PROJECT 04
“CHROME INSPECT”

SUBMITTED BY:

MANASWI M. PATIL

OBJECTIVES:

The primary aim of this project is to provide interns with comprehensive knowledge and hands-on experience in using Google Chrome's Inspect feature, part of Chrome Developer Tools. The project is designed to help interns master web development debugging, performance analysis, and website inspection techniques, essential for efficient web development and troubleshooting.

INDEX

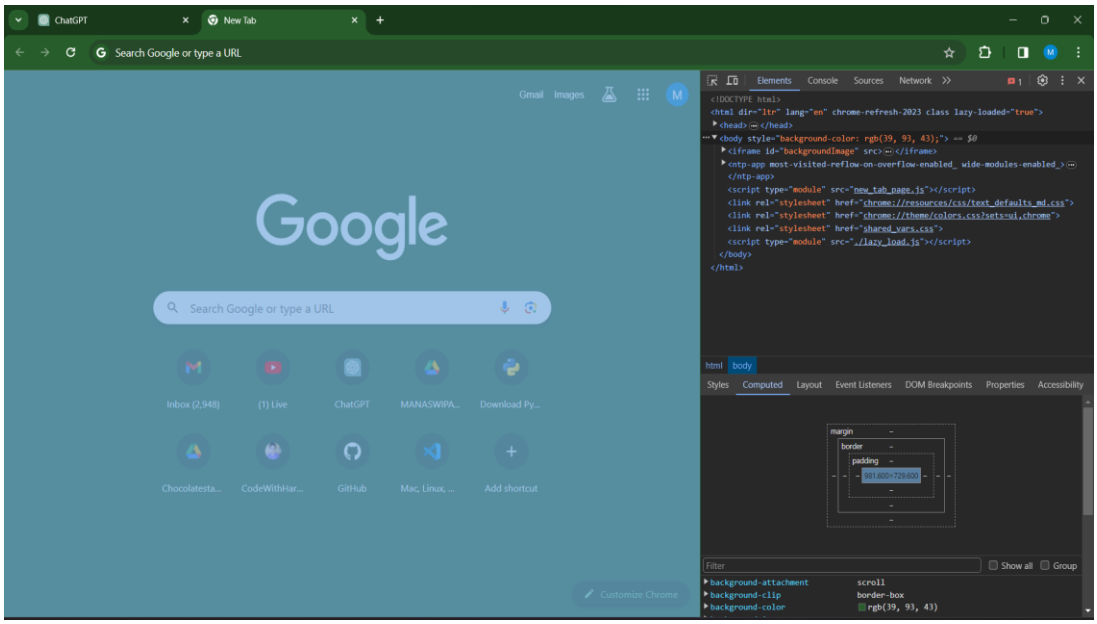
Submitted by: Manaswi M. Patil

1. Introduction to Chrome's Inspect Feature

I. What is exactly Chrome Developer Tools?

Chrome Developer Tools "DevTools" is a set of web authoring and debugging tools built into the Google Chrome browser. It allows developers to inspect, edit, debug, and monitor web pages or applications in real-time. DevTools is a powerful resource for web developers to analyse and optimize their code, as well as troubleshoot issues.

This is the window of developer tools

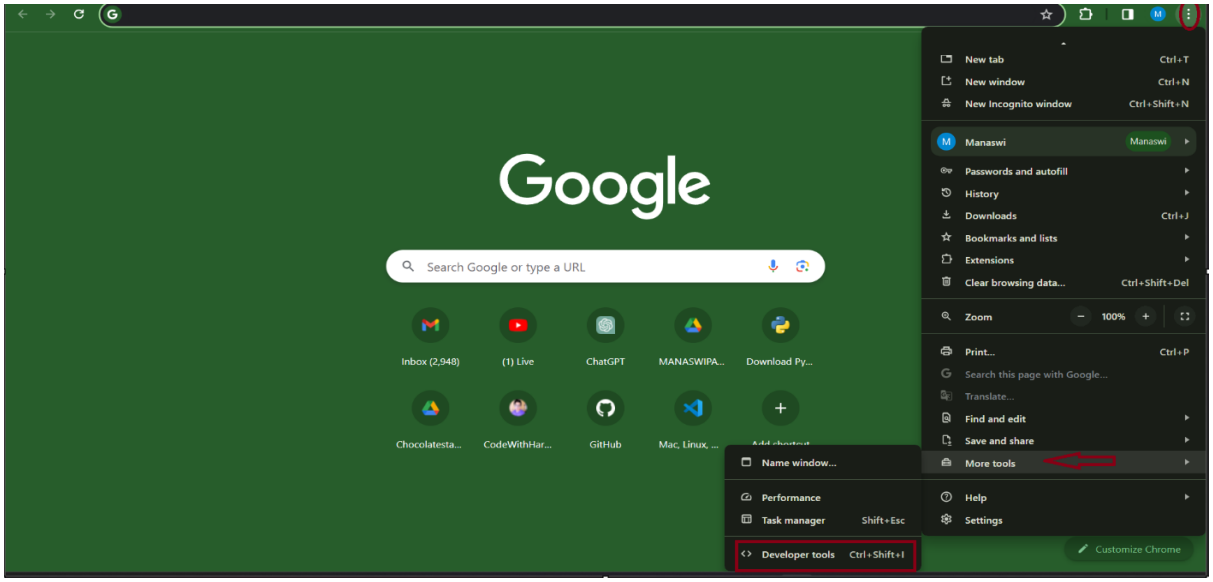


II. Basics of accessing and navigating the Inspect interface.

A. Basics of accessing Inspect Interface

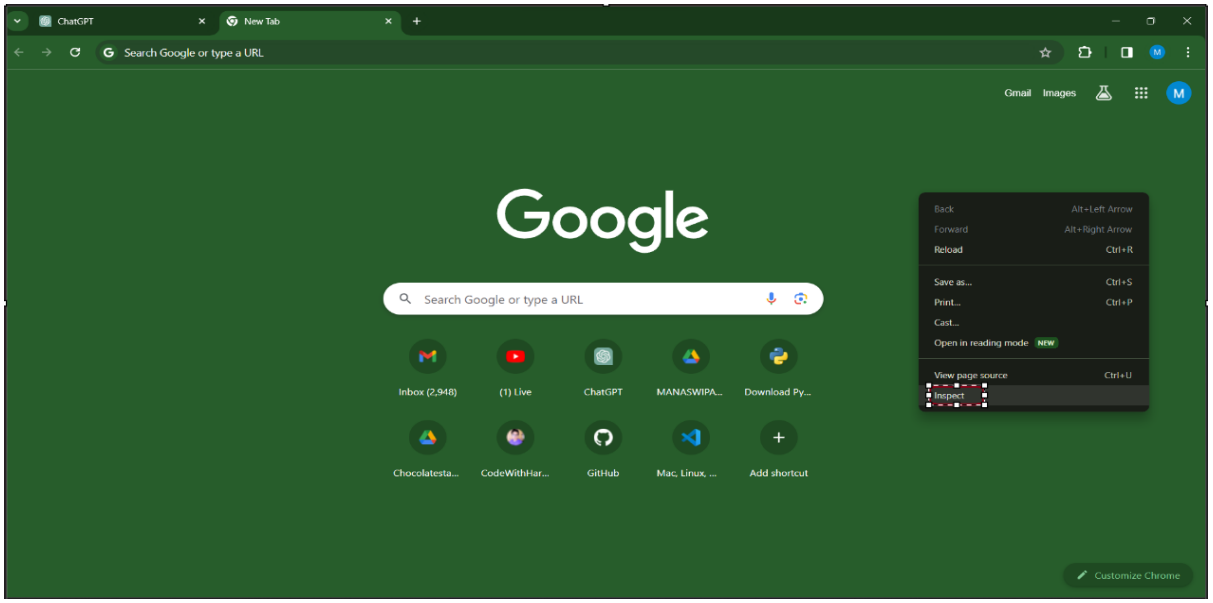
- Open Chrome Browser →to Open DevTools→**"Right-click"on** any element on the page you want to inspect→Select **"Inspect"** from the context menu→or you can use the keyboard shortcut→windows linux: **Ctrl + Shift + I or F12**→Mac: **Cmd + Opt + I.**
- From chrome menu→open chrome browser→go to right corner of page→click **'more tools'→developers tools→**click.

Submitted by: Manaswi M. Patil



B. How to Navigating the Inspect interface?

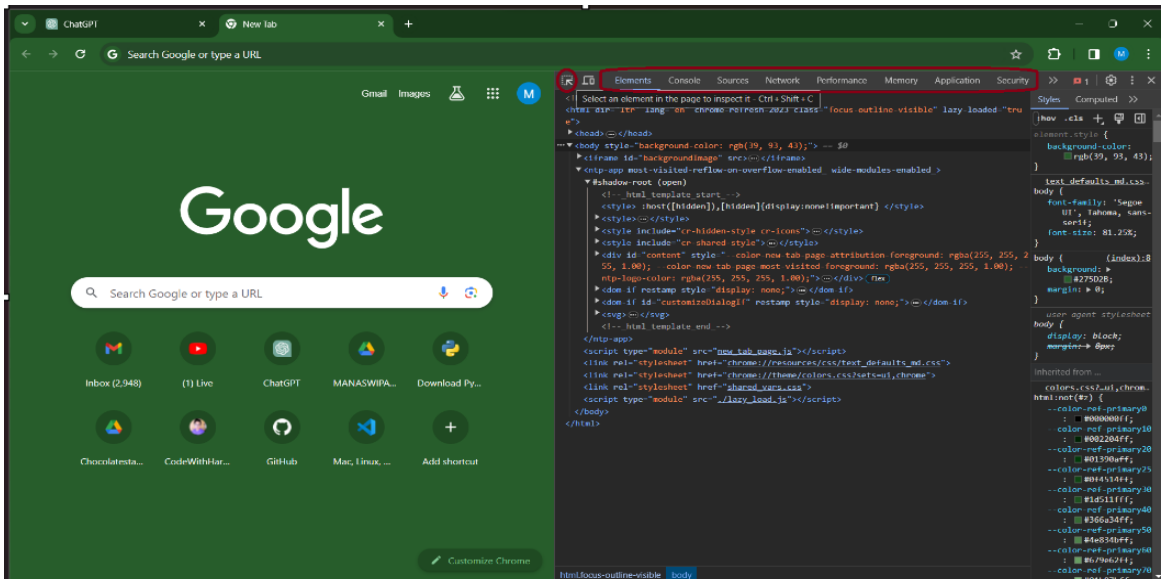
- On your chrome page →right click →click "**inspect**".



Submitted by: Manaswi M. Patil

2. PANELS IN DEVELOPER TOOLS

- Open chrome browser→click on **3 dot** of right top corner of page→'more tools'→DevTools →click→or click shortcut keys as **'ctrl+shift+I".**

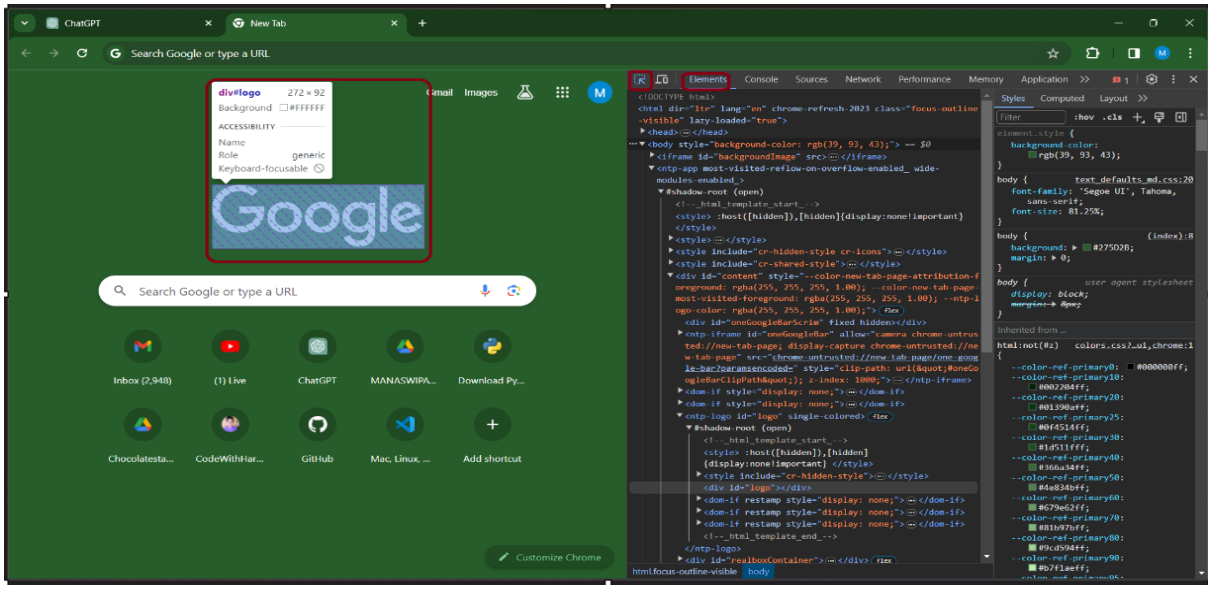


A. Elements Panel

The Elements panel in browser Developer Tools is a powerful tool for inspecting and manipulating the HTML and CSS of a webpage. It allows developers to view and modify the structure and styling of a page in real-time.

