CHAPTER 1

**INTRODUCTION**

A browserextension is a plug-in that extends the functionality of a web browser. Some extensions are authored using web technologies such as HTML, JavaScript, and CSS. Others are developed using machine code and application programming interfaces (APIs) provided by web browsers, such as NPAPI and PPAPI. Browser extensions can change the user interface of the web browser without directly affecting viewable content of a web page

Extensions are small software programs that customize the browsing experience. They enable users to tailor Chrome functionality and behaviour to individual needs or preferences. They are built on web technologies such as HTML, JavaScript, and CSS. An extension must fulfil a [single purpose](https://developer.chrome.com/single_purpose) that is narrowly defined and easy to understand. A single extension can include multiple components and a range of functionality, as long as everything contributes towards a common purpose.

User interfaces should be minimal and have intent. They can range from a simple icon, such as the [Google Mail Checker extension](https://developer.chrome.com/samples#google-mail-checker) shown on the right, to [overriding](https://developer.chrome.com/override) an entire page.

Extension files are zipped into a single .crxpackage that the user downloads and installs. This means extensions do not depend on content from the web, unlike the ordinary that the user downloads and installs web crawling apps.

* 1. ***OUTLINE OF THE PROJECT***

The main goal of the project is to develop the Chrome Extension which could increase the efficiency of the Google Search and can reduce the human effort by searching through your webpage to give you the right results relevant to the context that the client is searching.

An extension’s architecture will depend on its functionality, but many robust extensions will include multiple components:

* Manifest
* Background Script
* UI Elements
* Content Script
* Option Page

The whole process can be done with the Google Search. But Computer Science is all about optimising the technology which can reduce the human effort with less cost and also making the user experience better than ever.

***1.2* Literature Review**

Nowadays finding the required information quickly and easily on the web remains a major challenge and if the searcher has little prior knowledge of search strategies and search techniques of search engines then it becomes really difficult to search over the web. Once the search is complete, Browse capabilities provide the user with the capability to determine which items are of interest and select those to be displayed. So browse capabilities plays a major role in information retrieval.

As mentioned above the whole process can be done with the Google Search. But Computer Science is all about optimising the technology which can reduce the human effort with less cost and also making the user experience better than ever. The main objective behind this software is to improve the google search efficiently so that it produces the explicit result related to the context of the user.

The importance of browse capabilities would be lessened, if searcher resulted in high precision. The main intention of browse capabilities is to support the user in focusing on items that have the highest probability in meeting his need. There are several researchers done comparison works related to search engine performance issues. Gayakwad and Phulpagar focused on comparative analysis of re-ranking models. Working process of Meta crawler was elaborated in the works of Erik Selberg et al. Lawrence and Giles simplified the process of control meta search engine process for improving search engine process. Further resource aggregation on the Web proposed by Selberg and E-zine for meta crawler architecture. Dreilinger and Howe proposed architecture for processing queries based on multi-agent model.

* How is it different form the normal google search?

When we google search about any particular topic, for instance say, when we open.Stack overflow and read about some problem and when that problem does not meet our requirement we will have to again search the same problem over the internet all over again.

* 1. ***PROBLEM STATEMENT***

The World Wide Web has grown over the years to a size that became hard to foresee. It is claimed that indexable WWW can have more than 11.5 billion pages. It is no wonder that such a powerful tool attracts commerce and WWW gives: to the web users and to their owners is also the problem that the designer of the web site needs to face. Having many web sites to choose from, user can ditch one web site if it is too hard to browse, designers are challenged to fulfil the goal of users’ navigational requirements. Beside the business needs for high quality, the web site quality evaluation is needed because of many existing ad-hoc solution for the web site design. This continuous measure of quality is also required since, although designers are following standards of the high quality layout, still they cannot predict the gap between their expectations and actual usage of designed site. Support for the web site changes decisions is needed. Using the one outcome of the appropriate evaluation techniques one can improve existing site and if possible, use gained knowledge for future designs. The problems that need to be solved is to identify that whether the content is useful to the user or not.

This projects help to solve such kind of problem “DeepCrwaler” will be useful. It will provide the relevant information to the user when the extension is pushed. The software is designed simple and easy to use.

***1.4 Objective***

To developing such a computerization system is to improve the efficiency of the web searching and reduce the time spent while searching on different topics of time and to provide the precise information.

## Chapter 2

**Algorithms and Methods**

Browser extensions are used for improving a browser's user interface, security or accessibility, blocking advertisements, and various other features to make browsing the internet easier and more pleasant. There are many types of extensions that can be used to control various aspects of browsing privacy and mitigate threats. For instance, they may prevent third parties from tracking the user's movements, block ads and scripts, or enforce good habits.

A browser toolbar is a common type of browser extensions that alters the user interface. It is a toolbar that resides within a browser's window. All major web browsers provide support for browser toolbars as a way to extend the browser's UI and functionality. Browser toolbars are specific to each browser, which means that a toolbar working on a browser does not work on another one.

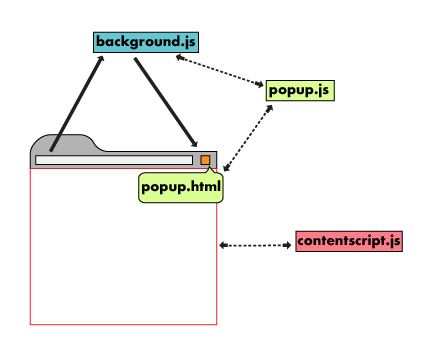
Chrome extensions can also be created to work only on certain pages through the use of Page Actions, they can run code in the background using Background Pages*,* and they can even modify an existing page loaded in the browser usingContent Scripts.

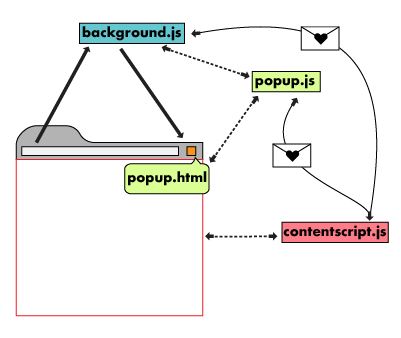
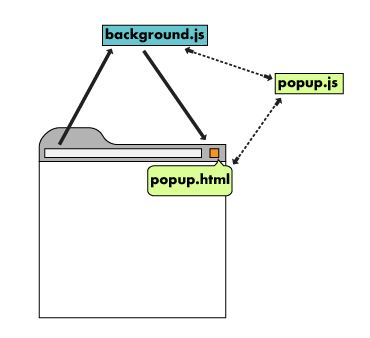
**2.1 Flow Chart**

Flow chart is a diagrammatic representation of an algorithm and essential part of planning the system. Flow charts are widely used in technical analysis and programming for easy writing programs and explaining them to others. So, one of the most popular type of flow charts is Technical Flow Chart. Technical Flow Chart can be drawn by pencil on the paper, but it will be easier to use for designing a special software

A Flowchart is a graphical representation of process, algorithm, workflow or step-by-step solution of the problem. It shows the steps as boxes of various kinds and connects them by arrows in a defined order depicting a flow. There are twelve main Flowchart types: Basic Flowchart, Business Process Modelling Diagram (BPMN), Cross Functional Flowchart, Data Flow Diagram (DFD), IDEF (Integrated Definition) Flowchart, Event-driven Process Chain (EPC) Diagram, and Influence Diagram (ID), Swim lane Flowchart, Process Flow Diagram (PFD), Specification and Description Language (SDL) Diagram, Value Stream Mapping, Workflow Diagram. Using the Flowcharts solution from the What is a Diagram area of Concept Draw Solution Park you can eay and quickly design a Flowchart of any of these types. This solution offer lot of special predesigned vector symbols for each of these widely used notations. They will make the drawing process of Flowcharts much easier than ever.

The main reason of using Process Flowchart or PFD is to show relations between major parts of the system. Process Flowcharts for single unit or multiple units differ in their structure and implementations are shown in ***Fig. 2.1.1***



  **Fig 2.1.1**

***2.2 Overview of the platform***

*2.2.1 Overview*

Extensions are zipped bundles of HTML, CSS, JavaScript, images, and other files used in the web platform, that customize the Google Chrome browsing experience. Extensions are built using web technology and can use the same APIs the browser provides to the open web.

