_circular=True, allow\_nan=True, cls=None, indent=None, separators=None, default=None, sort\_keys=False, \*\*kw)*

Serialize *obj* as a JSON formatted stream to *fp*. The [json](https://docs.python.org/3.5/library/json.html#module-json) module always produces [str](https://docs.python.org/3.5/library/stdtypes.html#str) objects

If *skipkeys* is true (default: False), then dict keys that are not of a basic type ([str](https://docs.python.org/3.5/library/stdtypes.html#str), [int](https://docs.python.org/3.5/library/functions.html#int), [float](https://docs.python.org/3.5/library/functions.html#float), [bool](https://docs.python.org/3.5/library/functions.html#bool), None) will be skipped instead of raising a [TypeError](https://docs.python.org/3.5/library/exceptions.html#TypeError).

*json.dumps(obj, skipkeys=False, ensure\_ascii=True, check\_circular=True, allow\_nan=True, cls=None, indent=None, separators=None, default=None, sort\_keys=False, \*\*kw)*

Serialize obj to a JSON formatted [str](https://docs.python.org/3.5/library/stdtypes.html#str) using this [conversion table](https://docs.python.org/3.5/library/json.html#py-to-json-table). The arguments have the same meaning as in [dump()](https://docs.python.org/3.5/library/json.html#json.dump).

*json.load(fp, cls=None, object\_hook=None, parse\_float=None, parse\_int=None, parse\_constant=None, object\_pairs\_hook=None, \*\*kw)*

Deserialize fp (a .read()-supporting [file-like object](https://docs.python.org/3.5/glossary.html#term-file-like-object) containing a JSON document) to a Python object using this [conversion table](https://docs.python.org/3.5/library/json.html#json-to-py-table).

*json.loads(*s*,*encoding=None*,*cls=None*,*object\_hook=None*,*parse\_float=None*,*parse\_int=None*,*parse\_constant=None*,*object\_pairs\_hook=None*,*\*\*kw*)*

Deserialize s (a [str](https://docs.python.org/3.5/library/stdtypes.html#str) instance containing a JSON document) to a Python object using this [conversion table](https://docs.python.org/3.5/library/json.html#json-to-py-table).

The other arguments have the same meaning as in [load()](https://docs.python.org/3.5/library/json.html#json.load), except encoding which is ignored and deprecated.

If the data being deserialized is not a valid JSON document, a [JSONDecodeError](https://docs.python.org/3.5/library/json.html#json.JSONDecodeError) will be raised.

*class json.JSONDecoder(object\_hook=None, parse\_float=None, parse\_int=None, parse\_constant=None, strict=True, object\_pairs\_hook=None)*

Simple JSON decoder.

Performs the following translations in decoding by default:

|  |  |
| --- | --- |
| json | Python |
| object | dict |
| array | list |
| string | str |
| number (int) | int |
| number (real) | float |
| True | True |
| False | False |
| null | None |

*decode(s)*

Return the Python representation of s (a [str](https://docs.python.org/3.5/library/stdtypes.html#str)instance containing a JSON document).

[JSONDecodeError](https://docs.python.org/3.5/library/json.html#json.JSONDecodeError) will be raised if the given JSON document is not valid.

*raw\_decode(s)*

Decode a JSON document from s (a [str](https://docs.python.org/3.5/library/stdtypes.html#str)beginning with a JSON document) and return a 2-tuple of the Python representation and the index in s where the document ended.

This can be used to decode a JSON document from a string that may have extraneous data at the end.

Опис можливостей html

This module defines utilities to manipulate HTML.

*html.escape(s, quote=True)*

Convert the characters &, < and > in string s to HTML-safe sequences. Use this if you need to display text that might contain such characters in HTML. If the optional flag quote is true, the characters (") and (') are also translated; this helps for inclusion in an HTML attribute value delimited by quotes, as in <ahref="...">.

*html.unescape(s)*

Convert all named and numeric character references (e.g. &gt;, &#62;, &x3e;) in the string s to the corresponding unicode characters. This function uses the rules defined by the HTML 5 standard for both valid and invalid character references, and the [list of HTML 5 named character references](https://docs.python.org/3/library/html.entities.html#html.entities.html5).

*class html.parser.HTMLParser(\*, convert\_charrefs=True)*

Create a parser instance able to parse invalid markup.

If convert\_charrefs is True (the default), all character references (except the ones in script/styleelements) are automatically converted to the corresponding Unicode characters.

An [HTMLParser](https://docs.python.org/3/library/html.parser.html#html.parser.HTMLParser) instance is fed HTML data and calls handler methods when start tags, end tags, text, comments, and other markup elements are encountered. The user should subclass [HTMLParser](https://docs.python.org/3/library/html.parser.html#html.parser.HTMLParser) and override its methods to implement the desired behavior.

This parser does not check that end tags match start tags or call the end-tag handler for elements which are closed implicitly by closing an outer element.

[HTMLParser](https://docs.python.org/3/library/html.parser.html#html.parser.HTMLParser) instances have the following methods:

*HTMLParser.feed(data)*

Feed some text to the parser. It is processed insofar as it consists of complete elements; incomplete data is buffered until more data is fed or [close()](https://docs.python.org/3/library/html.parser.html#html.parser.HTMLParser.close) is called. data must be [str](https://docs.python.org/3/library/stdtypes.html#str).

*HTMLParser.close()*

Force processing of all buffered data as if it were followed by an end-of-file mark. This method may be redefined by a derived class to define additional processing at the end of the input, but the redefined version should always call the [HTMLParser](https://docs.python.org/3/library/html.parser.html#html.parser.HTMLParser) base class method [close()](https://docs.python.org/3/library/html.parser.html#html.parser.HTMLParser.close).

*HTMLParser.reset()*

Reset the instance. Loses all unprocessed data. This is called implicitly at instantiation time.

*HTMLParser.getpos()*

Return current line number and offset.

Опис даних які буде отримувати програма

1. Параметри пошуку від користувача
2. Тип пошуку – bool

1 for searching top-n pages based on the given page

0 for searching top-n pages based on the given wiki source

1. Ресурс – str

For searching by page – the name of the page or link (str)

For searching by wiki source – the name of the source (str)

1. The num of pages that would be shown

For searching by page – str

For searching by wiki source – default 10

1. For scenario 1
2. Get the list of categories using Wiki API (list of str)
3. Choose one of the category
4. Get the title and id of the max possible num of pages from the category – json format
5. Store data in array
6. Get the num of watcher using mwviews for every page – json format
7. Format the given info and show user
8. For scenario 3
9. Get the name of source (str)
10. Get the top 10 (default num) pages from the source that were visited during following month – json format
11. Format the given info and show user