Select this element to inspect it.

- Google symbol is highlighted after clicking on specific area.



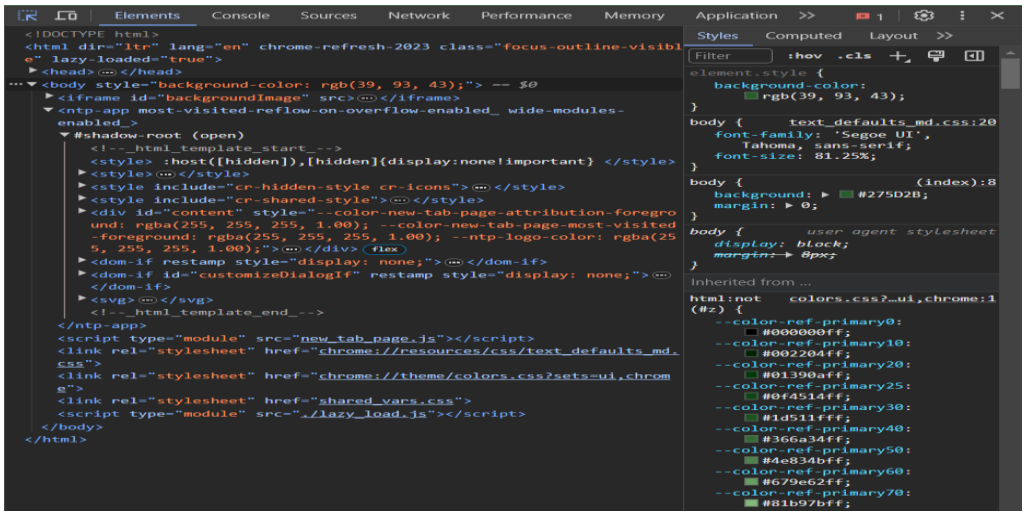
Submitted by: Manaswi M. Patil

Key features of elements panel:

i) HTML Tree:

The main area of the Elements panel displays the hierarchical structure of the HTML document. Each HTML element is represented as a node in a tree structure. You can expand and collapse nodes to navigate through the HTML hierarchy.

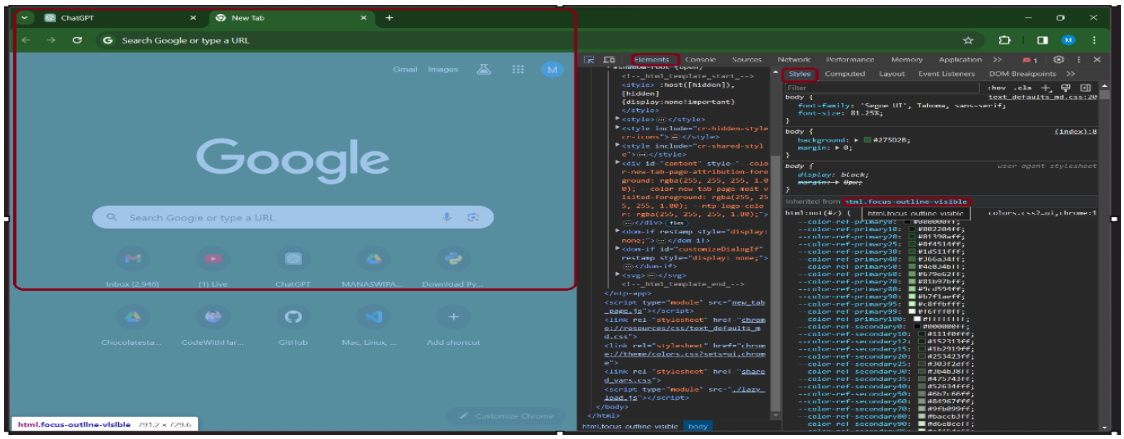
- Click on main element panel→displays the html tree structure like below.



ii) Styles and Computed Styles:

The right sidebar of the Elements panel shows the styles applied to the selected HTML element. This includes styles from inline styles, stylesheets, and user agent styles. The Computed tab provides a complete set of styles applied to the element, including styles inherited from parent elements.

- Click on 'Elements panel'→click sidebar→click on **styles**.

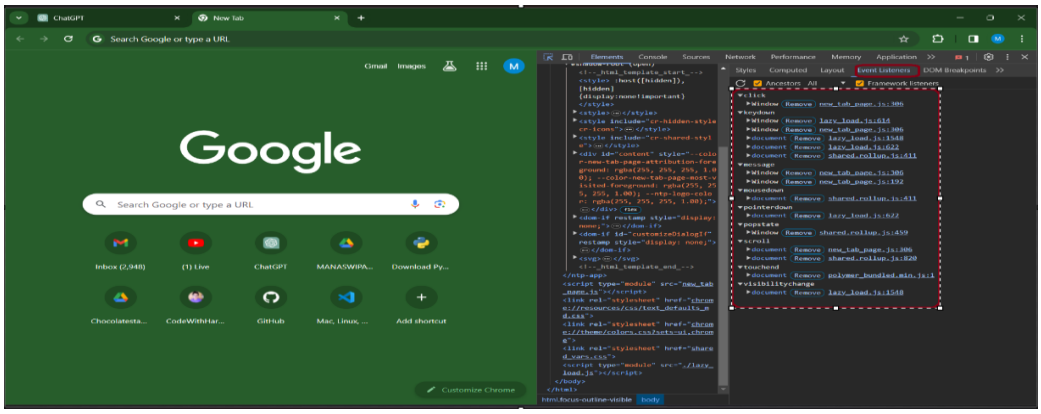


iii) Event Listeners:

The "Event Listeners" tab in the right sidebar allows you to view and manage event listeners attached to the selected element. It provides information about the type of events and the functions that handle them.

Submitted by: Manaswi M. Patil

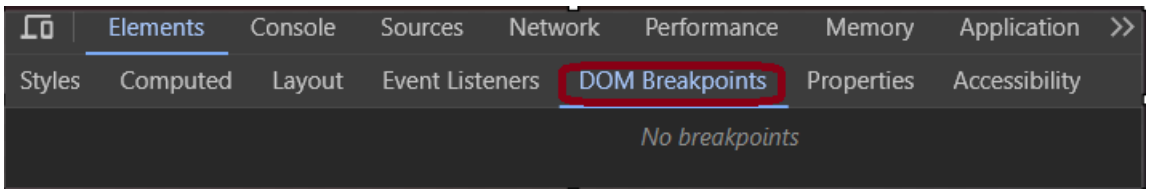
- Click on ‘Elements panel’→click sidebar→click ‘**Event Listner**’.



iv) DOM Breakpoints:

The "DOM Breakpoints" section allows you to set breakpoints on specific DOM events, such as **subtree modifications**, **attribute modifications, and node removal**. This can be useful for debugging when certain changes occur in the DOM.

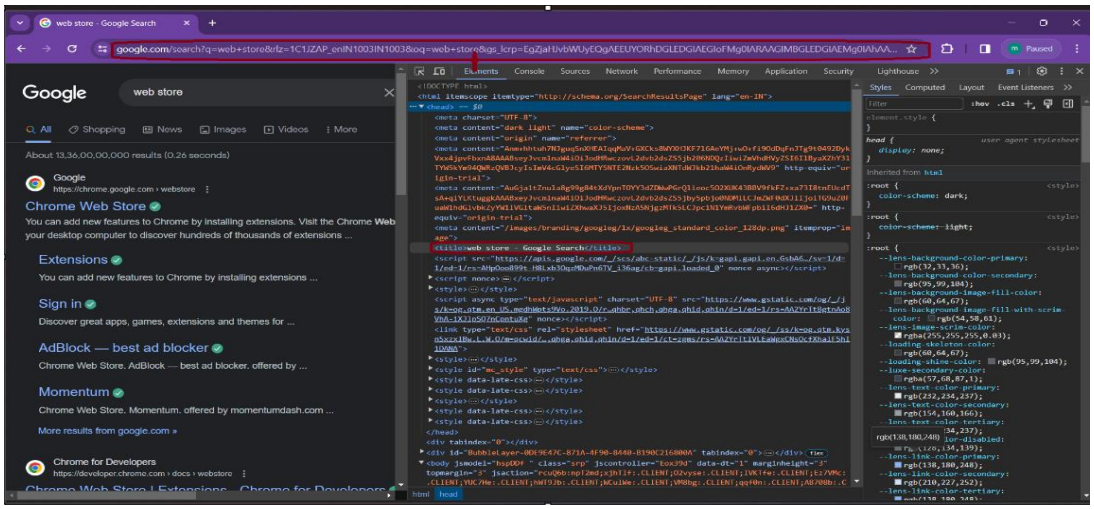
- Click on ‘Elements panel’→click sidebar→**click ‘breakpoints’.**



v) Search and Filter:

You can search for elements using the search bar at the top of the Elements panel. It helps in quickly locating specific elements on the page. Filter options are available to show only visible elements, only elements with event listeners.

- Click on ‘Elements panel’→click top of element panel.

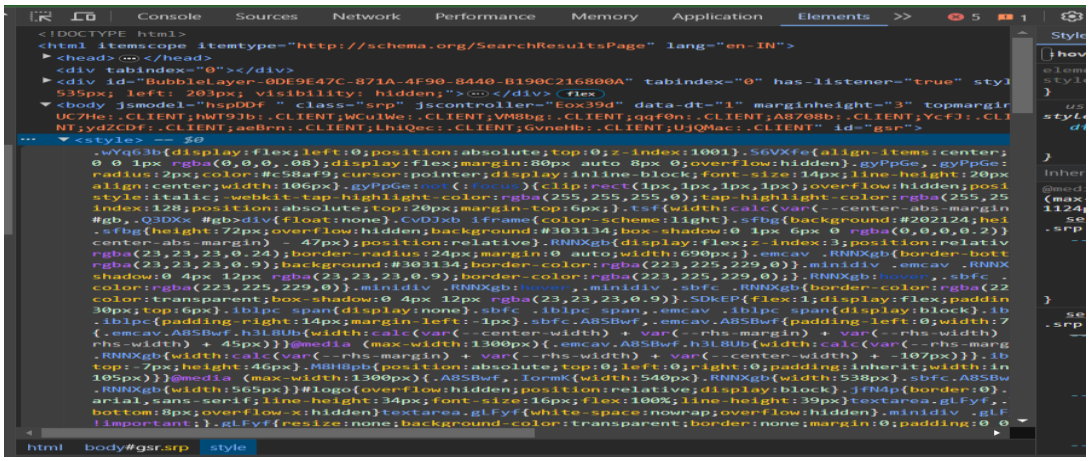


Submitted by: Manaswi M. Patil

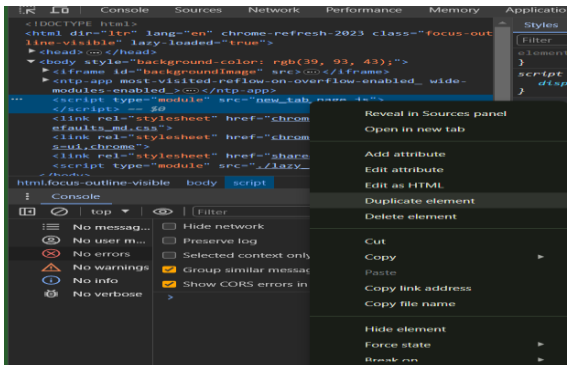
vi) Editing and Live Editing:

You can directly edit HTML content or CSS styles within the Elements panel. Changes made here are reflected live on the webpage.

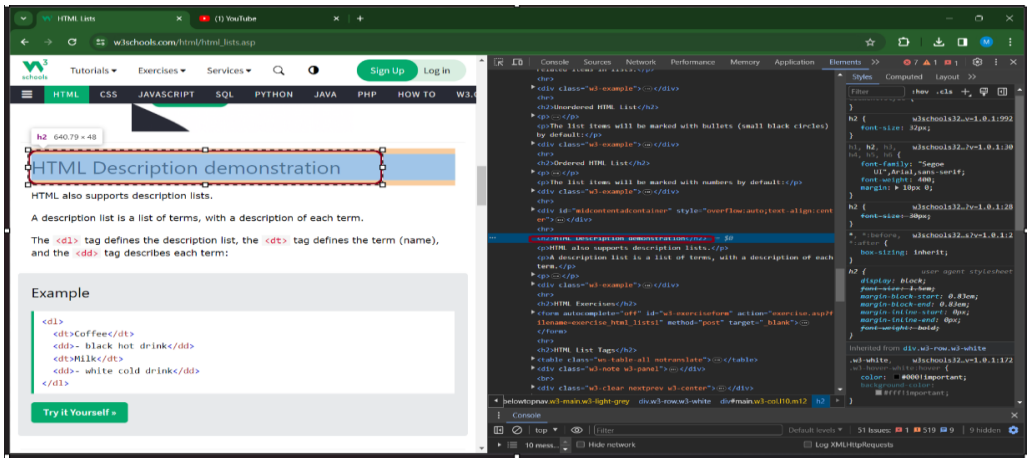
- Press **left** >arrow to list collapses.
- **Down** arrow to go down.
- **Up** arrow press to go up.



