**Функціональні та нефункціональні вимоги до програми**

**Функціональні вимоги:**

**Вимоги щодо роботи:**

*(Процеси, які повинна виконуати програма)*

* Програма повинна дозволяти користувачам реєструватися та авторизуватися;
* Програма повинна дозволяти користувачу обирати свій настрій та виконавця;
* На основі обраних даних, програма повинна рекомендувати користувачу пісні для прослуховування;
* Програма повинна дозволяти користувачу обирати пісню та видавати її емоційну забарвленість та іншу інформацію про композицію;
* Програма повинна містити контактну форму.

**Вимоги щодо інформації:**

*(Інформація, яку повинна містити програма)*

* Програма повинна містити інформацію про дискографію обраного користувачем виконавця(кількість альбомів, кількість пісень, тематика компзицій, найчастіше вживані слова, емоційна забарвленість пісень);
* Програма повинна містити інформацію про обрану користувачем пісню(емоційна забарвленість, тематика, ключові слова).

**Нефункціональні вимоги:**

**Операційні вимоги:**

*(Середовища, у яких буде працювати програма)*

* Програма повинна працювати на ПК та мобільних пристроях;
* Програма повинна працювати у будь-якому Веб-браузері;

**Вимоги роботи:**

*(Швидкість, продуктивність, надійність програми)*

* Будь-яка взаємодія між користувачем і програмою не повинна перевищувати 5 секунд;
* Програма повинна бути доступною до використання 24 години на день, 365 днів на рік;
* Програма повинна одночасно підтримувати від 50 користувачів;

**Вимоги безпеки:**

*(Хто та за яких обставин має авторизований доступ до програми)*

* Тільки розробник може змінювати інтерфейс та функціональні можливості програми;
* Окрім розробника, всі користувачі мають однаковий рівень доступу до програми та можуть користуватися усіма доступними функціями, які забезпечує програма.
* Програма повинна гарантувати відсутність вірусів, черв’яків, троянських конів і тд.

**Культурні та політичні вимоги:**

*(Культурні, політичні фактори та юридичні вимоги, від яких залежить програма)*

* Доступ до програми повинен забезпечуватися з будь-якої країни;
* Програма не повинна порушувати авторські права виконаців.

**Можливості пакетів JSON, XML та HTML**

**JSON**

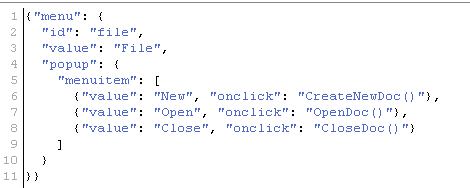
**Що це?**

JSON(JavaScript Object Notation) ー формат для обміну та збереження даних, який базується на JavaScript, є легким для читання людьми та обробки й генерування програмами.

**Можливості:**

* Перетворює будь-який об'єкт JavaScript у JSON і надсилає JSON на сервер;
* Перетворює будь-який JSON, отриманий з сервера, на об'єкти JavaScript;
* Працює з даними, як з об'єктами JavaScript, без складного парсингу.

**Приклад:**



**XML**

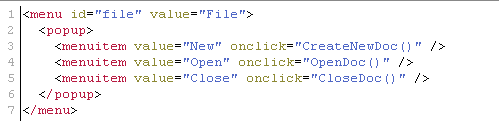
**Що це?**

XML(eXtensible Markup Language) ー це мова розмітки, яка визначає набір правил для кодування документів у форматі, що читається як людиною, так і машиною, і призначена для передачі та збереження даних.

**Можливості:**

* Зберігає дані в простому текстовому форматі;
* Забезпечує незалежний від програмного забезпечення спосіб зберігання та транспортування даних;
* Полегшує розширення або оновлення до нових операційних систем, додатків або нових веб-браузерів без втрати даних.

**Приклад:**

****

**HTML**

**Що це?**

HTML(HyperText Markup Language) ー це мова розмітки для створення та відображення веб-сторінок та веб-додатків. Елементи HTML представлені тегами, які браузери не показують, але використовують їх для відтворення вмісту сторінки.

**Можливості:**

* Створює структурований документ шляхом позначення структурного складу тексту;
* Зберігає дані, та відображає їх у зручному для читанні вигляді;
* Отримує інформацію з Інтернету через гіперпосилання;
* Окрім тексту зберігає та відображає зображення, відео, аудіо та інші об’єкти;
* Створює інтерактивні форми

**Приклад:**

.

**Опис даних**

При роботі з Musixmatch API та ALIEN Text Analysis API дані можна отримувати як в JSON, так і в XML форматі.Зважаючи на ряд причин, я вирішила зберігати для подальшого опрацювання дані у форматі JSON.

**Переваги JSON над XML:**

* JSON коротший;
* JSON швидше читати та записувати;
* JSON може використовувати масиви;
* XML потрібно аналізувати за допомогою парсера XML, тоді як JSON може бути проаналізовано стандартною функцією JavaScript;
* XML набагато складніше розібрати, ніж JSON;
* JSON аналізується в готовому для використання об'єкті JavaScript;
* JSON працює швидше і простіше, ніж XML:



**Musixmatch API**

Одним із необхідних запитів під час розробки даної програми - є запит artist.get для отримання інформації про певного виконавця. Вище наведено приклад отриманих даних у форматі JSON після запиту з виконавцем “Queen”.

**Структура отриманих даних:**

1. Об’єкт, який містить усю зібрану інформацію (неупорядкований набір пар ім’я / значення. Виділяється фігурними дужками { } );
2. Члени об’єкту у вигляді пар string: number ("artist\_id": 118, "artist\_rating": 91 );
3. Члени об’єкту у вигляді пар string: string ("artist\_name": "Queen", "artist\_country": "GB");
4. Член об’єкту у вигляді пари string: array ("artist\_alias\_list": [{"artist\_alias": "\u5973\u738b"}]). Масив(array) - впорядкована колекція значень. Виділяється квадратними дужками [ ].
5. Члени об’єкту у вигляді пари string: object("header": {"status\_code": 200,"execute\_time": 0.071532011032104}).

**ALIEN Text Analisys API**

Схоже, використовуючи ALIEN Text Analisys API, після аналізу тональності тексту, буде отримано дані у форматі JSON, який містить об’єкт з парами string: string та string: number.

