**Завдання:**

1. Сканувати кілька сторінок. Адреси сторінок вказуємо в параметрах виклику скрипта.

2. Сканувати увесь сайт. Адресу сайту вказуємо в параметрах виклику скрипта. Додатково

можна вказувати обмеження на кількість сторінок чи глибину сканування.

**Критерії оцінювання**

Правильність роботи, швидкість роботи і споживання системних ресурсів

**Джерело:**

<http://free-templates.ru/template-view/203/index.html>

**СSS:**

<http://free-templates.ru/template-view/203/css/html.css>

<http://free-templates.ru/template-view/203/css/layout.css>

<http://free-templates.ru/template-view/203/css/print.css>

1. Провести порівняння часу виконання аналізу використаних сss правил (див. пункт 2) у html коді за допомогою створеного алгоритму та запитів у БД.

* Створити масив CSS правил та масив HTML тегів.
* Запустити таймер.
* Виконати пошук CSS правил, що не використовуються.
* Зупинити таймер.
* Записати масив CSS правил до таблиці БД та масив HTML тегів до таблиці БД.
* Запустити таймер.
* Виконати SQL запит до таблиць щоб знайти CSS правила, що не використовуються.
* Зупинити таймер.

1. Список сss правил :
   1. Наприклад з назвами тегів:

**h1 { color: blue; }  
p { padding: 5px; }  
td { background-color: #ddd; }**

* 1. Наприклад з назвами тегів та їх атрибутами: **h1, h2, h3, h4**{ padding: 1em; } **.highlight p, div .highlight ul** { margin-left: .5em; } **#main p, #main ul**{ padding-top: 1em; }

1. **Не перевіряти СSS правила, що не використовують атрибутів тегів:**
   1. **@media screen and (min-device-width: 1600px) {**

div {width: 1500px;}

**}**

* 1. **::-moz-selection** {

background:#f7e123;  
color:#fff;

}