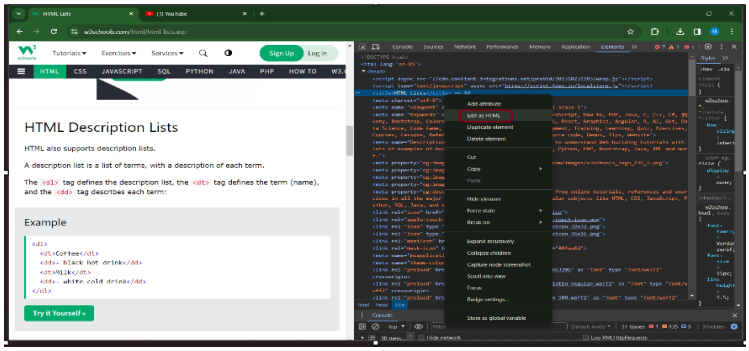
➢ Right-clicking on an element provides window→see options to copy, delete, or add attributes.



➢ Right click on text which want to edited to you→click option on edited as html→inspect the element where u want to change.



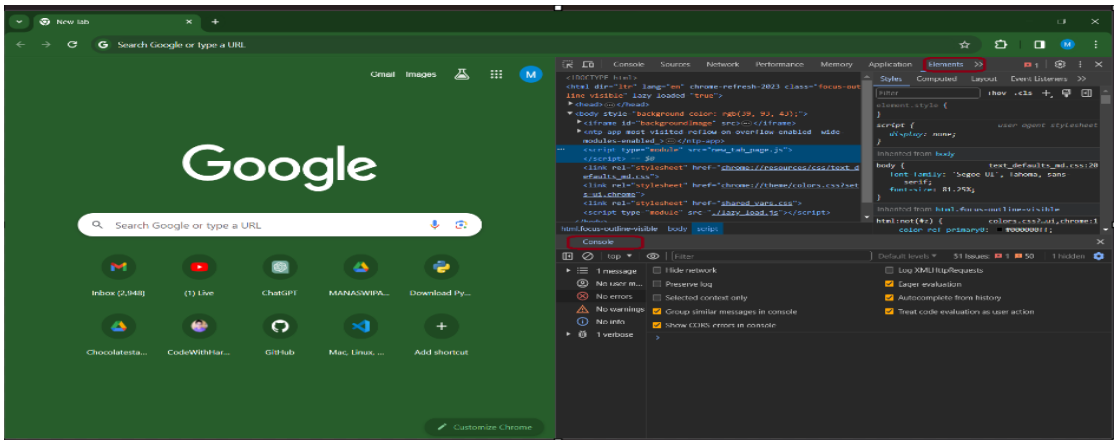
Submitted by: Manaswi M. Patil



vii) Console Interaction:

The console at the bottom of the Elements panel allows you to interact with the document using JavaScript. You can execute scripts and see the results directly.

- Click→elements→go bottom of element panel→click **console.**



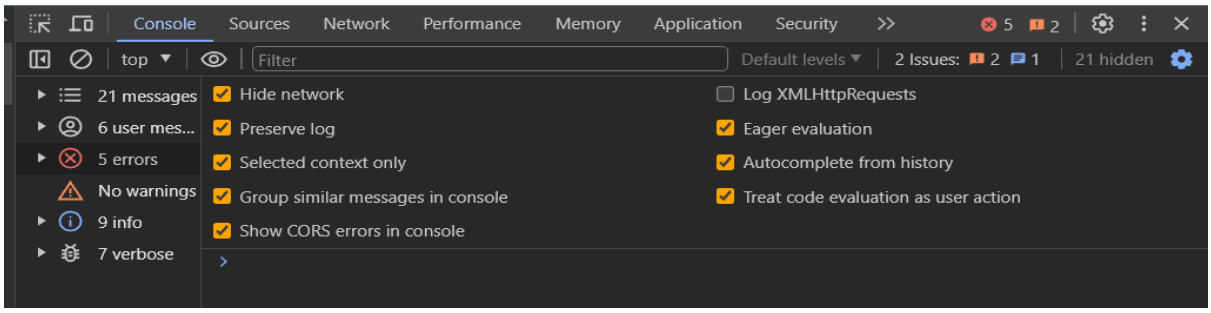
How to change background color ?

- Change background colour to text→click on element to inspect→in html editor right click→select as html to make changes→and make changes or add lines to change color of text background.

B. Console Panel

The Console panel allows you to interact with the JavaScript code on the page. You can execute JavaScript commands and see console logs, errors, and warnings.

- Click dev tools setting→click console panel.



Features of console panel

A. Console for debugging

It offers various methods that can significantly aid in identifying and fixing issues in your JavaScript code. Here are some ways you can use the console for debugging:

i. Logging message:

Logging messages to the console panel in various programming languages or environments is a common practice for debugging and monitoring code.

a) **Using console.log ():** Logs a simple message

```
> console.log('hello people')
hello people                                    VM1823:1
```

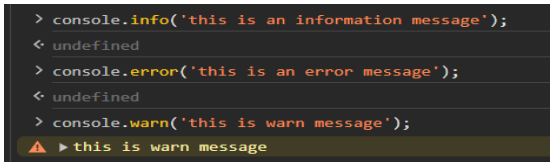
b) **Logging Variables:** Logs a variable value

```
> num =30;
< 30
> console.log('the value of num is:',num);
the value of num is: 30
```

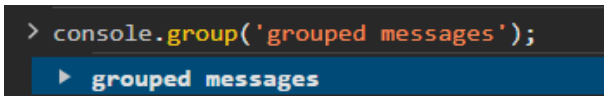
c) **Logging Objects**: Logs an object

```
> const object = {
      name: 'toy',
      city: 'mumbai' };
      console.log('object:',object);
object: ▶ {name: 'toy', city: 'mumbai'}
```

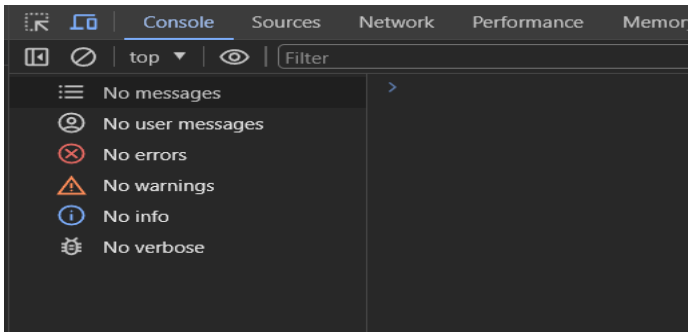
d) **Info, Warning, Error**



e) Grouping Messages



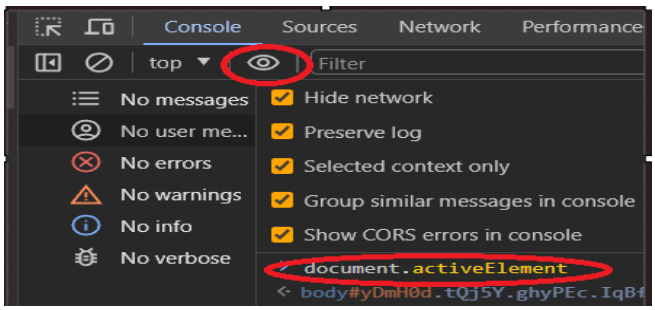
f. Filtering messages:



B. Running JavaScript

i. Create a live expression

➢ Click Create Live Expression symbol →The Live Expression text box appears→Type your expression in the text box→you can use a live expression to track element focus.



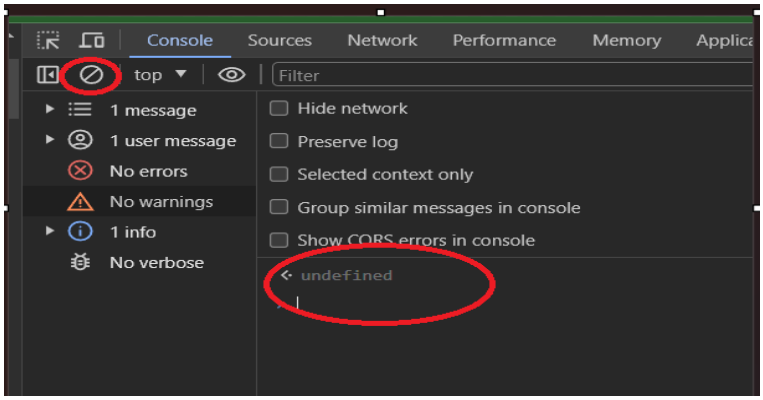
Adding multiple expression using live expression icon.

To remove expression used close button .

Submitted by: Manaswi M. Patil

ii. Clearing Console

- Type ‘console.clear()’ in console panel → to clear console → Or click on the clear Console icon.



C. Source Panel

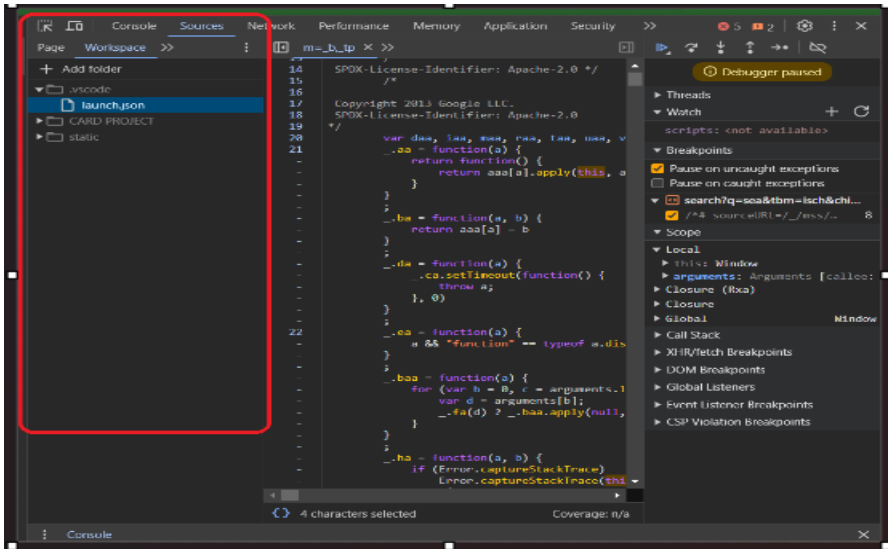
The Sources panel is used for debugging JavaScript code. You can set breakpoints, step through code, and inspect variables. Source panel has 3 window as following .

- Click devtool setting→click sources panel.

Features of source panel

❖ File navigator pane

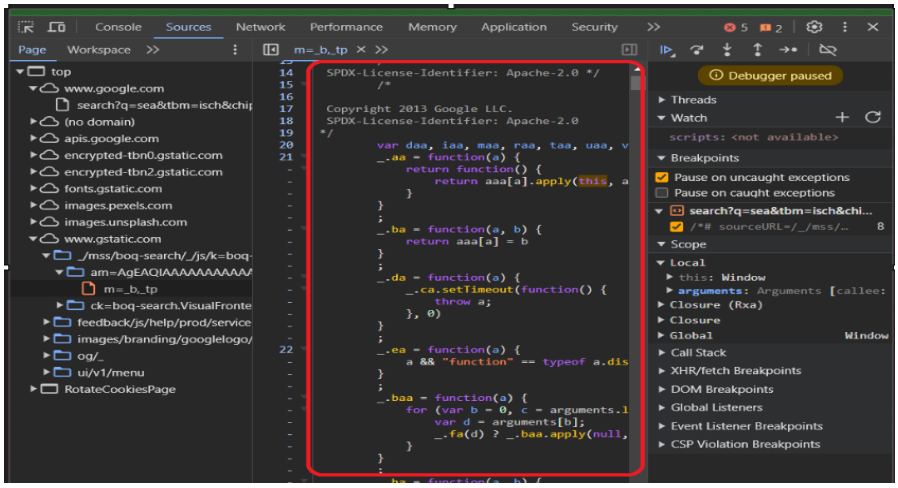
Open file or added file view in this pane.-->Using add folder you can any folder you want.



❖ Code navigator pane

Code of open particular file display here.

