**2.1 Визначення функціональних та нефункціональних вимог до програми**

Функціональні вимоги:

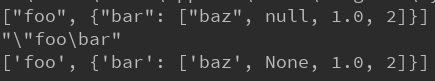
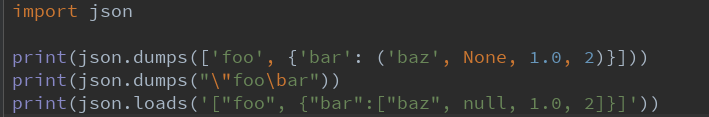
1. Програма має отримувати JSON файл та зчитувати з нього потрібну інформацію
2. Програма повинна коректно визначати категорію події, або у разі відсутності категорії групувати такі події.
3. Програма повинна визначати місце проведення події з максимальною точністю за вказаними даними.
4. Програма повинна формувати теплову карту за категоріями
5. Програма повинна мати можливість вибирати місто, у якому будуть визначатись ці події
6. Має бути реалізована можливість переключатись між категоріями

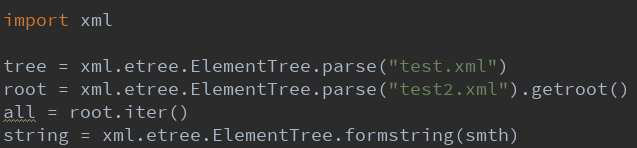
Нефункціональні вимоги:

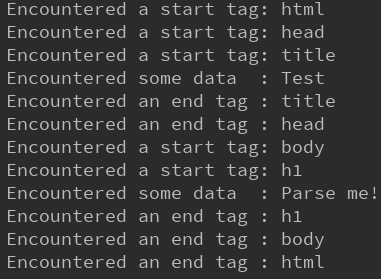
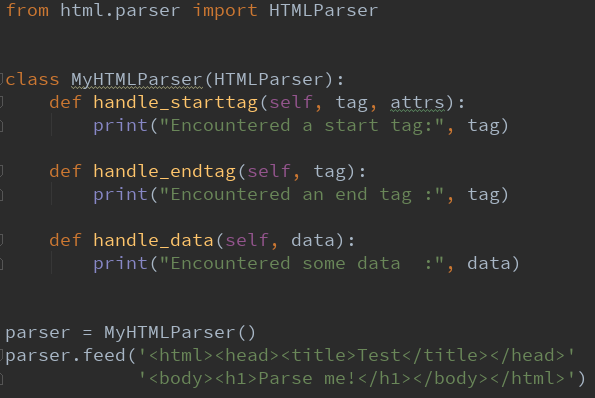
1. Додаток повинен мати простий та зрозумілий інтерфейс
2. Додаток має коректно працювати у різних браузерах
3. Ресурс має бути доступним та безкоштовним.

**2.3 Опис можливостей пакетів json, xml, html.**

Json – JavaScript Object Notation, формат подання даних. Має чітку структуру, що допомагає дуже просто і зручно формувати словники (dict) у python`і та проводити зворотну процедуру. Використовується для передачі даних між сервером та іншою програмою. Сучасні API найчастіше використовують саме цей тип файлів для обміну інформацією.

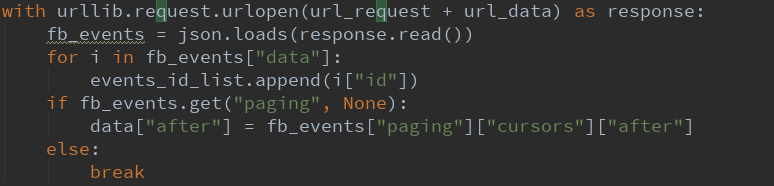
Пакет json для python спрощує роботу з файлами цього типу, адже вся інформація у файлі зберігається за структурою ключ: значення. Основними функціями dump(s) та load(s). dump – формує закодований у json формат текст у fp (параметр який підтримує метод write). Dumps – формує json стрічку. Load і loads навпаки перетворюють json стрічку у відповідні типи

Xml – Extensible Markup Language, формат передачі даних, який використовувався до json та ще досі використовується, проте є не таким зручним та безпечним. Дані є ієрархічно структуровані. У python є пакет xml, який дозволяє працювати з даними у форматі xml. Клас ElementTree представляє весь xml файл як структуру даних, а клас Element лише певну гілку. Можна проходитись по тегам файлу або додавати нові. Об’єкт класу ElementTree репрезентує xml файл. Об’єкт класу Element з коренем ElementTree.parse(“filename.xml”).getroot().

HTML -- HyperText Markup Language, стандартна мова розмітки веб-сторінок в Інтернеті.

**2.4 Опис даних які буде отримувати програма з Інтернету та з якими буде потрібно працювати програмі.**

Основним завданням моєї курсової є отримання інформації про події за допомогою FB Graph API у форматі json файлу та подальшої роботи з нею (розміщення на карті, поділ по категоріям тощо). Під час першого етапу вже був зданий файл з прикладом використання даних отриманих з Graph API.



Тут обробляється response – об’єкт класу http.client.HTTPResponse, отриманий завдяки бібліотеці url.request. Далі з цього об’єкту формується стрічка завдяки методу read, а потім словник завдяки методу loads бібліотеки json. Далі я отримую id подій, що можна дуже легко зробити завдяки чіткій структурі json файлів, та додаю їх у список. Далі аналогічним методом формую словник проводжу пошук по id подій та перевірю чи є у них категорія, формую словник де ключ – категорія, а значення – назва події. Якщо у події немає вказаної категорії проте є опис, то вона додається у цей словник під ключем ‘NONE’.