**Top1Today: Chrome Extension**

**Architectural Mechanisms**

**Security:**

The content script accesses websites for their activity(cursor/media playback), and does not use any other info, thereby securing the user sensitive data.

**Accuracy:**

The background scripts using Chrome API's qualifies only cursror tasks and media playback on each tab of a specific website as webpage activity and considers any other idle time on website etc

as invalid and is not calculated in background script's Top1 Today time spent calculator.

**Performance:**

Dynamic calculation of the Top1 website with most time spent is done in the background script's time spent calculator in real time, thereby reducing any lag/delay

in displaying the extension results.

**Availability:**

The Top1 Today extension present near the address bar shows the extension name on hover. And on clicking a pop-up is displayed with the Top 5 websites in order of most

time spent. The pop-up should be displayed at all times with the accurate results and this will be achieved from reliable communication between background script's time spent calculator and the pop-up/UI.

**Marchitecture: Top1Today**

Diagram

Description automatically generated

**Commentary**

Top1today becomes available near the search bar of the browser once the user adds it to is extensions from Chrome store.

The marchitecture above explains the components Top1today interacts with, to accurately show the final top 5 websites that were longest most actively used until then on that day.

The UI of our extension comprises of popup.html and popup.js. A Manifest file gives defines the extension so the browser understands them. A **pop-up page(popup.html) is displayed as soon as the user clicks on the extension.**

The **background script is the event handler** that listens for the user to perform any of the actions on a given tab and fires when one of these actions are performed. Background script listens for **events through Chrome API**. Events like **hover, click, scroll, play-back video** are some of the examples of **defining an active tab**.

Once user performs any of these activities, the **event handler starts tracking the time for that particular website utilizing Chrome API**. (We could also create a separate js file to perform time calculation and return that to popup.js, TBD)

When any **new tab is opened, content script tracks the URL and pass the information to background script via messages**. Background communicates this information to popup script (which might interact with another script which calculates overall time for every URL) and displays the final content on the extension.

**Chrome storage API** can be utilized to save important data when and if the extension crashes.

The extension communicates with browser through Chrome API by tracking all active websites and displays the result dynamically. Using a framework can also be considered but is TBD for now.

**Risks**

**Resource Availability :**

The Top1 Today Project team consists of only 2 dedicated software developers and no test engineer, with the time spent calculator to be built from scratch,

extensive tests are required for the reliable and accurate time spent calculation qualification.

**Mitigation:** Dedicated time for developer testing is planned.

**Complexity of Requirement:**

The software team being new to the complex chrome extensions development might have understanding issues with the working of each module and thereby risking implementation of these modules.

**Mitigation:** Development team to have Regular Technical training on Chrome extensions.

**Change Management:**

Expected security policy changes on Chrome API's in 2020.With this being the first version of Top1, security changes in chrome extensions will need to be taken care of.

**Mitigation:** A Change Control Review is formed consisting of Top1 developers to monitor and plan for all changes in environment during the project.