Submitted by: Manaswi M. Patil



❖ JavaScripts debugging Pane

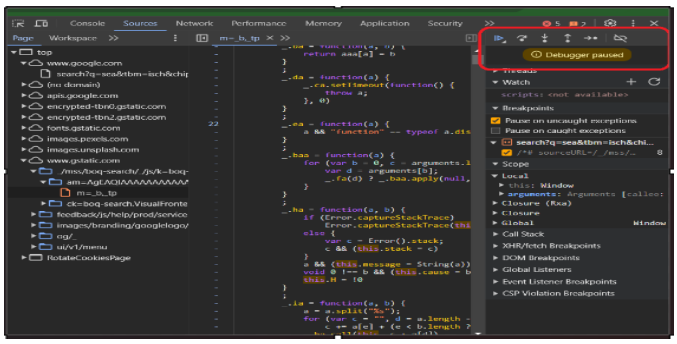
The JavaScript Debugging pane: Various tools for inspecting the page's JavaScript. If your DevTools window is wide, this pane is displayed to the right of the Code Editor pane.

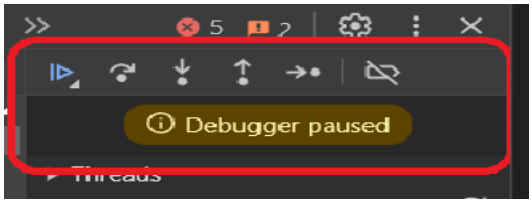
Breakpoints

Place breakpoints in your code to pause execution at specific points. This allows you to inspect variables and step through code.

Setting a Breakpoint

- Open your JavaScript file in the Code Editor pane of "Sources" panel→Find the line number where you want to place the breakpoint→Click on the line number to set a breakpoint.





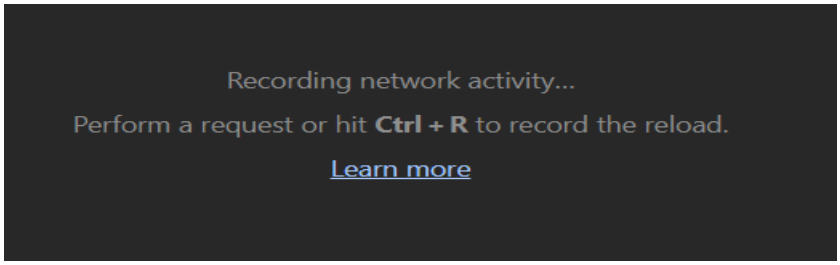
- Resume execution
- Step over
- Step into
- Step out

Submitted by: Manaswi M. Patil

D. Network Panel

The Network panel shows the network activity of the page, including HTTP requests and responses. Useful for analysing loading times and diagnosing network-related issues. It captures and displays information about each network request made by the webpage, including resources like HTML, CSS, JavaScript files, images, API requests, and more.

- Click→dev tools settings→click ‘network panel’.



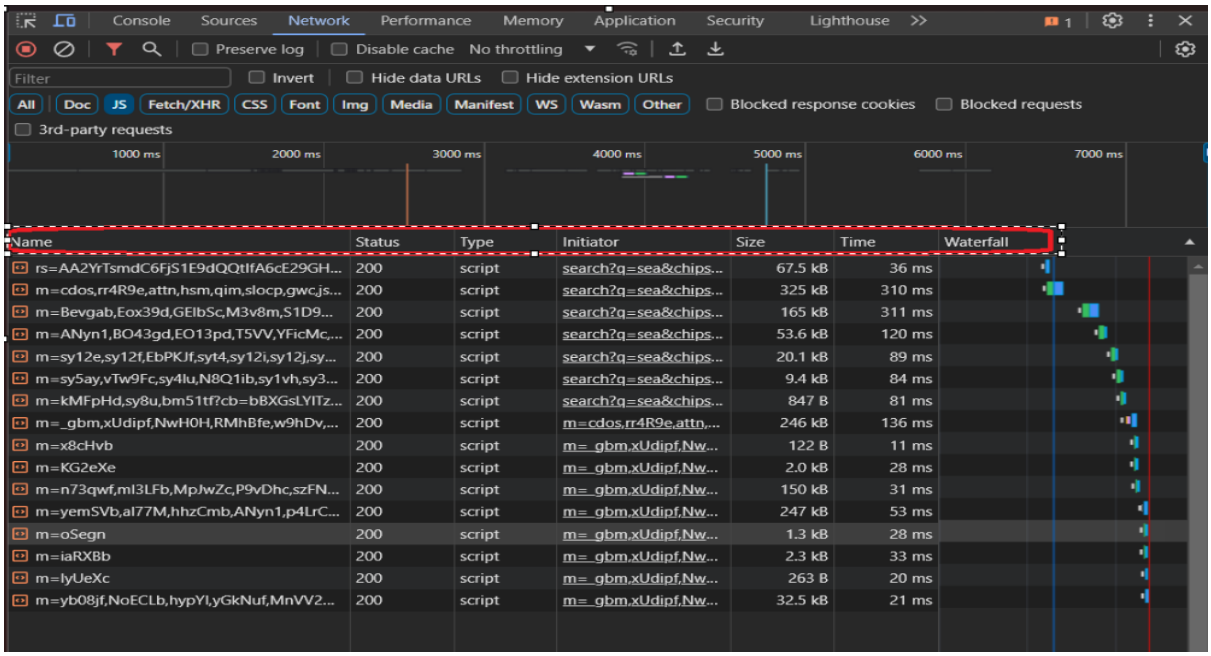
Right now the Network panel is empty. That's because DevTools only logs network activity while it's open and no network activity has occurred since you opened DevTools.

Features of Network Panel

A. Network log:

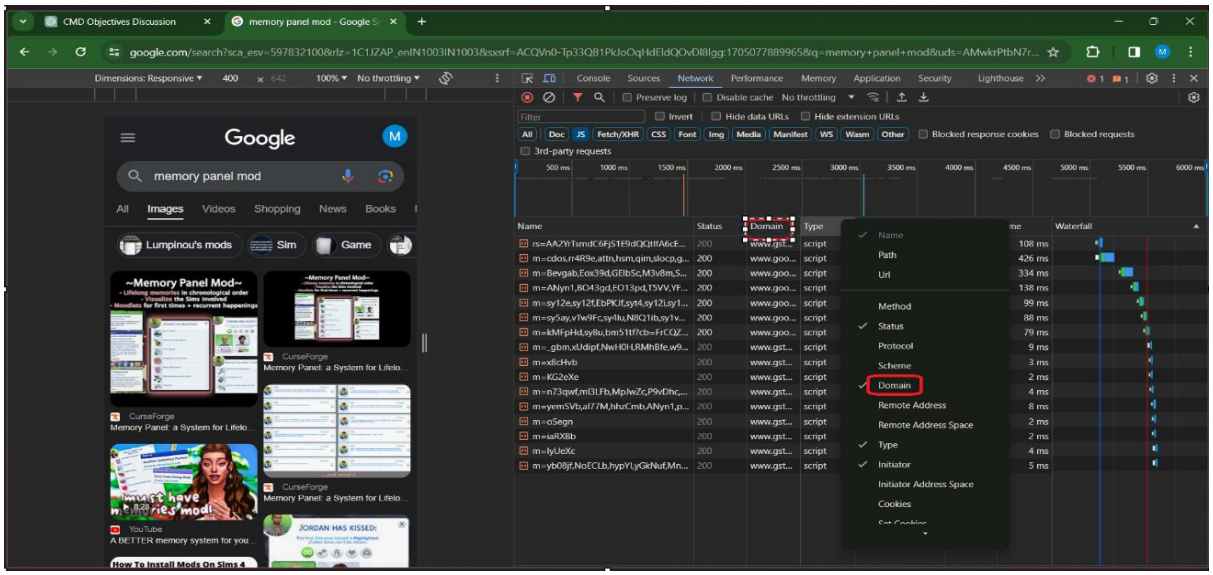
The Network panel logs all network activity in the Network Log. Each row of the Network Log represents a resource.

Defaults columns in network log are view in below window.



B. View more information

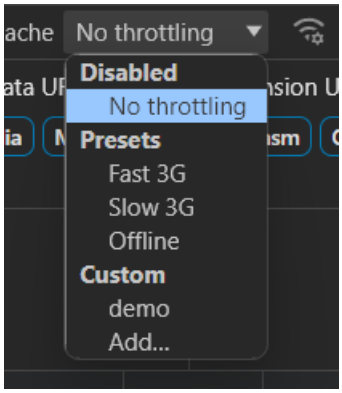
Right click on header of network panel to select domain.



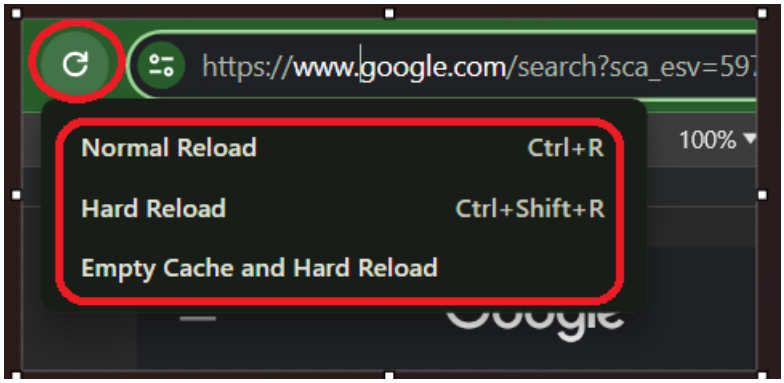
To hide the Domain column again click on header of network log and uncheck the domain.

C. Network Throttling:

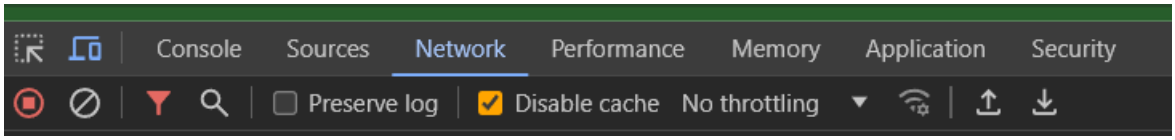
Simulates different network conditions like slow 3G, fast 3G, or offline mode to test and optimize performance under varied circumstances. using throttling the page, you can get a better idea of how long a page takes to load on a mobile device.



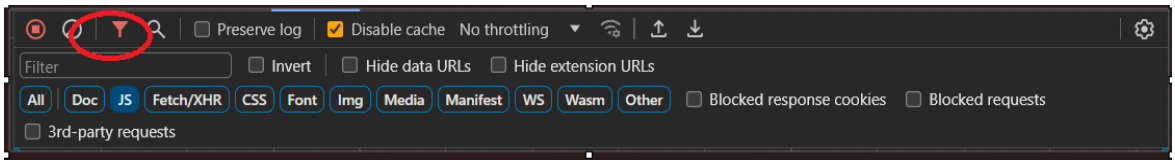
D. Empty cache and reload



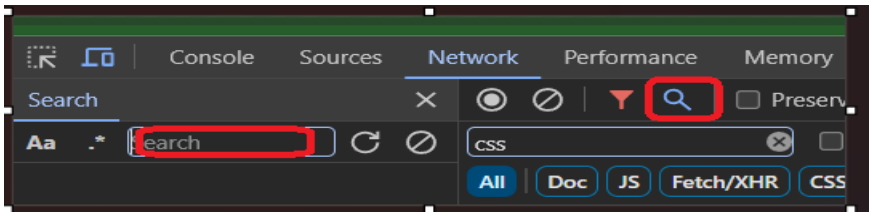
E. Disable cache



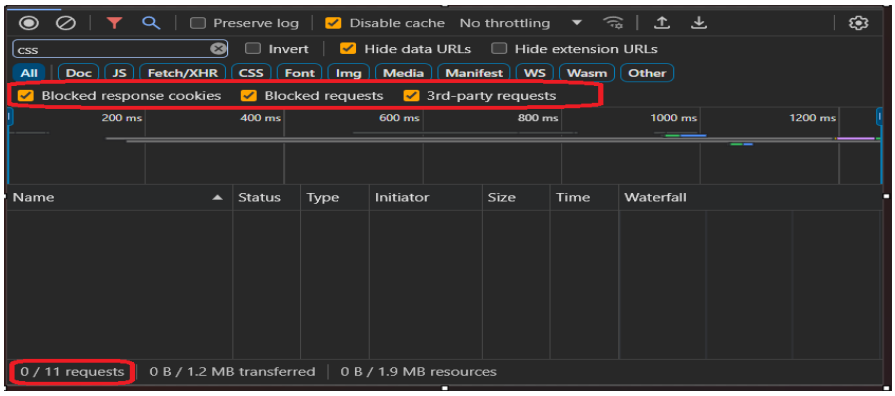
F. Filtering request by using resources



G. Search network headers and responses



H. Blocked response cookies

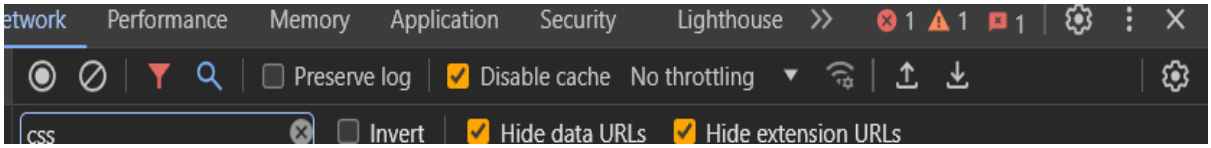


I. Hide data URLs

Data urls are small files embedded into other documents. Any request that you see in the Requests table that starts with data. To hide these requests, check Checkbox. Data URLs hidden from the Requests table. The status bar at the bottom displays the number of the shown requests out of the total.