Extensions have a wide range of functional possibilities. They can modify web content users see and interact with or extend and change the behaviour of the browser itself. Consider extensions the gateway to making the Chrome browser the most personalized browser.

2.2.2 Extension Files

Extensions vary in types of files and amount of directories, but they are all required to have a [manifest](https://developer.chrome.com/extensions/extensions/manifest). Some basic, but useful, extensions may consist of just the manifest and its toolbar icon.

The manifest file, titled manifest.json, gives the browser information about the extension, such as the most important files and the capabilities the extension might use.

{

"name": "DeepCrwaler",

"version": "1.0,

"description": "Gets information from Google.",

"icons": {

"128": "icon\_16.png",

"128": "icon\_32.png",

"128": "icon\_48.png",

"128": "icon\_128.png"

},

"background": {

"persistent": false,

"scripts": ["background\_script.js"]

},

"permissions": ["https://\*.google.com/", "activeTab"],

"browser\_action": {

"default\_icon": "icon\_16.png",

"default\_popup": "popup.html"

}

}

Extensions must have an icon that sits in the browser toolbar. Toolbar icons allow easy access and keep users aware of which extensions are installed. Most users will interact with an extension that uses a [popup](https://developer.chrome.com/extensions/user_interface#popup) by clicking on the icon.

|  |  |
| --- | --- |
| screenshot |  |
| This [**Google Mail Checker extension**](https://developer.chrome.com/extensions/samples#search:google%20mail%20checker) uses a [**browser action**](https://developer.chrome.com/extensions/browserAction). | This [**DeepCrawler extension**](https://developer.chrome.com/extensions/samples#search:mappy) uses a [**page action**](https://developer.chrome.com/extensions/pageAction) and [**content script**](https://developer.chrome.com/extensions/overview#contentScripts). |

### Fig 2.2

### *2.2.3 Referring to files*

An extension's files can be referred to by using a relative URL, just as files in an ordinary HTML page.

< src=”images/my\_image.png”>

Additionally, each file can also be accessed using an absolute URL.

chrome-extension://<extensionID>**/**<pathToFile>

In the absolute URL, the <extensionID> is a unique identifier that the extension system generates for each extension. The IDs for all loaded extensions can be viewed by going to the URL chrome://extensions. The <pathToFile> is the location of the file under the extension's top folder; it matches the relative URL.

While working on an unpacked extension the extension ID can change. Specifically, the ID of an unpacked extension will change if the extension is loaded from a different directory; the ID will change again when the extension is packaged. If an extension's code relies on an absolute URL, it can use the[chrome.runtime.getURL()](https://developer.chrome.com/extensions/extensions/runtime#method-getURL)method to avoid hardcoding the ID during development.

## *2.3 Architecture*

An extension’s architecture will depend on its functionality, but many robust extensions will include multiple components:

* [Background Script](https://developer.chrome.com/extensions/overview#background_script)
* [UI Elements](https://developer.chrome.com/extensions/overview#pages)
* [Content Script](https://developer.chrome.com/extensions/overview#contentScripts)
* [Options Page](https://developer.chrome.com/extensions/overview#optionsPage)

*2.3.1 Background Script*

The [background script](https://developer.chrome.com/background_pages) is the extension's event handler; it contains listeners for browser events that are important to the extension. It lies dormant until an event is fired then performs the instructed logic. An effective background script is only loaded when it is needed and unloaded when it goes idle.

*2.3.2* UI Elements

An [extension's user interface](https://developer.chrome.com/extensions/user_interface) should be purposeful and minimal. The UI should customize or enhance the browsing experience without distracting from it. Most extensions have a [browser action](https://developer.chrome.com/extensions/browserAction) or [page action](https://developer.chrome.com/extensions/pageAction), but can contain other forms of UI, such as [context menus](https://developer.chrome.com/contextMenus), use of the [omnibox](https://developer.chrome.com/extensions/omniBox), or creation of a [keyboard shortcut](https://developer.chrome.com/extensions/commands).

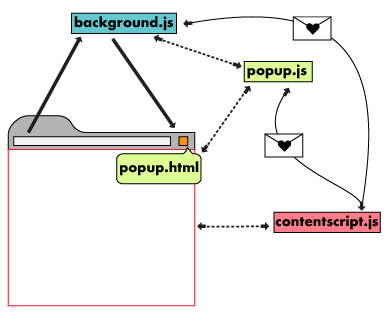
Extension UI pages, such as a [popup](https://developer.chrome.com/extensions/user_interface#popup), can contain ordinary HTML pages with JavaScript logic. Extensions can also call [tabs.create](https://developer.chrome.com/extensions/tabs#method-create) or window.open () to display additional HTML files present in the extension.

An extension using a page action and a popup can use the [declarative content](https://developer.chrome.com/extensions/declarativeContent) API to set rules in the background script for when the popup is available to users. When the conditions are met, the background script communicates with the popup to make it’s icon clickable to users.

*2.3.3 Content scripts*

Extensions that read or write to web pages utilize a [contentscript](https://developer.chrome.com/extensions/content_scripts). The content script contains JavaScript that executes in the contexts of a page that has been loaded into the browser. Content scripts read and modify the DOM of web pages the browser visits.

Content scripts can communicate with their parent extension by exchanging [messages](https://developer.chrome.com/extensions/messaging) and storing values using the [storage](https://developer.chrome.com/extensions/storage) API.



**Fig 2.3**

2.3.4 Options Page

Just as extensions allow users to customize the Chrome browser, the [options page](https://developer.chrome.com/extensions/options) enables customization of the extension. Options can be used to enable features and allow users to choose what functionality is relevant to their needs.

***2.4 Using Chrome APIs***

In addition to having access to the same APIs as web pages, extensions can also use [extension-specific APIs](https://developer.chrome.com/extensions/api_index) that create tight integration with the browser. Extensions and webpages can both access the standardwindow.open () method to open a URL, but extensions can specify which window that URL should be displayed in by using the Chrome API [tabs.create](https://developer.chrome.com/extensions/tabs#method-create) method instead.

* Asynchronous vs. synchronous methods?

Most Chrome API methods are asynchronous: they return immediately without waiting for the operation to finish. If an extension needs to know the outcome of an asynchronous operation it can pass a callback function into the method. The callback is executed later, potentially much later, after the method returns.

If the extension needed to navigate the user’s currently selected tab to a new URL, it would need to get the current tab’s ID and then update that tab’s address to the new URL.

// Signature for an asynchronous method

chrome.tabs.query(object *queryInfo*, function *callback*)

To correctly query a tab and update its URL the extension must use the callback parameter.

chrome.tabs.query({'active': true},function(tabs){

chrome.tabs.update(tabs[0].id, {url: newUrl});

});

someOtherFunction();

In the above code, the lines are executed in the following order: 1, 4, 2. The callback function specified to query() is called and then executes line 2, but only after information about the currently selected tab is available. This happens sometime after query() returns. Although update() is asynchronous the code doesn’t use a callback parameter, since the extension doesn’t do anything with the results of the update.

// Synchronous methods have no callback option and returns a type of string

string chrome.runtime.getURL()

**Chapter 3**

**System Implementation**

***3.1 Design User Interface***

The extension user interface should be purposeful and minimal. Just like extensions themselves, the UI should customize or enhance the browsing experience without distracting from it.

## *3.1.1Activate the extension on all pages*

Use a [browser\_action](https://developer.chrome.com/browserAction) when an extension's features are functional in most situations.

*3.1.2 Register browser action*

The "browser\_action" field is registered in the manifest.

{

"name": "DeepCrawler",

...

"browser\_action": {

...

}

...

}

Declaring "browser\_action" keeps the icon colorized, indicating the extension is available to users.

*3.1.3 Add a badge*

Badges display a colored banner with up to four characters on top of the browser icon. They can only be used by extensions that declare "browser\_action" in their manifest.

Use badges to indicate the state of the extension. The browser displays a badge to show the user that the extension is working correctly.

Set the text of the badge by calling [chrome.browserAction.setBadgeText](https://developer.chrome.com/browserAction#method-setBadgeText)and the banner color by calling[chrome.browserAction.setBadgeBackgroundColor](https://developer.chrome.com/browserAction#method-setBadgeBackgroundColor).

chrome.browserAction.setBadgeText({text: 'ON'});

chrome.browserAction.setBadgeBackgroundColor({color: '#4688F1'});

## Activate the extension on select pages

Use page action when an extension's features are only available under defined circumstances.

