**The guideline of the WCAG Accessibility Result Display on SERP(Google Extension)**

1.The action of this extension.

This extension displays the result of WCAG Accessibility on SERP as star image.

The user can plugin our custom extension.

And when user search with random keyword and display searched result on SERP, this extension acts as displaying the result of every searched webpage from WCAG Accessibility Guideline.

The user can turn on and off this extension.

This extension shows the result as the count of stars and the count of stars explains every webpage’s WCAG accessibility.

The user can know the amount of accessibility seeing the count of stars.

2.Structure.

There are common javascript files and I used “sublime text” editor.

The source of this extension consists of two sections.

1) Front end (backup folder)

This section shows the result of accessibility, and in here the user can turn on/off this extension.

The **mainfest.json** : this file is the main json file for extension.

You can setup the main js, main css, main html, main icon….

Here are folders: html, images, scripts, styles

-html folder

When user click this extension, the popup menu occurs.

This popup.html manages this menu.

-images folder

here are images necessary for extension.

-scripts folder.

Here are main engine js files.

Getting the urls from SERP and calculating the stars and display.

And Here is **main.js** and here are some activities in it.

**Checkloading**: waiting the result and display loading icon.

**CheckAXS**: receive the result and display the stars.

var rating=0;

if(score>=0.95) rating=5;

if(score>=0.8 && score<0.95) rating=4;

if(score>=0.6 && score<0.8) rating=3;

if(score>=0.4 && score<0.6) rating=2;

if(score>=0.2 && score<0.4) rating=1;

global server-> **url** = `<https://stackfull.me/?site=$>{url}`:

local server-> **url** = `[https://localhost:3000/?site=$](https://stackfull.me/?site=$){url}`:

And Here is **background.js** and here are some activities in it.

Here are check conditions for lighthouse. These are checked from WCAG.

Also Here is **axs**\_**testing.js** and here are some activities in it.

-styles folder.

Here are .css files for style.

2) **Back end(lighthouse\_backend)**

**B**ack end acts as the main engine for lighthouse.

The main file of backend is only **index.js.**

In this file, receiving url from front end.

And it calculates the score through the global server.

We have already created global server for this.

So user only plugin this extension into chrome browser.

**Index.js**

**function** launchChromeAndRunLighthouse:

this function is the main.

So receive the url from front end and launch chrome and run lighthouse virtually in the global server.

**3. setup steps**

**1)**Create server

There are two ways, the server is on local pc or on global server.

The “lighthouse\_backend” folder is for server.

2)plugin this extension into your chrome.

After creating server, you have to plugin this extension into your chrome.

The “backup” folder is the main plugin source.