Визначення функціональних та нефункціональних вимог до програми, яка буде розроблятися.

|  |  |
| --- | --- |
| Функціональні вимоги | |
| Доступ | Надсилаючи нам дані, користувач дає згоду на їхню обробку. |
| Пошук | Веб-додаток дозволяє використовувати його тільки для міста Львів. |
| Інтерфейс | Інтерфейс веб-додатку дозволить усім користувачам легко розібратися в ньому і відправити скаргу. |
| Нефункціональні вимоги | |
| Доступність | Веб-додаток буде доступний для будь-якого користувача. Основа вимога, що потрібно мати пошту gmail. |
| Конфіденційність | 1. Усі дані, які будуть надавати нам користувачі, ні в якому разі не будуть поширені стороннім персонам. 2. Дані можуть бути надані тільки працівникам, які відповідають за вирішення даної проблеми. |
| Підтримка | Якщо в користувача виникне якесь питання або помилка, то вони зможуть завжди задати відіслати нам повідомлення про це. |
| Потенціал | На даний момент веб-додаток буде працювати тільки зі скаргами на непрацюючі ліхтарі, але в майбутньому можливе збільшення можливостей |

Можливості пакетів json, xml та html

* JSON(JavaScript Object Notation) Package – простий формат обміну даними. Пакет JSON даних майже однаковий з пайтонівським словником. Він є коротшим за XML і може бути швидше спаршений браузером, оскільки використовує JavaScript синтаксис. Це робить JSON найкращим шляхом передачі даних між клієнтом і сервером.

Функціональні можливості:

* В основному модуль JSON використовується для конвертування пайтонівського словника у текст(string) JSON, який може буде записаний у формат для подальшого використання.

З цим нам допоможе методи json.dump() і json.dumps().

* json.dump() – записує JSON об’єкт у файл.
* json.dumps() – виводить JSON об’єкт у вигляді тексту.
* Протилежна дія до конвертування словника Пайтон у текст JSON – це отримувати текст JSON і конвертувати його назад в пайтонівський словник.

Для цього ми використовуємо методи json.load() і json.loads().

* json.load() – завантажує дані JSON з файлу.
* json.loads() – завантажує дані JSON з тексту.
* Так званий “Гарний друк”(Pretty printing), коли в консолі виводить або записується у файл у вигляді JSON даних.
* HTML Package – це модуль у Пайтоні, який дозволяє згенерувати HTML-код для таблиць і списків за допомогою Пайтона. Програмісти без знання HTML зможуть створювати прості веб-додатки, використовуючи іншої мови програмування(Пайтона).

Функціональні можливості:

* Таблиця складається з рядків і стовпчиків, тому в пайтоні ми можемо її уявити як список списків(list of lists). Тому в генеруванні сторінки нам допоможе метод HTML.table()
* XML Package – це модуль, який є подібним до JSON, але, оскільки він написаний на C, то його швидкість набагато більша.

Дані, які буде отримувати програма з інтернету та з якими вона буде працювати

* За допомого Google автентифікації я зможу отримувати такі дані від певного користувача:
* Ім’я
* Стать
* Вік
* Локація(при створенні пошти)

Ці дані дозволять сформувати акаунт користувача, завдяки якому вони будуть у змозі подавати скаргу не непрацюючі ліхтарі.

* Використовуючи Google Geocoding API будуть отримані наступні дані:
  + Локація точки, поставленої на карті.
  + Вулиця, яку зазначить користувач при відсиланні скарги.
  + Назва вулиці, де буде поставлена користувачем мітка.
  + Відстань і найшвидший маршрут від комунального підприємства до непрацюючого користувача(буде виконуватися по бажанню відповідальних осіб. Тобто деколи не буде необхідності їхати і міняти лампочку).

Ставимо на карті і пишемо вручну вулицю, бо не завжди легко знайти точне місце на карті, і перевіряємо чи поставлена мітка знаходиться на даній вулиці. Якщо ні, - то просимо виправити мітку.