### *3.1.4 Declare Page Action*

The "page\_action" field is registered in the manifest.

{

"name": ""\_\_MSG\_sp\_title\_\_",

...

"page\_action": {

...

}

...

}

Declaring "page\_action" will colorize the icon only when the extension is available to users, otherwise it will be displayed in greyscale.

*3.1.5 Define rules for activating the extension*

Define rules for when the extension is usable by calling [chrome.declarativeContent](https://developer.chrome.com/declarativeContent) under the[runtime.onInstalled](https://developer.chrome.com/runtime#event-onInstalled) listener in a [background script](https://developer.chrome.com/background_pages). The [Page action by URL](https://developer.chrome.com/samples#search:page%20action%20by%20url) sample extension sets a condition that the url must include a ‘g’. If the condition is met, the extension calls [declarativeContent.ShowPageAction()](https://developer.chrome.com/pageAction#method-show).

chrome.tabs.onUpdated.addListener(function (a, b, c) {

{

-1 == c.url.indexOf("chrome://") && -1 == c.url.indexOf("chrome-extension://") && -1 == c.url.indexOf("file://") || chrome.pageAction.setIcon({

tabId: a,

path: "images/sp-disabled-19.png"

});

chrome.pageAction.show(a)

});

window.addEventListener("load", function () {

chrome && (chrome.runtime.onMessage.addListener(function (a, b, c) {

switch (a.command) {

case "get\_request\_domain":

c(window.localStorage.request\_domain);

break;

case "set\_request\_domain":

window.localStorage.request\_domain = a.data

}

}), e())

}, !1);

*3.1.6 Enable or disable the extension*

Extensions using "page\_action" can activate and disable dynamically by calling [pageAction.show](https://developer.chrome.com/pageAction#method-show) and [pageAction.hide](https://developer.chrome.com/pageAction#method-hide).

The [Mappy](https://developer.chrome.com/extensions/samples#search:mappy)sample extension scans a web page for an address and shows its location on a static map in the [popup](https://developer.chrome.com/extensions/user_interface#popup). Because the extension is dependent on page content, it cannot declare rules to predict which pages will be relevant. Instead, if an address is found on a page it calls pageAction.show to colorize the icon and signal the extension is usable on that tab.

chrome.runtime.onMessage.addListener(function(req, sender) {

chrome.storage.local.set({'address': req.address})

chrome.pageAction.show(sender.tab.id);

chrome.pageAction.setTitle({tabId: sender.tab.id, title: req.address});

});

## *3.1.7Provide the extension icons*

Extensions require at least one icon to represent it. Provide icons in PNG format form the best visual results, although any format supported by WebKit including BMP, GIF, ICO, and JPEG is accepted.

### Designate toolbar icons

Icons specific to the toolbar are registered in the "default\_icon" field under [browser\_action](https://developer.chrome.com/browserAction) or[page\_action](https://developer.chrome.com/pageAction) in the manifest. Including multiple sizes is encouraged to scale for the 16-dip space. At minimum, 16x16 and 32x32 sizes are recommended.

{

"name": " \_\_MSG\_sp\_title\_\_",

...

"page\_action": {

"default\_icon": "images/sp-19.png",

"default\_popup": "popup.html",

"default\_title": "\_\_MSG\_sp\_default\_title\_\_" }

}

...

}

All icons should be square or they may be distorted. If no icons are supplied, Chrome will add a generic one to the toolbar.

### *3.1.8 Create and register additional icons*

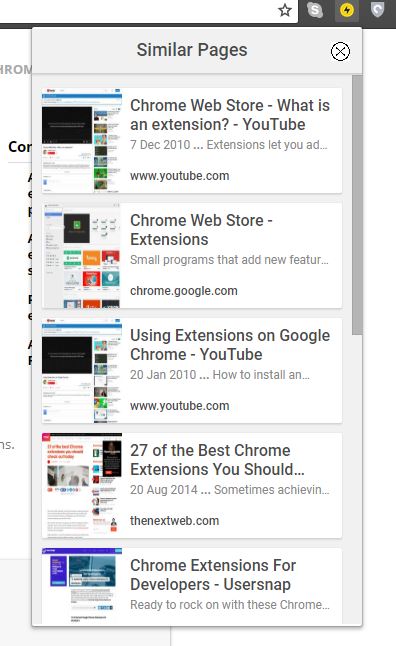
Include additional icons in the following sizes for uses outside of the toolbar.

|  |  |
| --- | --- |
| **Icon Size** | **Icon Use** |
| 16x16 | favicon on the extension's pages |
|  |  |
| 32x32 | Windows computers often require this size. Providing this option will prevent size distortion from shrinking the 48x48 option. |
|  |  |
| 48x48 | displays on the extensions management page |
|  |  |
| 128x128 | displays on installation and in the Chrome Web store |

## Additional UI Features

### *3.1.9 Popup*

A popup is an HTML file that is displayed in a special window when the user clicks the toolbar icon. A popup works very similarly to a web page; it can contain links to style sheets and script tags, but does not allow inline JavaScript.The [DeepCrawler](https://developer.chrome.com/extensions/samples#search:drink)example popup displays available timer options. Users set an alarm by clicking one of the provided buttons.



Popups can also be set dynamically by calling [**browserAction.setPopup**](https://developer.chrome.com/browserAction#method-setPopup) or [**pageAction.setPopup**](https://developer.chrome.com/pageAction#method-setPopup).

chrome && chrome.tabs.getSelected(null, function (a) {

var b = a.url,

c = window.localStorage.request\_domain; - 1 == b.search(B) ? D(b, c) : z(a, function (a) {

D(a.url, c)

})

});

### *3.1.10 Tooltip*

### Use a tooltip to give short descriptions or instructions to users when hovering over the browser icon.



Tooltips can also be set or updated by calling [browserAction.setTitle](https://developer.chrome.com/browserAction#method-setTitle) and [pageAction.setTitle](https://developer.chrome.com/pageAction#method-setTitle).

chrome.browserAction.onClicked.addListener(function(tab) {

chrome.browserAction.setTitle({tabId: tab.id, title: + tab.id});

});

Specialized locale strings are implemented with [Internationalization](https://developer.chrome.com/i18n). Create directories to house language specific messages within a folder called \_locales. The following image shows a file path for an extension that supports English locales.



Format messages inside of each language's messages.json.

{

"sp\_options\_about\_google": {

"message": "\u003Ca href='http://www.google.com/intl/$1/about.html'>About Google\u003C/a>",

"placeholders": {

"1": {

"content": "$1"

}

}

},

}

### *3.1.11 Context Menu*

Add new [context menu](https://developer.chrome.com/contextMenus) options by granting the "contextMenus" permission in the manifest.Create a context menu by calling [contextMenus.create](https://developer.chrome.com/contextMenus#method-create) in the [background script](https://developer.chrome.com/background_pages). This should be done under the [runtime.onInstalled](https://developer.chrome.com/runtime#event-onInstalled) listener event.

chrome.runtime.onInstalled.addListener(function() {

for (let key of Object.keys(kLocales)) {

chrome.contextMenus.create({

id: key,

title: kLocales[key],

type: 'normal',

contexts: ['selection'],

});

}

});

const kLocales = {

'com.au': 'Australia',

'com.br': 'Brazil',

'ca': 'Canada',

'cn': 'China',

'fr': 'France',

'it': 'Italy',

'co.in': 'India',

'co.jp': 'Japan',

'com.ms': 'Mexico',

'ru': 'Russia',

'co.za': 'South Africa',

'co.uk': 'United Kingdom'

};

The Global Google Search context menu example creates multiple options from the list in [locales.js](https://developer.chrome.com/extensions/examples/api/contextMenus/global_context_search/locales.js). When an extension contains more than one context menu, Google Chrome automatically collapses them into a single parent menu.