J. Hide extension URLs

To focus on the code you author, you can filter out irrelevant requests sent by extensions you may have installed in Chrome. Extension requests have URLs that start with chrome-extension://.To hide extension requests, check Checkbox. Hide extension URLs.

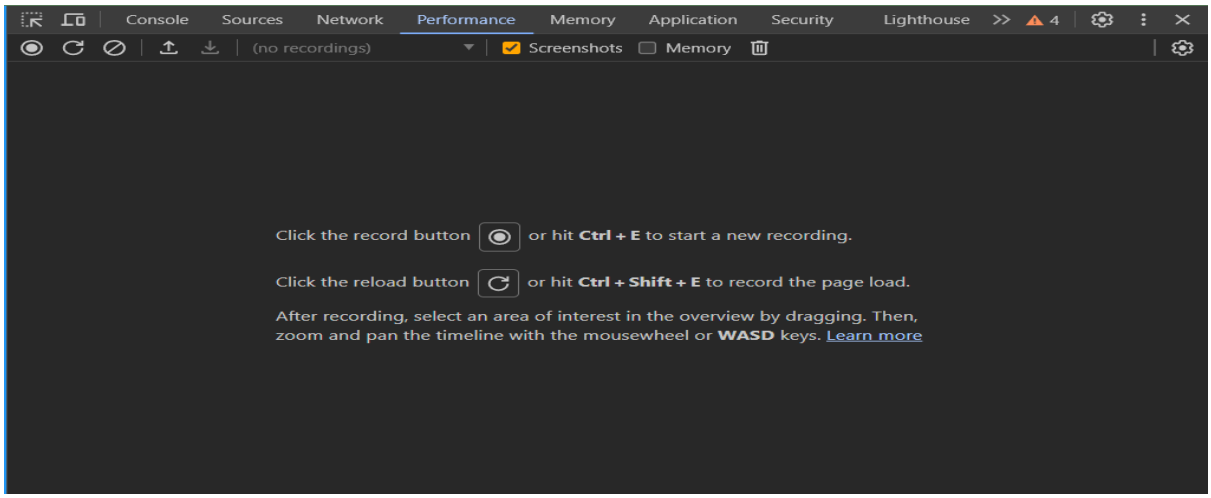


Submitted by: Manaswi M. Patil

E. Performance Panel

The Performance panel helps you analyse the runtime performance of your web application. Record and analyse the timeline to identify bottlenecks.

- Click →dev tool setting→click 'performance panel'→open.

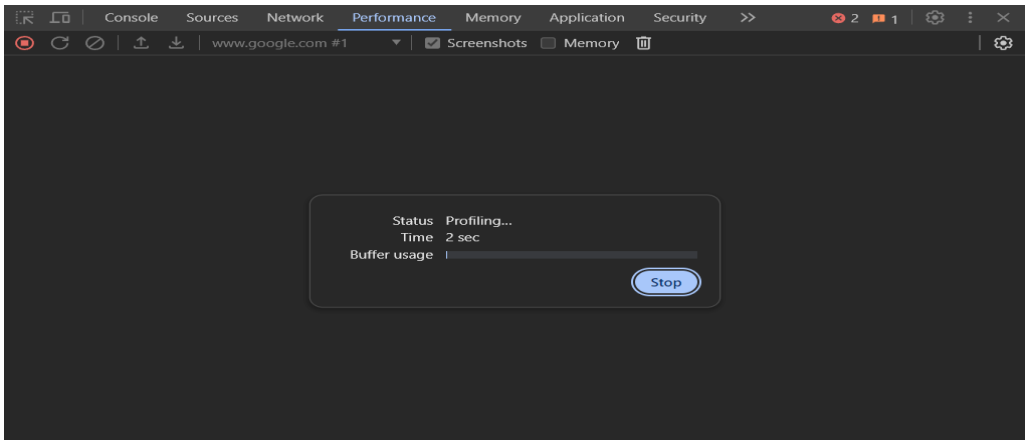


To test application performance with different numbers of processor cores, you can configure the Hardware concurrency value. DevTools displays a warning icon next to the Performance tab to remind you that hardware concurrency emulation is enabled.

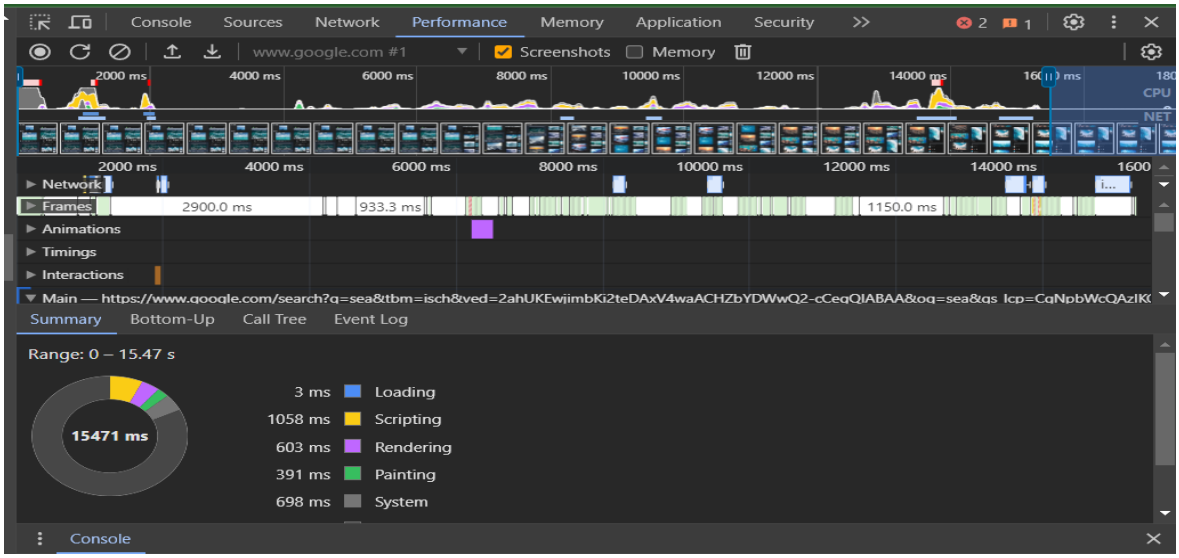
- To revert to the default value, click the Revert button.

Recording timeline

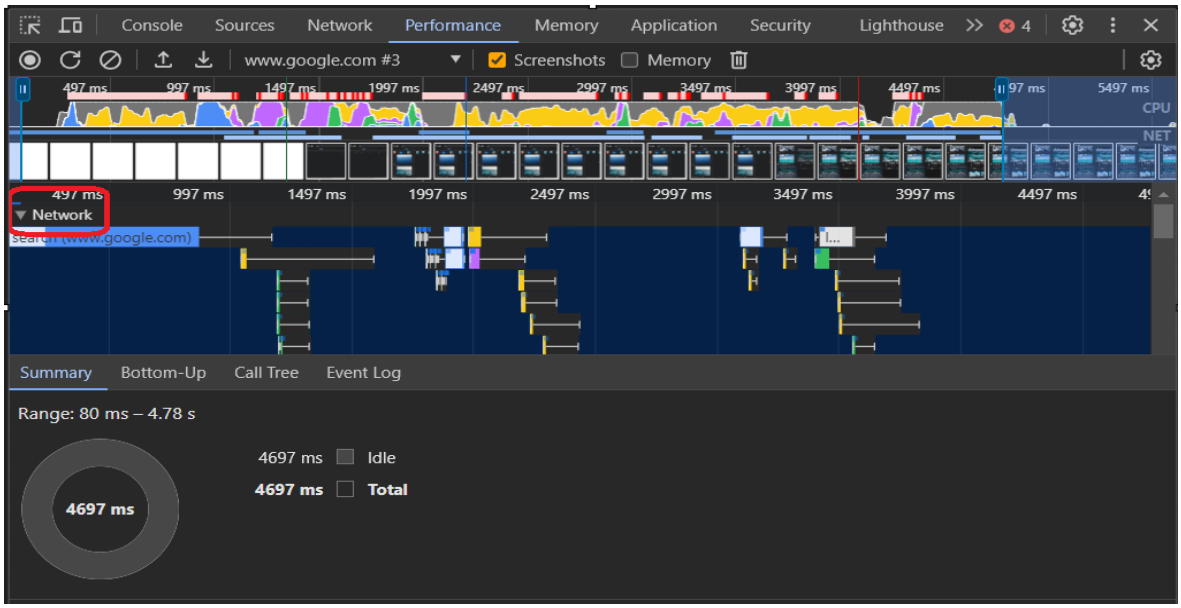
- Search for images in search bar of chrome→click on performance panel➔click on recording.



→stop recording→result will display like this

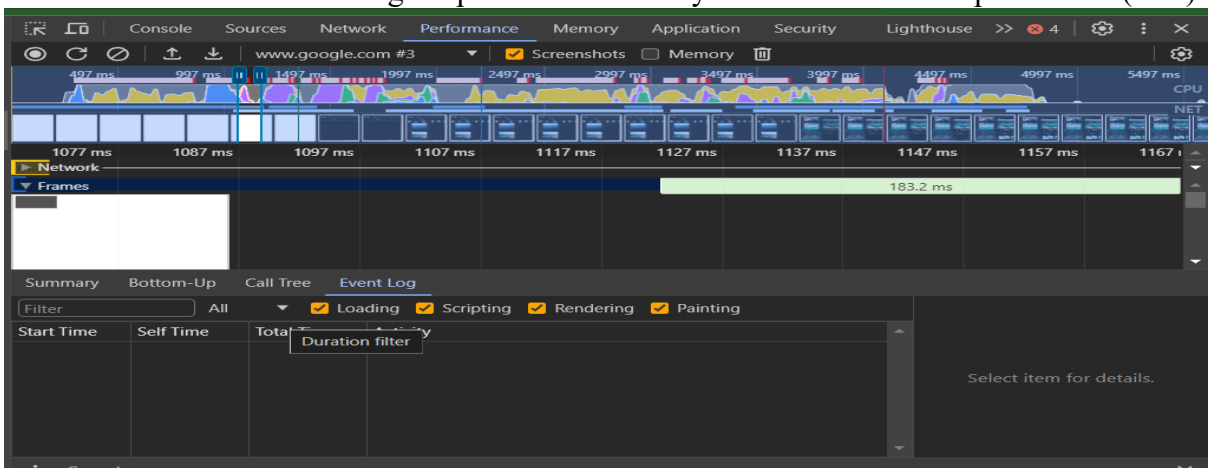


Click network



Analyse frames per second

The main metric for measuring the performance of any animation is frames per second (FPS).