Commands

Extensions can define specific [commands](https://developer.chrome.com/commands) and bind them to a key combination. Register one or more commands in the manifest under the "commands" field.

chrome.commands.onCommand.addListener(function(command) {

chrome.tabs.query({currentWindow: true}, function(tabs) {

// Sort tabs according to their index in the window.

tabs.sort((a, b) => { return a.index < b.index; });

let activeIndex = tabs.findIndex((tab) => { return tab.active; });

let lastTab = tabs.length - 1;

let newIndex = -1;

if (command === 'flip-tabs-forward')

newIndex = activeIndex === 0 ? lastTab : activeIndex - 1;

else // 'flip-tabs-backwards'

newIndex = activeIndex === lastTab ? 0 : activeIndex + 1;

chrome.tabs.update(tabs[newIndex].id, {active: true, highlighted: true});

});

});

Because the extension defines a [broswer\_action](https://developer.chrome.com/user_interface#browser) it can specify "execute\_browser\_action" in the commands to open the popup file without including a [background script](https://developer.chrome.com/background_pages). If using [page\_action](https://developer.chrome.com/user_interface#page), it can be replaced with "execute\_page\_action". Both browser and extension commands can be used in the same extension.

Override Pages :

An extension can [override](https://developer.chrome.com/override) and replace the History, New Tab, or Bookmarks web page with a custom HTML file. Like a [popup](https://developer.chrome.com/extensions/user_interface#popup), it can include specialized logic and style, but does not allow inline JavaScript. A single extension is limited to overriding only one of the three possible pages.

Register an override page in the manifest under the "chrome\_url\_overrides" field.

{

"name": "Awesome Override Extension",

...

"chrome\_url\_overrides" : {

"newtab": "override\_page.html"

},

...

}

The "newtab" field should be replaed with "bookmarks" or "history" when overriding those pages.

<html>

<head>

<title>Deep Crawler</title>

<link href="https://fonts.googleapis.com/css?family=Roboto:100,300,400,500,700" rel="stylesheet" type="text/css">

<link type="text/css" rel="stylesheet" href="popup.css">

<script src="popup\_main.js"></script>

</head>

<body>

<body style="max-height:590px; overflow-y: auto">

</body>

</html>

***3.2 Modular Design***

The last thing we need to do to create the plugin is to implement the logic that should execute when a user clicks the “Check this page now!” button inside of a tab.

We’ll want to add an event listener to listen for the click event on the checkPagebutton. When it is clicked, we’ll need to create a new form to submit to Deep Crawler that contains the URL of the current page, submits it, and then displays the result.

# Content Scripts

Content scripts are files that run in the context of web pages. By using the standard Document Object Model (DOM), they are able to read details of the web pages the browser visits, make changes to them and pass information to their parent extension.

## *3.2.1 Understand Content Script Capabilities*

Content scripts can access Chrome APIs used by their parent extension by exchanging [messages](https://developer.chrome.com/extensions/messaging) and access information by making [cross-site XMLHttpRequests](https://developer.chrome.com/extensions/xhr) to parent sites. They can also access the URL of an extension's file withchrome.runtime.getURL() and use the result the same as other URLs.

*//Code for displaying <extensionDir>/images/myimage.png:*

var imgURL =chrome.runtime.getURl(“images/myimage.png”);

document.getElementById("someImage").src = imgURL;

Additionally, content script can access the following chrome APIs directly:

* [i18n](https://developer.chrome.com/extensions/i18n)
* [storage](https://developer.chrome.com/extensions/storage)
* [runtime](https://developer.chrome.com/extensions/runtime):
  + [connect](https://developer.chrome.com/extensions/runtime#method-connect)
  + [getManifest](https://developer.chrome.com/extensions/runtime#method-getManifest)
  + [getURL](https://developer.chrome.com/extensions/runtime#method-getURL)
  + [id](https://developer.chrome.com/extensions/runtime#property-id)
  + [onConnect](https://developer.chrome.com/extensions/runtime#event-onConnect)
  + [onMessage](https://developer.chrome.com/extensions/runtime#event-onMessage)
  + [sendMessage](https://developer.chrome.com/extensions/runtime#method-sendMessage)

Content scripts are unable to access other APIs directly.

## *3.2.2 Work in Isolated Worlds*

Content scripts live in an isolated world, allowing a content script to makes changes to its JavaScript environment without conflicting with the page or additional content scripts.

chrome.tabs.onUpdated.addListener(function (a, b, c) {

-1 == c.url.indexOf("chrome://") && -1 == c.url.indexOf("chrome-extension://") && -1 == c.url.indexOf("file://") || chrome.pageAction.setIcon({

tabId: a,

path: "images/sp-disabled-19.png"

});

chrome.pageAction.show(a)

});

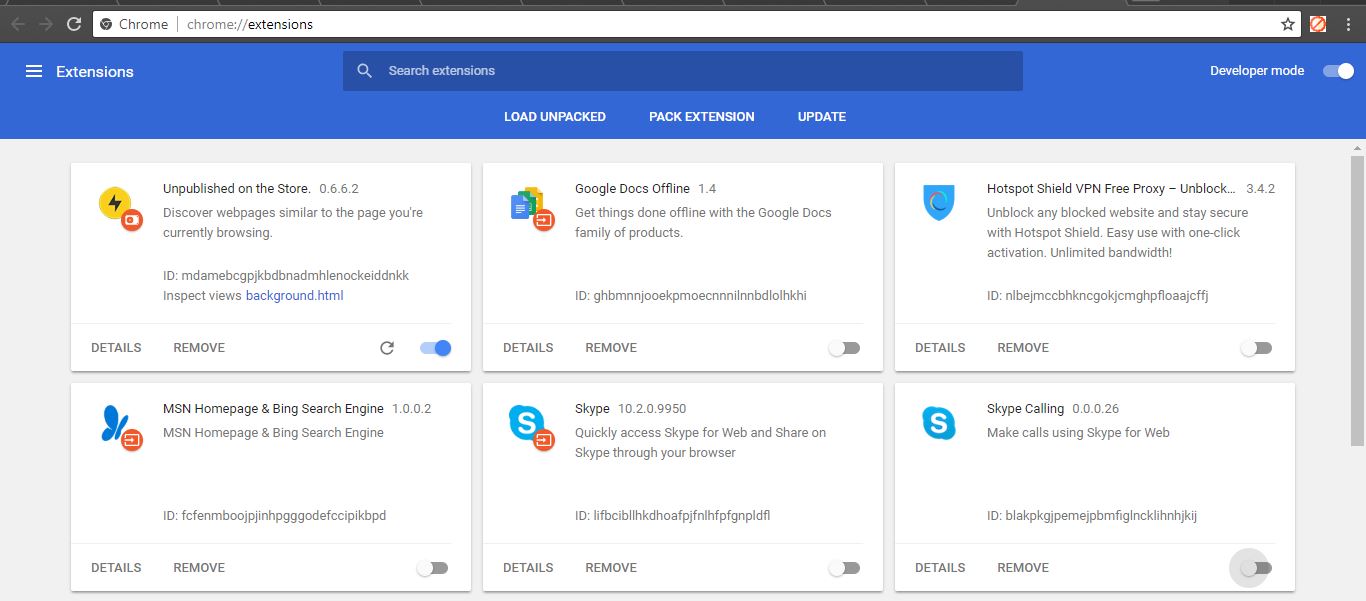
***3.3 Publishing to the store***

The directory holding the manifest file can be added as an extension in developer mode in its current state.

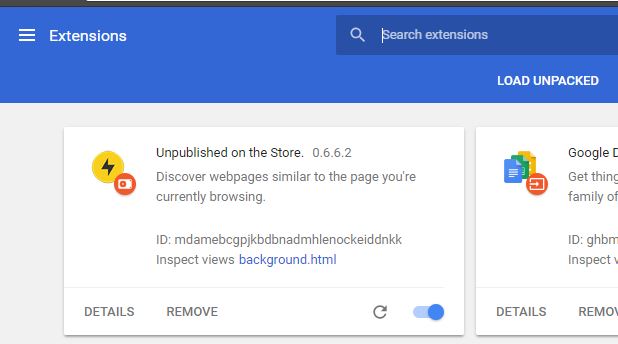
1. Open the Extension Management page by navigating to chrome://extensions.

The Extension Management page can also be opened by clicking on the Chrome menu, hovering over More Tools then selecting Extensions.

1. Enable Developer Mode by clicking the toggle switch next to Developer mode.
2. Click the LOAD UNPACKED button and select the extension directory.