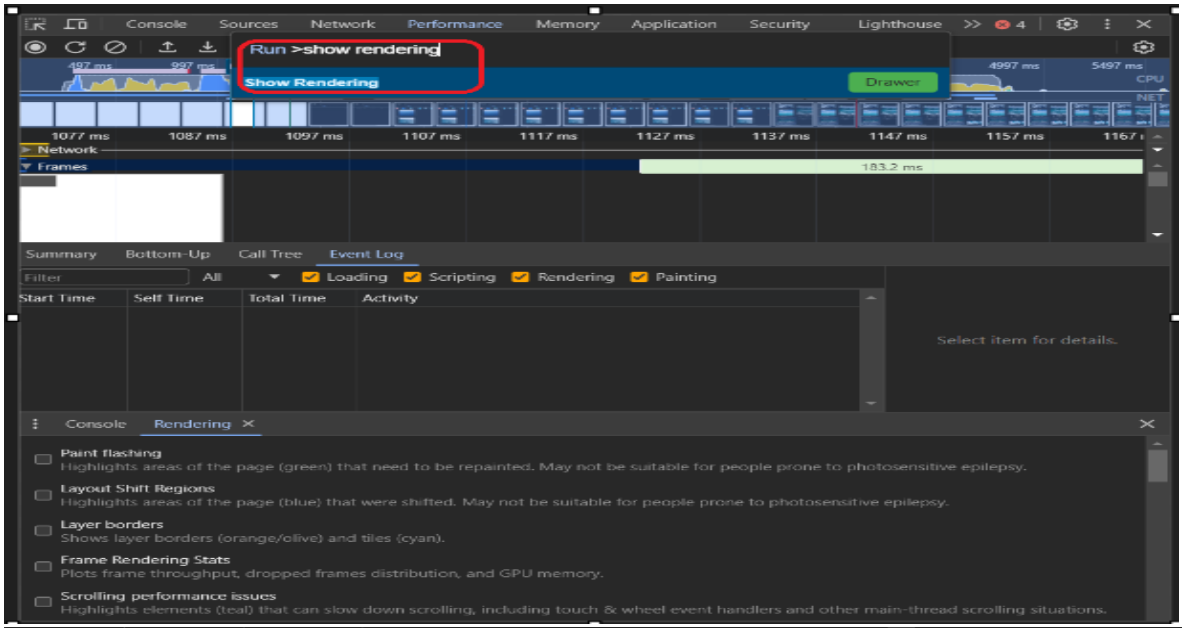


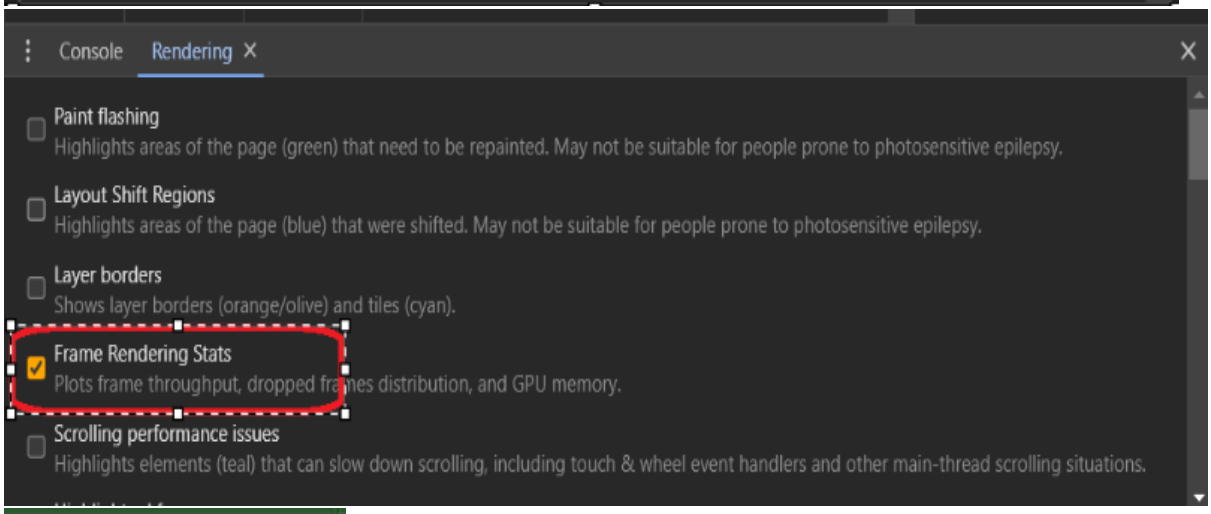
Open the FPS meter

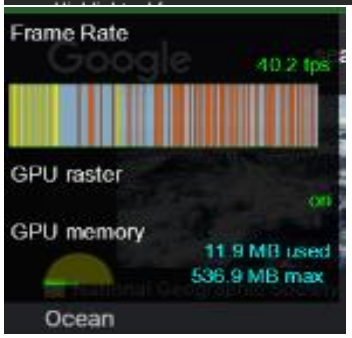
Another handy tool is the FPS meter, which provides real-time estimates for FPS as the page runs.

Submitted by: Manaswi M. Patil

➢ Press Control+Shift+P to open the Command Menu→ Type show rendering in the Command Menu and click on Drawer→The Rendering tab will open, enable FPS rendering stats. A new overlay appears in the top-right of your viewport→Disable it.



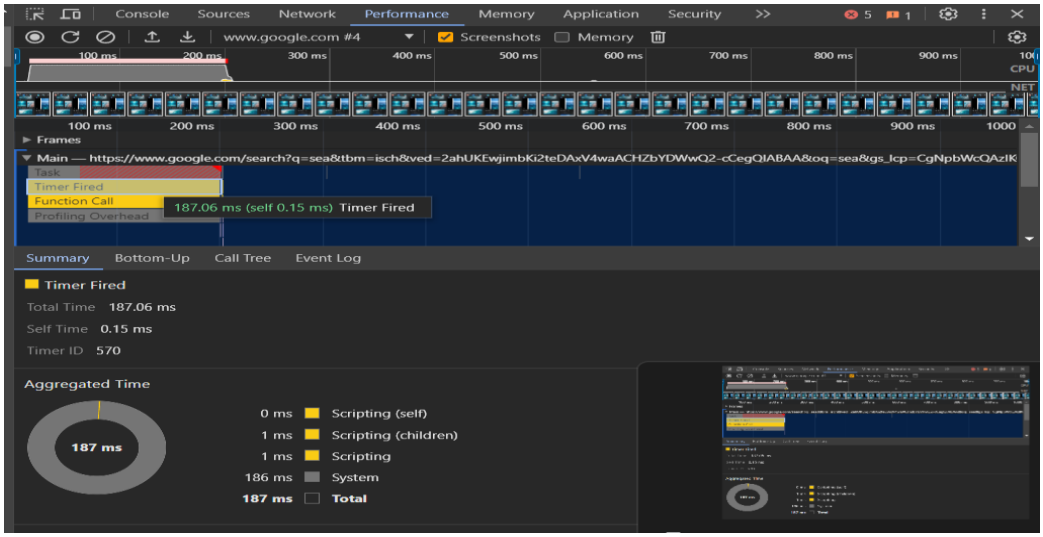




Finding the bottleneck

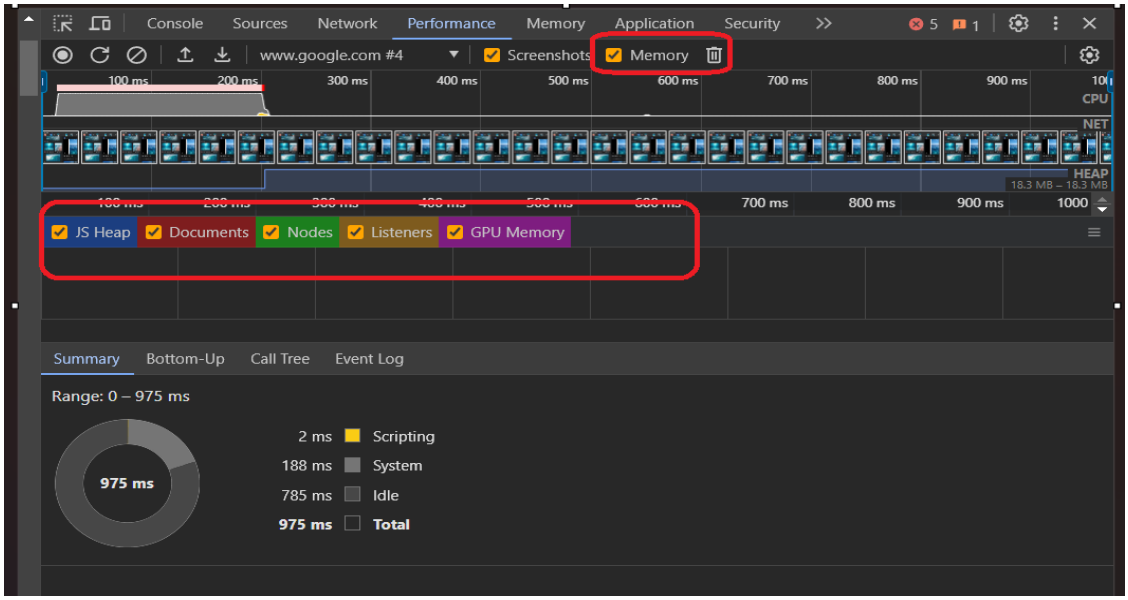
When no events are selected, summary tab shows a breakdown of activity. The page spent most of its time scripting.

Submitted by: Manaswi M. Patil



Memory metrices

DevTools displays a new Memory chart, above the Summary tab. There's also a new chart below the NET chart, called HEAP. The HEAP chart provides the same information as the JS Heap line in the Memory chart.

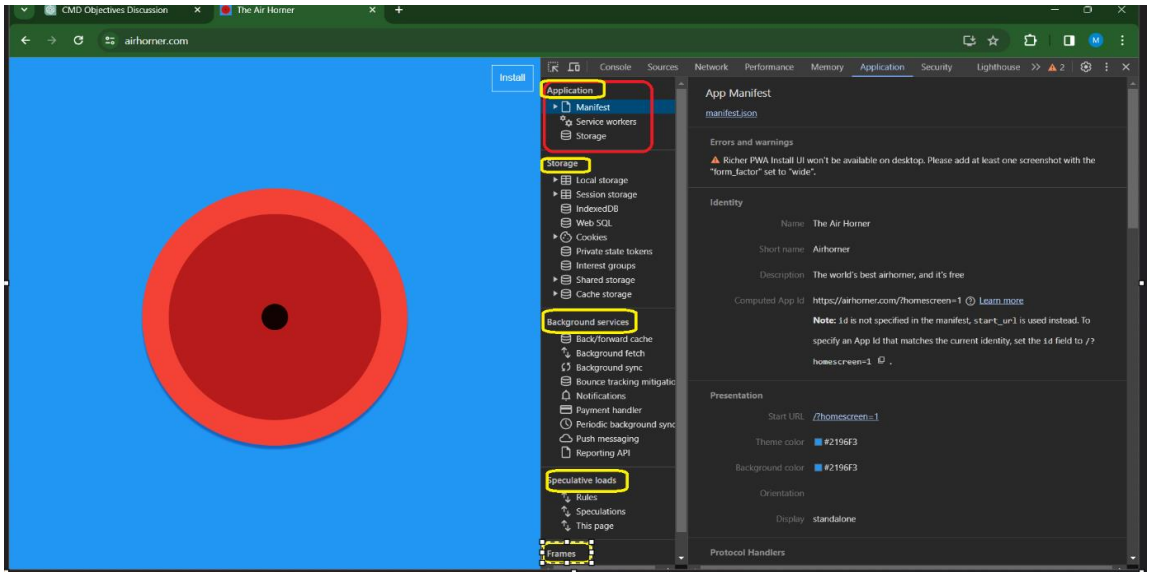


Submitted by: Manaswi M. Patil

F. Application Panel

The Application panel in browser developer tools. The Application panel deals with the storage, caching, and service workers of your web application. You can inspect and manage cookies, local storage, and more.

- Click devtool setting→click application panel to open→type in chrome' airhornre.com'→click on manifest.

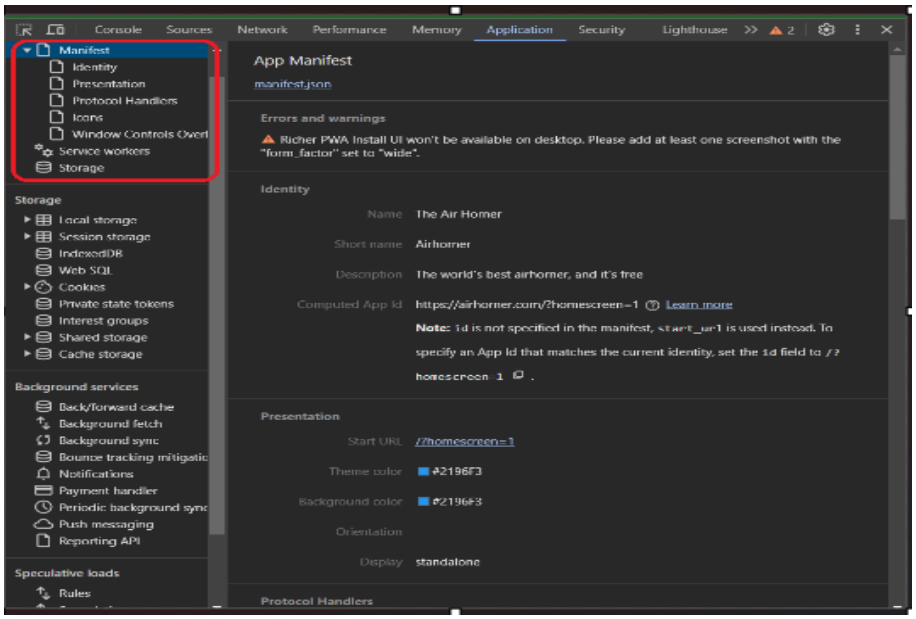


Features consists in Application panel are:

I. Application

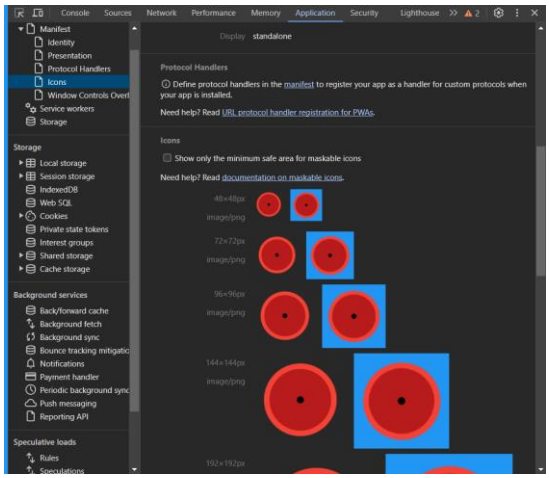
❖ Manifest

Use the Manifest pane to inspect your web app manifest. If your web app has a manifest file for Progressive Web Apps (PWAs), this shows details like the name, icons, and other settings defined in the manifest.



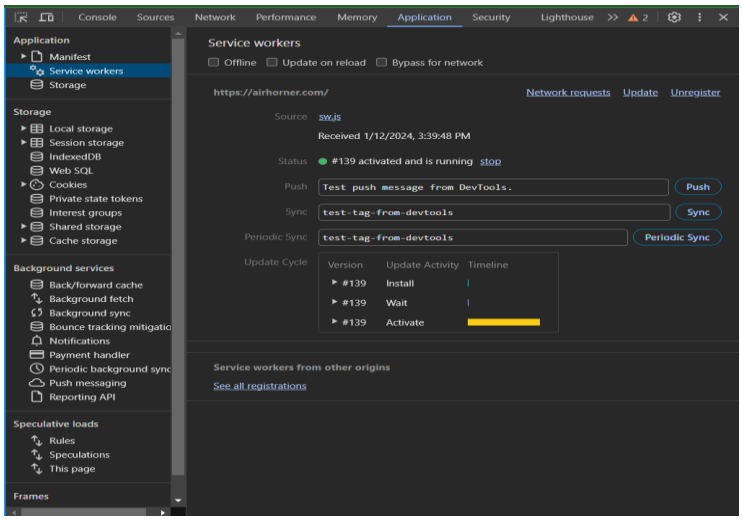
Submitted by: Manaswi M. Patil

- The Identity and Presentation sections just display fields from the manifest source in a more user-friendly way.
- The Protocol Handlers section lets you to test the URL protocol handler registration of your PWA with a click of a button. To learn more, see Test URL protocol handler registration.
- The Icons section displays every icon that you've specified and lets you check their masks.



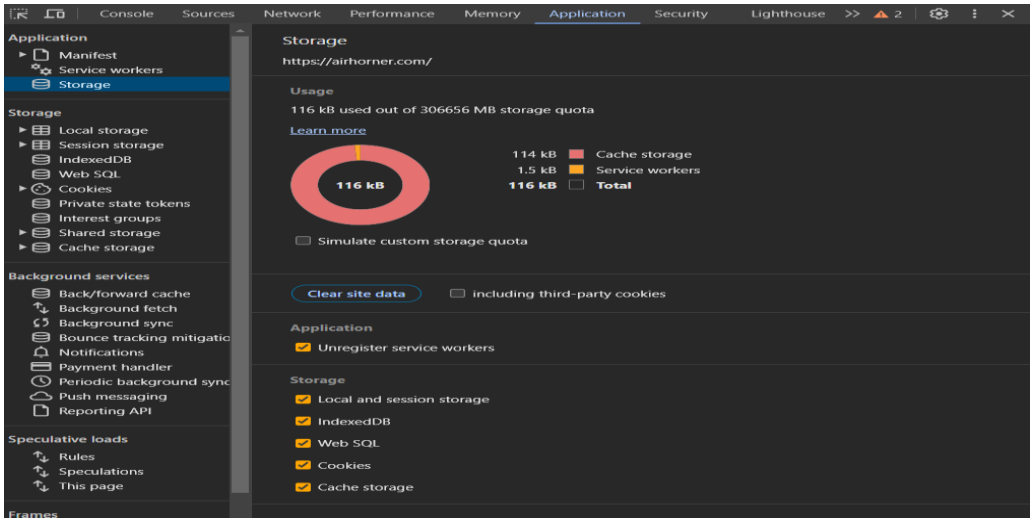
Service workers:

Use the Service Workers pane for a whole range of serviceworker-related tasks, like unregistering or updating a service, emulating push events, going offline, or stopping a service worker.



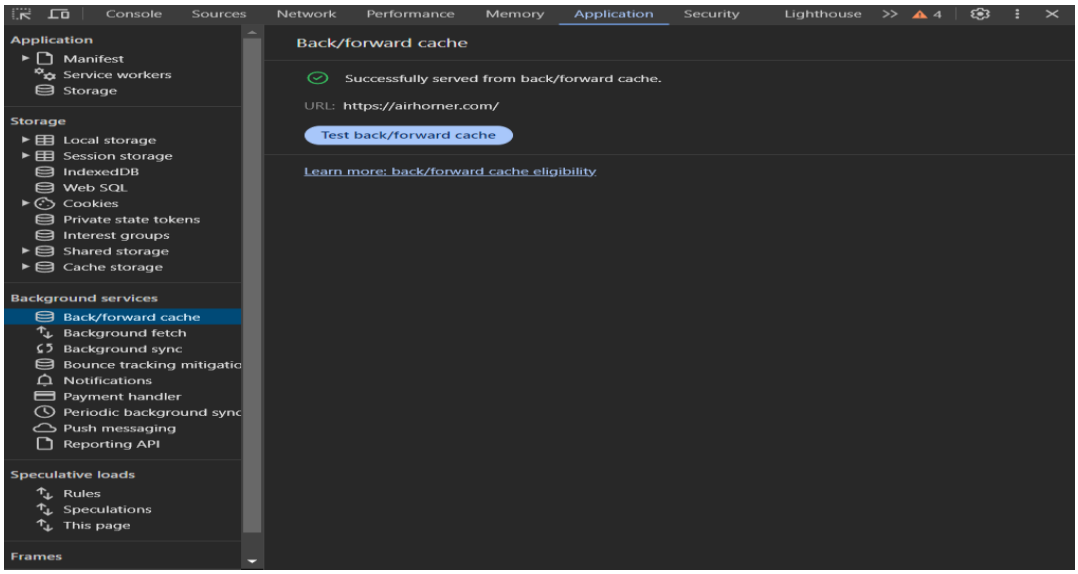
Storage

- View your service worker cache from the Cache Storage pane. →Unregister a service worker and →clear all storage →aches with a single button →click from the Clear storage pane



Test Bakforward cache

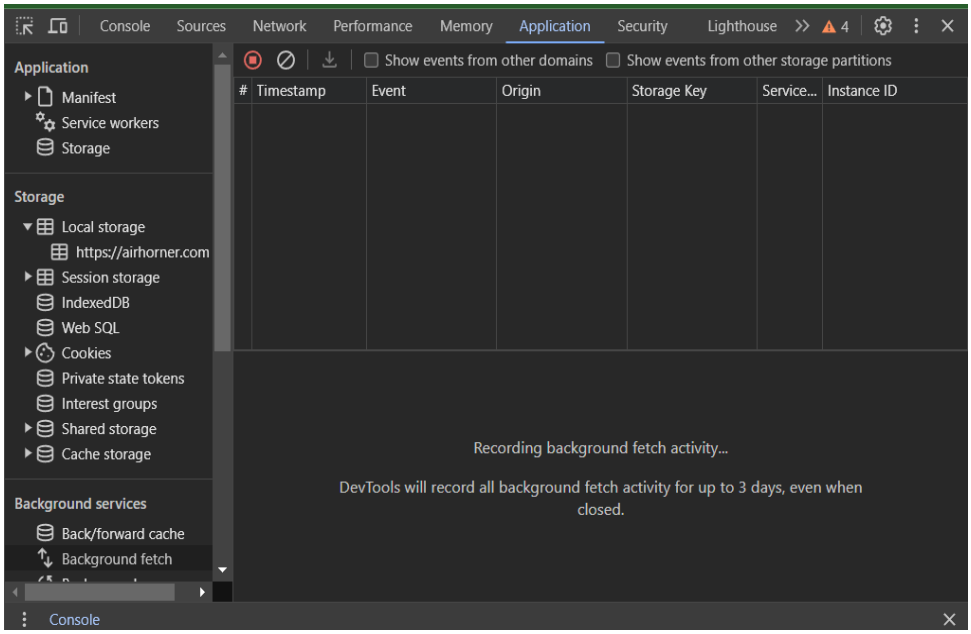
Back/forward cache is a browser optimization that enables instant back and forward navigation. It significantly improves the browsing experience for users—especially those with slower networks or devices. As web developers, it's critical to understand how to optimize your pages for bachforward cache across all browsers, so your users can reap the benefits.



Background fetch

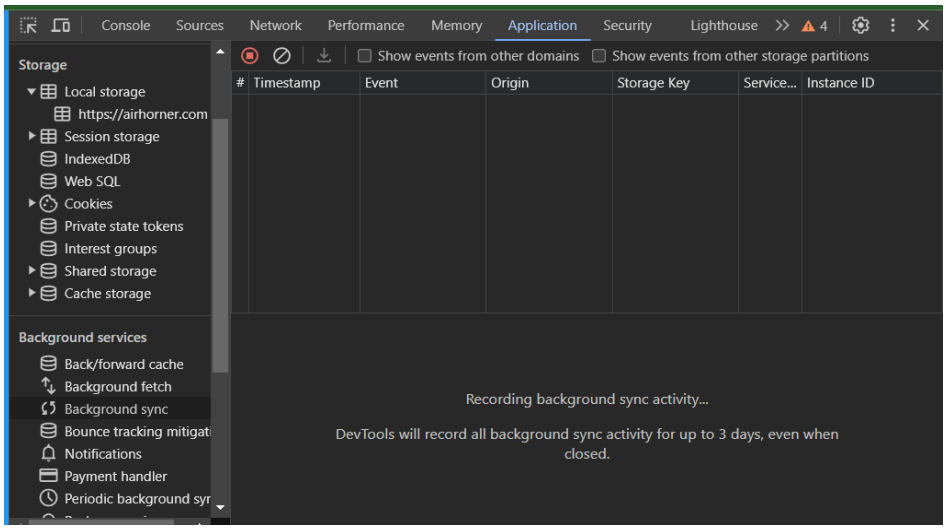
The Background Fetch API enables a service worker to reliably download large resources, like movies or podcasts, as a background service. To log background fetch events for three days, even when DevTools isn't open.

Submitted by: Manaswi M. Patil



Background sync

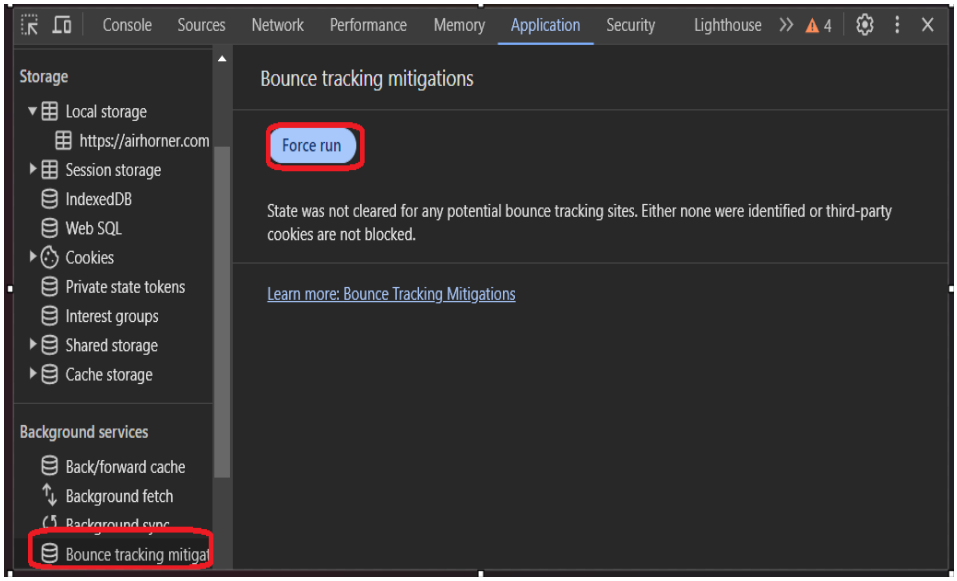
The Background Sync API enables an offline service worker to send data to a server once it has re-established a reliable internet connection. To log background sync events for three days, even when DevTools isn't open.