The extension has been successfully installed. Because no icons were included in the manifest, a generic toolbar icon will be created for the extension.



# *3.3.1 What is the Chrome Web Store?*

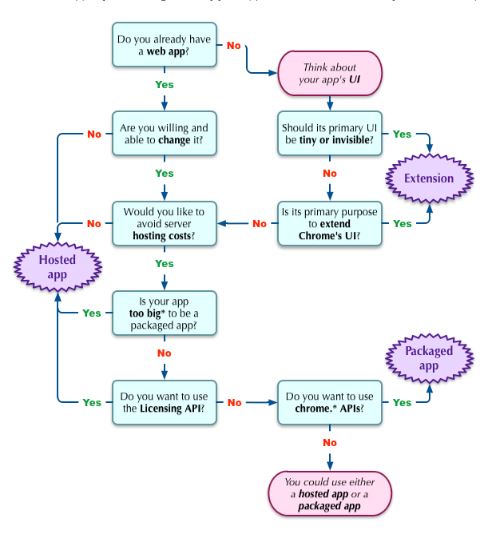
The [Chrome Web Store](http://chrome.google.com/webstore) lets you publish apps and games—either free or paid—where Google Chrome users can easily find them. Whenever Chrome users create a new window or tab, they see the apps they've installed, along with a store icon that reminds them to find more apps.

If you already have a web app, you can create a metadata file and then publish your app in the store within minutes. Or you can create and publish an app that's specialized for Chrome, such as an extension for Chrome or a web app that's been modified to be downloaded and run on the user's computer.

Your app can be free, or you can charge for it using either Chrome Web Store Payments or your own custom payment system. A [Licensing API](https://developer.chrome.com/webstore/check_for_payment.html) lets you check whether the current user has paid for your app with Chrome Web Store Payments.

To upload and publish apps, use the [Chrome Developer Dashboard](https://chrome.google.com/webstore/developer/dashboard).

*3.3.2 Flow Diagram of Extension*

****

To publish items to the Chrome Web Store, you first use the [Chrome Developer Dashboard](https://developer.chrome.com/webstore/overview) to upload your app and specify how its listing should look. Then you can publish it either to the world at large or to a few trusted testers. For details, see Publishing Your App

**Chapter 4**

**Conclusion and Future Work**

***4.1 Conclusion***

Analysis and implementation of collection fusion in MSEs reported, seems to be promising in improving the web search. Research into MSEs has been mainly focused on source engine selection, re-ranking and Integration of multiple Search Engine and its results. Apart from design and evaluation of full research in distributed search or the continuous google search, database selection and resulting merging combination techniques have a direct and result merging combination techniques have a direct bearing on the domain of meta-searching. Recent smarted Meta search includes the data clustering and analysis of the Meta data and then producing to the browser. With the Emergence of the web 2.0 services like social networking and weblogs and feeds increases the non-textual information like podcasting, and online-video and convergence of the non-conventional forms of communication such as mobile phones with Internet will require easy information retrieval in these areas.

**4.2 Future Works**

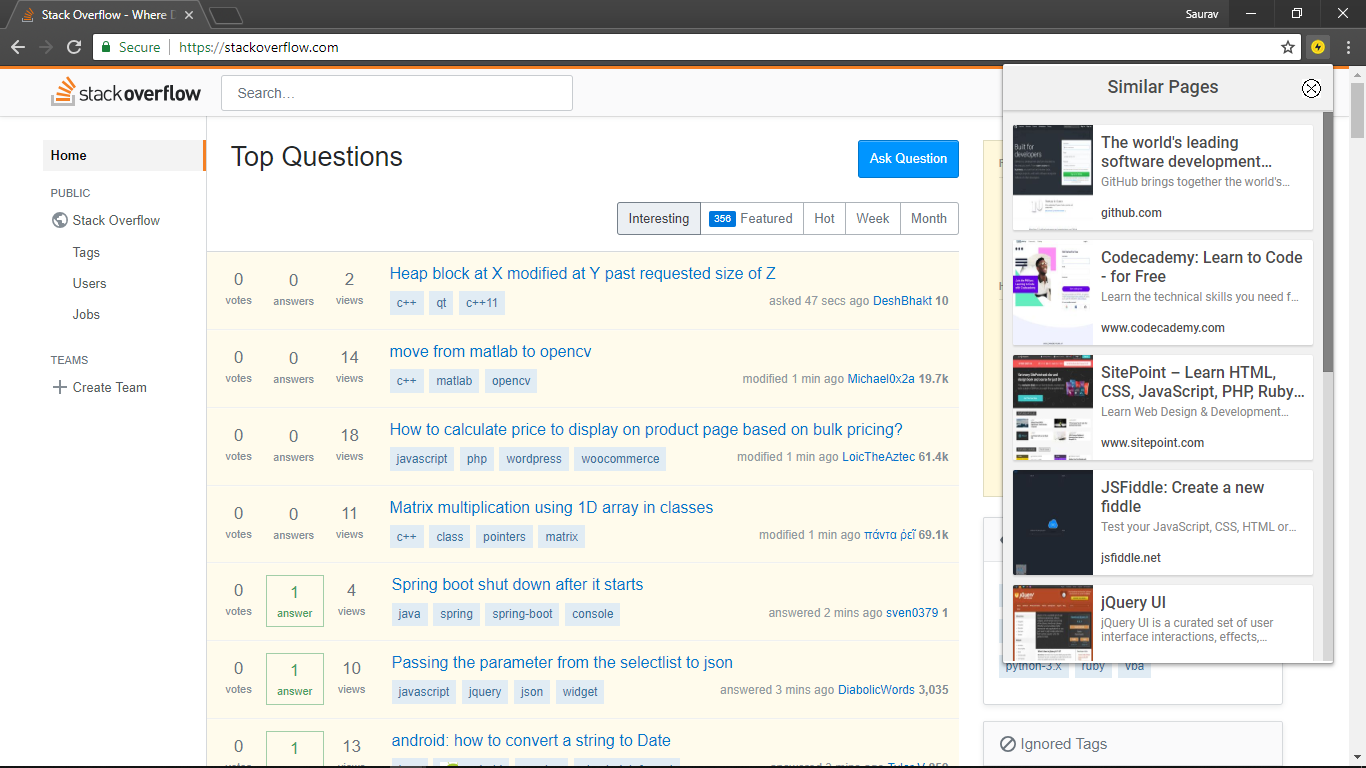
DeepCrwaler would be helpful in many ways giving the extensive support to the users like developers, bloggers, readers and those people who lack the knowledge of deep google searching. It is not only focusing on the development of the application but also to the growth trajectory of a company by enabling the many features which will be improve the revenue model. Moreover it will improve the Search Engine Optimization by providing more relevant links to its clients. It will also help in the sector like research and development by identifying the unanswered questions ease of searching.

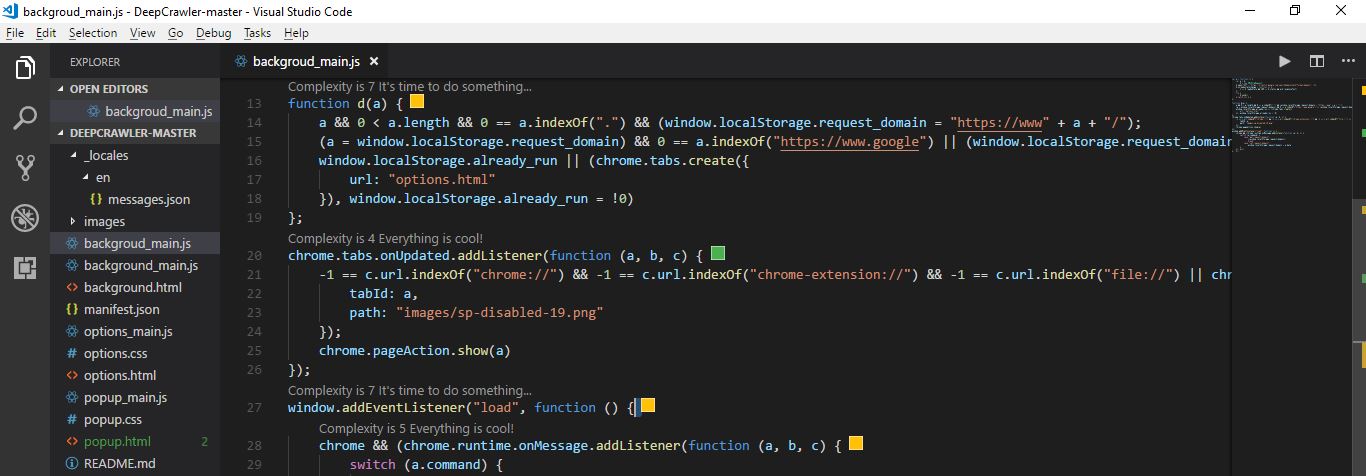
Deep Crawler will become more adhesive when we will be able to analysis the data and could be able to produce most relevant and likeable result to the client and to enable such requirements could be met by using machine learning and artificial intelligence.

**Chapter 5**

**Reference Appendix**

How the Extension will look?



**Source Code:**

***Manifest.json***

{

"background": {

"page": "background.html"

},

"default\_locale": "en",

"description": "\_\_MSG\_sp\_description\_\_",

"icons": {

"128": "images/sp-128.png",

"19": "images/sp-19.png",

"32": "images/sp-32.png",

"48": "images/sp-48.png"

},

"manifest\_version": 2,

"name": "\_\_MSG\_sp\_title\_\_",

"options\_page": "options.html",

"page\_action": {

"default\_icon": "images/sp-19.png",

"default\_popup": "popup.html",

"default\_title": "\_\_MSG\_sp\_default\_title\_\_"

},

"permissions": [ "tabs", "https://\*/", "http://\*/" ],

"update\_url": "https://clients2.google.com/service/update2/crx",

"version": "0.6.6.2"

}

**Popup.html**

<html>

<head>

<link href="https://fonts.googleapis.com/css?family=Roboto:100,300,400,500,700" rel="stylesheet" type="text/css">

<link type="text/css" rel="stylesheet" href="popup.css">

<script src="popup\_main.js"></script>

</head>

<body style="max-height:590px; overflow-y: auto"></body>

</html>

**Popup.css**

/\* Top-level containers \*/

body {

margin: 0;

background-color: #f1f1f1;

height: 100%;

}

.SP\_Hide {display: none}

.SP\_MainContainer, .SP\_ZeroResultsContainer, .SP\_SpecialCaseContainer {

font-family:sans-serif;

font-size: 8pt;

}

.SP\_MainContainer, .SP\_SpecialCaseContainer {

margin: 0px 0px 10px 0px;

}

.SP\_MainContainer {

min-width: 330px;

margin-bottom: 5px;

background: #f1f1f1;

width: 320px;

max-height: 100%;

top: 0;

left: 0;

z-index: 20;

}

::-webkit-scrollbar {

width: 10px;

height: 10px;

background: #f1f1f1;

}

::-webkit-scrollbar-track {

border-radius: 0px;

}

::-webkit-scrollbar-track:hover {

background-color: rgba(0,0,0,0.1);

}

::-webkit-scrollbar-thumb {

border-radius: 0px;

background-color: rgba(0,0,0,0.25);

-webkit-box-shadow: inset 1px 1px 0 rgba(0,0,0,0.10),inset 0 -1px 0

rgba(0,0,0,0.07)

}

::-webkit-scrollbar-thumb:hover {

background-color:rgba(0,0,0,.4)

}

.SP\_ZeroResultsContainer, .SP\_SpecialCaseContainer {

color: #666666;

}

.SP\_ZeroResultsContainer {

font-weight: bold;

padding-left: 20px;

padding-top: 10px;

width: 180px;

height: 20px;

}

.SP\_TopBar {

margin-bottom: 2px;

height: 24px;

padding: 10px;

padding-left: 0px;

color: #4a4a4a;

text-align: center;

font-weight: 500;

font-family: 'Roboto';

font-size: 18px;

-moz-box-shadow: 0px 0px 1px 0px rgba(0,0,0,0.10), 0px 2px 1px 0px rgba(0,0,0,0.10);

-webkit-box-shadow: 0px 0px 1px 0px rgba(0,0,0,0.10), 0px 2px 1px 0px rgba(0,0,0,0.10);

box-shadow: 0px 0px 1px 0px rgba(0,0,0,0.10), 0px 2px 1px 0px rgba(0,0,0,0.10);

}

.SP\_Message {

color: #4a4a4a;

text-align: center;

font-weight: 400;

font-family: 'Roboto';

font-size: 12px;

padding: 10px;

width: 300px;

}

.SP\_SpecialCaseContainer {

width: 320px;

}

.SP\_PageEntry {

width: 300px;

height: 105px;

margin: 5px 10px;

float: left;

border-radius: 2px;

background: #FFFFFF;

-moz-box-shadow: 0px 0px 1px 0px rgba(0,0,0,0.10), 0px 2px 1px 0px rgba(0,0,0,0.10);

-webkit-box-shadow: 0px 0px 1px 0px rgba(0,0,0,0.10), 0px 2px 1px 0px rgba(0,0,0,0.10);

box-shadow: 0px 0px 1px 0px rgba(0,0,0,0.10), 0px 2px 1px 0px rgba(0,0,0,0.10);

overflow: hidden;

transition: all 0.6s cubic-bezier(.4,0,.2,1);

cursor: pointer;

font-family: 'Roboto', 'Helvetica Neue', Arial, sans-serif;

}

#sp\_logo {

}

#sp\_close\_button {

float: right;

}

img#sp\_close\_button:hover {

background-color: #eeeeff;

}

#sp\_results {

overflow-x: hidden;

overflow-y: scroll;

max-height: 541px;

padding-top: 8px;

}

.SP\_PageEntryText {

width: 220px;

height: 105px;

float: left;

overflow: hidden;

text-align: left;

box-sizing: border-box;

padding: 8px 8px 8px 8px;

color:#4a4a4a;

}

.SP\_PageEntryTitle, .SP\_PageEntryTitle > a {

display: -webkit-box;

-webkit-line-clamp: 2;

-webkit-box-orient: vertical;

font-size: 16px;

font-weight: 500;

margin-bottom: 2px;

text-decoration: none;

text-overflow: ellipsis;

overflow: hidden !important;

color: #4a4a4a;

}

.SP\_PageEntryTitle\_Snippet {

height: 65px;

overflow: hidden !important;

}

.SP\_PageEntrySnippet {

display: -webkit-box;

-webkit-line-clamp: 2;

-webkit-box-orient: vertical;

font-size: 12px;

font-weight: 400;

color: #878787;

text-overflow: ellipsis;

overflow: hidden !important;

white-space: no-wrap;

max-height: 56px;

}

.SP\_PageEntrySnippet1 {

display: -webkit-box;

-webkit-line-clamp: 1;

-webkit-box-orient: vertical;

font-size: 12px;

font-weight: 400;

color: #878787;

text-overflow: ellipsis;

overflow: hidden !important;

white-space: no-wrap;

max-height: 56px;

}

.SP\_PageEntryUrl {

font-size: 12px;

font-weight: 500;

color: #4a4a4a;

text-overflow: ellipsis;

overflow: hidden !important;

white-space: nowrap;

padding-top: 8px;

}

.SP\_PageEntryUrl:hover {

color: #4083F1;

}

.SP\_ImageAnchor {

color: #C3C1C1;

text-decoration: none;

font-size: 10pt;

}

.SP\_ImageWrapper {

width: 80px;

height: 105px;

float: left;

overflow: hidden;

text-align: center;

}

.SP\_ImageNotFound {

text-align: center;

padding-top: 20px;

height: 105px;

width: 80px;

background-image: url('images/spacer\_no\_preview.png');

background-size: cover;

}

.SP\_ImageThumb {

width: 80px;

height: 105px;

}

.SP\_SimilarDiv {

clear: both;

border-top: 1px solid #dedede;

padding-top: 5px;

padding-left: 10px;

display: block;

width: 320px;

}

.SP\_Similar {

font-size: 12px;

float: left;

text-decoration: none;

color: #4083F1;

font-weight: 500;

}

.SP\_WebResult {margin: 8px 0 0 0;}

.SP\_WebUrl {color: green;}

.SP\_CloseButton:hover {

background: rgba(255,255,255,0.7) url('images/cross.png') no-repeat 50% 50%;

}

.SP\_CloseButton {

background: #f1f1f1 url('images/cross.png') no-repeat 50% 50%;

width: 44px;

height: 44px;

cursor: pointer;

position: absolute;

top: 00px;

right: 00px;

}

***Popup.js***

var h = function (a, b) {

function c() { }

c.prototype = b.prototype;

a.f = b.prototype;

a.prototype = new c;

a.e = function (a, c, g) {

return b.prototype[c].apply(a, Array.prototype.slice.call(arguments, 2))

}

};

var l = function () {

this.d = this.c = this.url = this.b = this.title = ""

},

m = function () {

this.a = [];

this.parseError = !1

},

n = function (a) {

l.call(this);

a.getElementsByTagName("CT\_URL").length && (this.b = a.getElementsByTagName("CT\_URL")[0].textContent);

a.getElementsByTagName("U").length && (this.url = a.getElementsByTagName("U")[0].textContent);

a.getElementsByTagName("T").length && (this.title = a.getElementsByTagName("T")[0].textContent);

a.getElementsByTagName("PREVIEW\_IMG").length && (this.c = a.getElementsByTagName("PREVIEW\_IMG")[0].textContent);

a.getElementsByTagName("S").length && (this.d = a.getElementsByTagName("S")[0].textContent.replace(/<br>/g, ""))

};

h(n, l);

var p = function (a) {

m.call(this);

if (a.firstChild && a.getElementsByTagName("GSP").length) {

if (a = a.getElementsByTagName("GSP")[0], a.getElementsByTagName("RES").length) {

a = a.getElementsByTagName("RES")[0].getElementsByTagName("R");

for (var b = 0; b < a.length; b++) {

var c = new n(a[b]);

"" != c.b && "" != c.url && this.a.push(c)

}

}

} else this.parseError = !0

};

h(p, m);

var r = function () {

var a = document.createElement("div");

a.className = "SP\_TopBar";

a.appendChild(document.createTextNode("Similar Pages"));

a.appendChild(q());

return a

},

q = function () {

var a = document.createElement("a");

a.className = "SP\_CloseButton";

a.id = "sp\_close\_button";

a.src = chrome.extension.getURL("images/cross.png");

a.addEventListener("click", function () {

window.close()

}, !1);

return a

},

s = function () {

var a = document.getElementById("sp\_main\_container");

a || (a = document.createElement("div"), a.id = "sp\_main\_container",

a.className = "SP\_MainContainer");

a.appendChild(r());

var b = document.createElement("div");

b.innerHTML = chrome.i18n.getMessage("sp\_zero\_results");

b.className = "SP\_Message";

a.appendChild(b);

document.body.appendChild(a)

},

t = function (a) {

var b = document.createElement("div");

b.className = "SP\_SpecialCaseContainer";

var c = r();

b.appendChild(c);

c = document.createElement("div");

c.innerHTML = a;

c.className = "SP\_Message";

c.style["text-align"] = "justify";

b.appendChild(c);

document.body.appendChild(b)

},

w = function (a, b) {

var c = document.createElement("div");

c.id = "page\_" + b;

c.className = " SP\_PageEntry ";

var e = u(a.b + "&output=br"),

f = document.createElement("a");

f.className = "SP\_ImageAnchor";

f.addEventListener("click", function () {

window.close()

}, !1);

var g = document.createElement("div");

g.className = "SP\_ImageWrapper";

var d = document.createElement("img");

d.id = "img\_" + b;

d.className = "SP\_ImageThumb";

v(d);

d.src = a.c;

g.appendChild(d);

f.appendChild(g);

c.appendChild(f);

f = document.createElement("div");

f.className = "SP\_PageEntryText";

g = document.createElement("div");

g.className =

"SP\_PageEntryTitle\_Snippet";

d = document.createElement("div");

d.className = "SP\_PageEntryTitle";

var k = document.createElement("a");

k.addEventListener("click", function () {

window.close()

}, !1);

k.innerHTML = a.title;

k.title = a.title.replace("<b>", "").replace("</b>", "");

d.appendChild(k);

g.appendChild(d);

d = document.createElement("span");

d.className = 25 < a.title.length ? "SP\_PageEntrySnippet1" : "SP\_PageEntrySnippet";

d.innerHTML = a.d;

g.appendChild(d);

f.appendChild(g);

g = document.createElement("a");

g.href = a.url;

d = document.createElement("div");

d.className = "SP\_PageEntryUrl";

d.textContent = g.hostname;

f.appendChild(d);

c.appendChild(f);

c.onclick = function () {

chrome.tabs.create({

url: e

})

};

return c

},

u = function (a) {

return chrome.i18n.getMessage("sp\_redirect\_host") + a

},

v = function (a) {

var b = a.id;

a.addEventListener("error", function () {

var a = document.getElementById(b),

e = a.parentNode;

e.className += " SP\_ImageNotFound ";

e.removeChild(a);

a = document.createElement("span");

a.innerHTML = chrome.i18n.getMessage("sp\_preview\_not\_available");

e.appendChild(a)

}, !1)

};

var y = function (a) {

var b = x,

c = new XMLHttpRequest;

c.open("GET", a, !1);

c.onreadystatechange = function () {

4 == c.readyState && 200 == c.status && b(c.responseXML)

};

try {

c.send()

} catch (e) { }

},

z = function (a, b) {

var c = a.id;

chrome.tabs.getAllInWindow(null, function (e) {

for (var f = 1; f < e.length; ++f)

if (e[f].id == c) {

b(e[f - 1]);

return

}

b(a)

})

};

var A = /(chrome|file|http?:\/\/chrome\.google\.com\/extenions)/,

B = /^chrome-extension:\/\/[a-z]+\/popup\.html$/,

x = function (a) {

var b = new p(a);

if (b.parseError) a = document.getElementById("sp\_main\_container"), a || (a = document.createElement("div"), a.id = "sp\_main\_container", a.className = "SP\_MainContainer"), a.appendChild(r()), b = document.createElement("div"), b.innerHTML = chrome.i18n.getMessage("sp\_error"), b.className = "SP\_Message", a.appendChild(b), document.body.appendChild(a);

else if (b.a.length) {

a = document.getElementById("sp\_main\_container");

a || (a = document.createElement("div"), a.id = "sp\_main\_container", a.className = "SP\_MainContainer");

var c = r();

a.appendChild(c);

c = document.createElement("div");

c.id = "sp\_results";

a.appendChild(c);

document.body.appendChild(a);

document.addEventListener("keydown", C, !1);

if (b.a.length) {

a = document.getElementById("sp\_results");

for (var b = b.a, c = Math.min(b.length, 20), e = 0; e < c; e++) {

var f = w(b[e], e);

a.appendChild(f)

}

}

} else s()

},

C = function (a) {

27 == a.keyCode && window.close()

},

D = function (a, b) {

t(chrome.i18n.getMessage("sp\_tld\_special\_case\_text")) : b && y(b + "search?output=br&q=related:" + encodeURIComponent(a)) : s()

};

});

***Option.html***

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>

<title>Google Similar Page</title>

<meta http-equiv="Content-Type" content="text/html; charset=us-ascii" />

<link href="http://www.google.com/tools/dlpage/res/c/css/goog.css"

rel="stylesheet" type="text/css" />

<link href="http://www.google.com/tools/dlpage/res/c/css/dlpage.css"

rel="stylesheet" type="text/css" />

<meta name="author" content="Saurav Jaiswal">

<meta property="og:url" content="https://github.com/sauravjaiswalsj/deepcrawler">

<meta property="og:type" content="profile" />

<meta name="keywords" content="DeepCrawler,web search, deep search,google search,developer,coder,programmer,computer science,engineering,tv,series,blog" />

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link href="https://fonts.googleapis.com/css?family=Exo+2" rel="stylesheet">

<meta property="og:image" content="#" />

<meta property="og:image:width" content="200">

<meta property="og:image:height" content="200">

<meta property="og:description" content="This is a chrome extension which google search and gives the result as popup;, published on chrome store." />

<link href="options.css" rel="stylesheet" type="text/css" />

<script src="options\_main.js" type="text/javascript"></script>

</head>

<body>

<div id="container">

<div id="main">

<img class="logo" src="images/sp-48.png" alt="logo"/>

<div class="parallex">

</div>

<div>

<h2 id="title"></h2>

<p id="options\_intro"></p>

<p id="options\_legal"></p>

<p id="options\_feedback"></p>

<h3 id="options\_uninstall"></h3>

<p id="options\_uninstall\_message"></p>

<div id="footer">

<span id="options\_copy"></span>

<span id="options\_about\_google"></span> -

<span id="options\_privacy\_policy"></span> -

<span id="options\_tos"></span>

</div>

</div>

</div>

</div>

</body>

</html>

***Option.css***

/\* Additional CSS for About page \*/

h1, h2, h3, h4 {

padding: 0 0 .2em;

}

input[type=checkbox] {

margin: 0;

}

p {

text-align: left;

}

#container {

width: 800px;

}

#header {

height: 40px;

}

#main {

padding: 0.2em 0 0 60px;

}

#title {

padding: 1em 0 0.5em;

}

#close {

clear: both;

margin-top: 1em;

text-align: left;

}

.logo {

margin: 16px 0 0 -60px;

}

.screenshot {

float: right;

padding-left: 1em;

}

**Option.js**

var c = function (b) {

var a = new XMLHttpRequest;

a.open("GET", "http://www.google.com/ncr", !1);

a.onreadystatechange = function () {

4 == a.readyState && 200 == a.status && b(a.responseText)

};

try {

a.send()

} catch (f) { }

};

function d(b, a) {

document.getElementById(b).innerHTML = chrome.i18n.getMessage("sp\_" + b, a)

}

function e() {

var b = window.localStorage.request\_domain;

if (b) {

var a = document.getElementById("domain\_select\_block");

a.innerHTML = chrome.i18n.getMessage("sp\_domain\_message", b.replace("http://", "").replace("/", ""));

"http://www.google.com/" != b && (a.innerHTML += chrome.i18n.getMessage("sp\_domain\_change\_message"));

a.innerHTML += "<p>"

} (b = document.getElementById("over")) && b.addEventListener("click", function () {

window.localStorage.request\_domain = "http://www.google.com/";

c(function () {

var a = document.getElementById("domain\_select\_block");

a.innerHTML = chrome.i18n.getMessage("sp\_domain\_clicked\_message");

a.innerHTML += "<p>"

})

}, !1)

}

function \_writeMessage(b, a) {

document.write(chrome.i18n.getMessage(b, a))

}

document.addEventListener("DOMContentLoaded", function () {

d("title");

d("options\_intro");

d("options\_legal", chrome.i18n.getMessage("locale"));

d("options\_feedback");

d("options\_uninstall");

d("options\_uninstall\_message");

d("options\_copy");

d("options\_about\_google", chrome.i18n.getMessage("locale"));

d("options\_privacy\_policy", chrome.i18n.getMessage("locale"));

d("options\_tos", chrome.i18n.getMessage("locale"));

e()

});

$(function () {

$(":file").change(function () {

if (this.files && this.files[0]) {

var reader = new FileReader();

reader.onload = imageIsLoaded;

reader.readAsDataURL(this.files[0]);

}

});

});

function imageIsLoaded(e) {

$('images/img\_parallax.jpg').attr('src', e.target.result);

$('images/img\_parallax.jpg').attr('src', e.target.result);

};

***Background.html***

<html>

<head>

<script src="background\_main.js"></script></head>

</html>

***Background\_main.js***

var e = function () {

var a = d,

b = new XMLHttpRequest;

b.open("GET", "https://clients1.google.com/searchdomaincheck?format=domain", !1);

b.onreadystatechange = function () {

4 == b.readyState && 200 == b.status && a(b.responseText)

};

try {

b.send()

} catch (c) { }

};

function d(a) {

|  |
| --- |
|  |
| arget: function(b) { |
|  | var c = a.location && a.location.hash; |
|  | return c && c.slice(1) === b.id |
|  | }, |
|  | root: function(a) { |
|  | return a === o |
|  | }, |
|  | focus: function(a) { |
|  | return a === n.activeElement && (!n.hasFocus || n.hasFocus()) && !!(a.type || a.href || ~a.tabIndex) |
|  | }, |
|  | enabled: oa(!1), |
|  | disabled: oa(!0), |
|  | checked: function(a) { |
|  | var b = a.nodeName.toLowerCase(); |
|  | return "input" === b && !!a.checked || "option" === b && !!a.selected |
|  | }, |
|  | selected: function(a) { |
|  | return a.parentNode && a.parentNode.selectedIndex, a.selected === !0 |
|  | }, |
|  | empty: function(a) { |
|  | for (a = a.firstChild; a; a = a.nextSibling) |
|  | if (a.nodeType < 6) return !1; |
|  | return !0 |
|  | }, |
|  | parent: function(a) { |
|  | return !d.pseudos.empty(a) |
|  | }, |
|  | header: function(a) { |
|  | return X.test(a.nodeName) |
|  | }, |
|  | input: function(a) { |
|  | return W.test(a.nodeName) |
|  | }, |
|  | button: function(a) { |
|  | var b = a.nodeName.toLowerCase(); |
|  | return "input" === b && "button" === a.type || "button" === b |
|  | }, |
|  | text: function(a) { |
|  | var b; |
|  | return "input" === a.nodeName.toLowerCase() && "text" === a.type && (null == (b = a.getAttribute("type")) || "text" === b.toLowerCase()) |
|  | }, |
|  | first: pa(function() { |
|  | return [0] |
|  | }), |
|  | last: pa(function(a, b) { |
|  | return [b - 1] |
|  | }), |
|  | eq: pa(function(a, b, c) { |
|  | return [c < 0 ? c + b : c] |
|  | }), |
|  | even: pa(function(a, b) { |
|  | for (var c = 0; c < b; c += 2) a.push(c); |
|  | return a |
|  | }), |
|  | odd: pa(function(a, b) { |
|  | for (var c = 1; c < b; c += 2) a.push(c); |
|  | return a |
|  | }), |
|  | lt: pa(function(a, b, c) { |
|  | for (var d = c < 0 ? c + b : c; --d >= 0;) a.push(d); |
|  | return a |
|  | }), |
|  | gt: pa(function(a, b, c) { |
|  | for (var d = c < 0 ? c + b : c; ++d < b;) a.push(d); |
|  | return a |
|  | }) |
|  | } |
|  | }, d.pseudos.nth = d.pseudos.eq; |
|  | for (b in { |
|  | radio: !0, |
|  | checkbox: !0, |
|  | file: !0, |
|  | password: !0, |
|  | image: !0 |
|  | }) d.pseudos[b] = ma(b); |
|  | for (b in { |
|  | submit: !0, |
|  | reset: !0 |
|  | }) d.pseudos[b] = na(b); |
|  |  |
|  | function ra() {} |
|  | ra.prototype = d.filters = d.pseudos, d.setFilters = new ra, g = ga.tokenize = function(a, b) { |
|  | var c, e, f, g, h, i, j, k = z[a + " "]; |
|  | if (k) return b ? 0 : k.slice(0); |
|  | h = a, i = [], j = d.preFilter; |
|  | while (h) { |
|  | c && !(e = Q.exec(h)) || (e && (h = h.slice(e[0].length) || h), i.push(f = [])), c = !1, (e = R.exec(h)) && (c = e.shift(), f.push({ |
|  | value: c, |
|  | type: e[0].replace(P, " ") |
|  | }), h = h.slice(c.length)); |
|  | for (g in d.filter) !(e = V[g].exec(h)) || j[g] && !(e = j[g](e)) || (c = e.shift(), f.push({ |
|  | value: c, |
|  | type: g, |
|  | matches: e |
|  | }), h = h.slice(c.length)); |
|  | if (!c) break |
|  | } |
|  | return b ? h.length : h ? ga.error(a) : z(a, i).slice(0) |
|  | }}; |
|  | } |
|  |  |

**Chapter 4**

**REFERENCES and APPENDIX**

<https://en.wikipedia.org/wiki/Browser_extension>

<https://developer.chrome.com/extensions>

<https://chrome.google.com/webstore/category/extensions?hl=en-US>

<https://developer.chrome.com/extensions>

<https://chrome.google.com/webstore/category/extensions?hl=en-US>