Bounce tracking mitigations

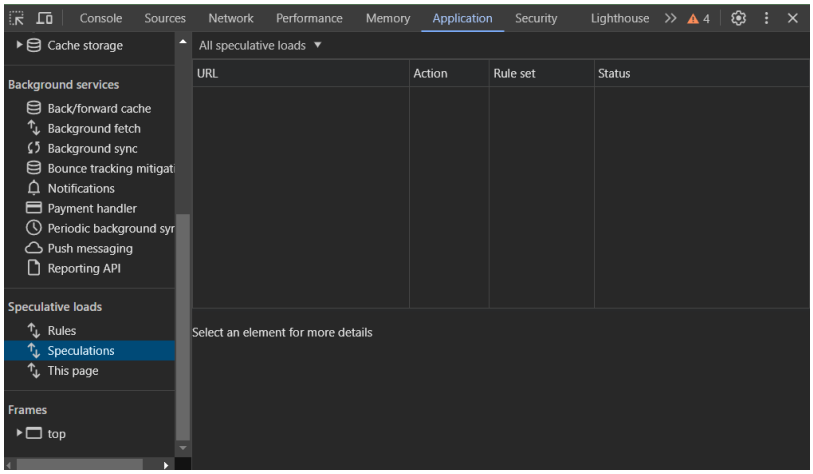
In chrome lets you identify and delete the state of sites that appear to perform cross-site tracking using the bounce tracking technique. You can manually force tracking mitigations and see a list of sites whose states were deleted.

- To force tracking mitigations:Block third-party cookies in Chrome→Navigate to →enable Three-dot menu→ Settings→ Security→Privacy and security→ Cookies and other site data > Radio button checked→Block third-party cookies.



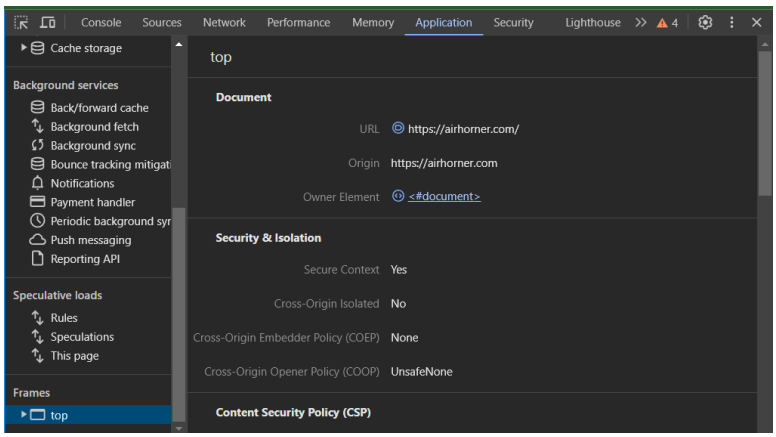
Spectulations

Contains a table with information on speculative loading attempts and their statuses. If an attempt failed, you can click it in the table to see detailed information and failure reason.



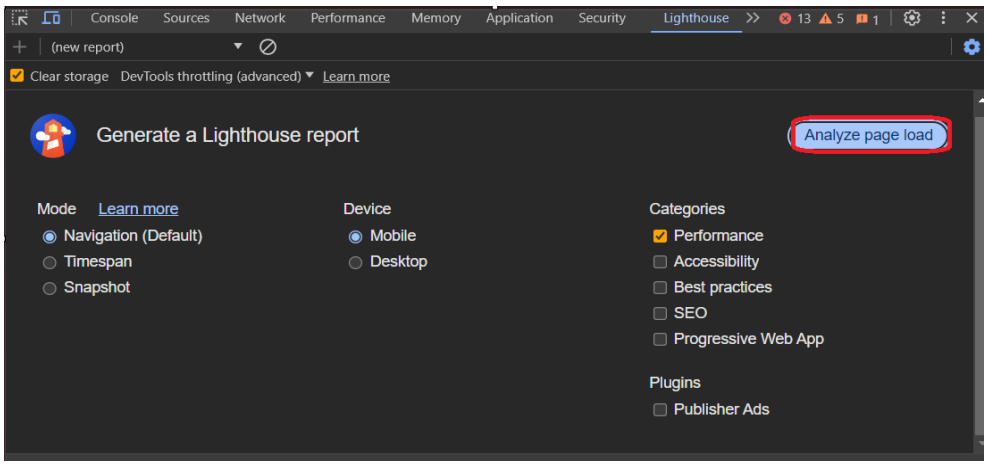
Frames

Frames let you divide web pages into multiple views that can load independently.

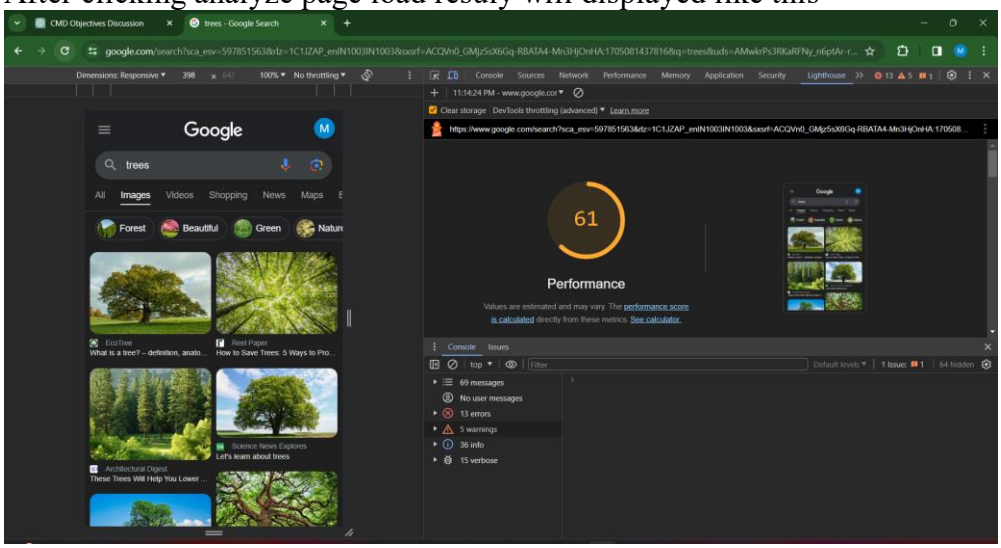


G. Lighthouse panel

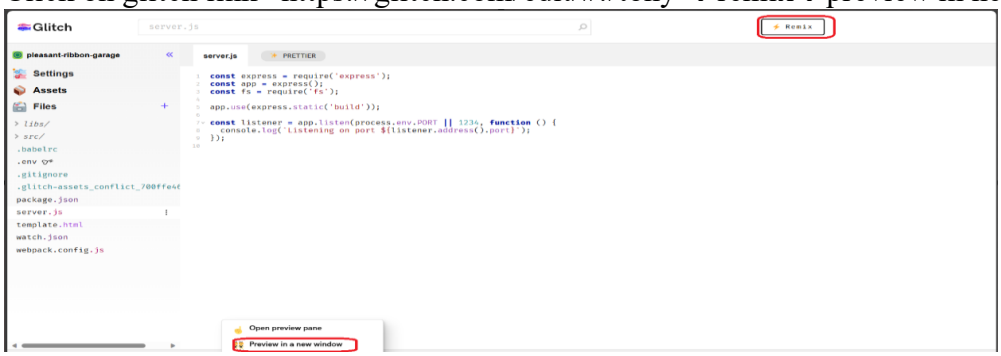
Lighthouse is a powerful tool in Google Chrome's DevTools that helps developers improve the quality of web pages by auditing various aspects of performance, accessibility, best practices, SEO, and more. The Lighthouse panel runs audits on a web page and generates a report with suggestions on how to enhance its performance and user experience.



After clicking analyze page load resuly will displayed like this



Click on glitch link https://glitch.com/edit/#!/tony →remix→preview in new window.



After 30 sec it will show the performance window.

Submitted by: Manaswi M. Patil

Lighthouse consists of following features

Audit Categories: covers several key aspects of web development through different audit categories. These categories include Performance, Accessibility, Best Practices, SEO (Search Engine Optimization), and Progressive Web App.

Running Audits: Developers can initiate Lighthouse audits directly from the DevTools. Upon running an audit, Lighthouse simulates the loading and rendering of a page and provides a comprehensive report on various aspects of web performance and user experience.

Scoring and Metrics: Lighthouse assigns scores for each audit category, providing a numerical representation of how well a web page adheres to best practices. Alongside scores, detailed metrics are provided, such as First Contentful Paint (FCP), Largest Contentful Paint (LCP), Total Blocking Time (TBT), and Cumulative Layout Shift (CLS) for performance.

Detailed Reports: Lighthouse generates detailed reports with actionable insights and recommendations. Each audit category is broken down into specific issues, with information on why each issue matters and how to address it.

Diagnostic Information: Lighthouse provides diagnostic information to help developers understand and resolve issues. This includes information about resources, network requests, and JavaScript execution.

Viewing Reports: After running an audit, developers can view the detailed Lighthouse report directly within the DevTools. The report includes a summary, detailed results, and opportunities for improvement.

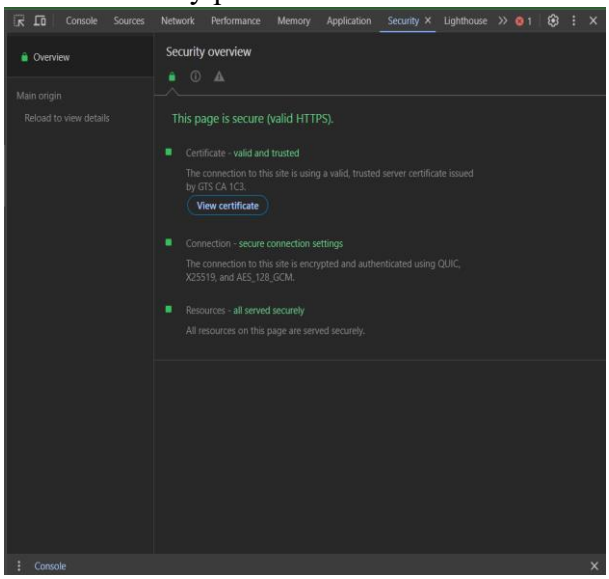
Command Line Usage: Lighthouse can also be run from the command line, making it suitable for automated testing and integration into build processes. This allows developers to regularly check the performance and quality of their web pages.

Integration with Other Tools: Lighthouse integrates with other tools and platforms, including online services and continuous integration systems. This allows developers to incorporate Lighthouse audits into their workflows seamlessly.

H. Security panel

The Security panel provides information about the security of the current page. It shows details about HTTPS, insecure content, and security issues.That provides insights into the security aspects of a web page. It offers information about the security protocols, certificates, and potential security issues on the loaded web page.

- Click security panel→click view certificate.



Security Overview: The panel typically includes a summary or overview section that quickly indicates the security status of the current page. It often includes icons or indicators to show if the connection is secure (using HTTPS) or if there are any security issues.

Connection Information: Details about the current connection, such as the negotiated security protocol (TLS/SSL), key exchange mechanisms, and cipher suites, may be available. This information helps developers understand the security protocols in use.

Certificate Details: Information about the SSL/TLS certificate used by the website is displayed. This includes details about the certificate issuer, expiration date, and the certificate chain.

Mixed Content Warnings: The panel may warn about mixed content, which is a security risk where a secure (HTTPS) page loads resources (like images or scripts) over an insecure (HTTP) connection. This can be a potential vulnerability, and the panel alerts developers to such issues.

Security Console Messages: Messages related to security events and issues are logged in the browser console. Developers can use the console to access more detailed information about security-related events.

I. Memory panel

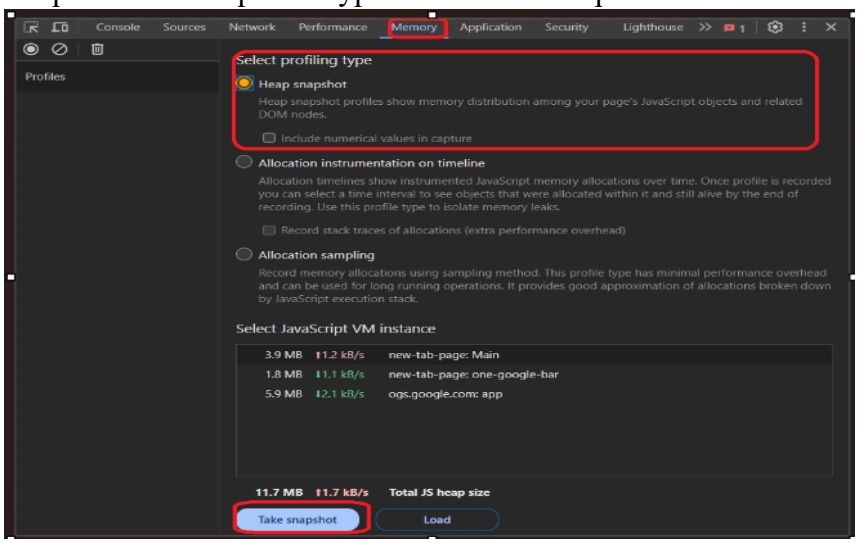
The "Memory" panel in Chrome DevTools is a tool designed to help developers analyse and optimize the memory usage of a web page or application.

Some features of memory panel

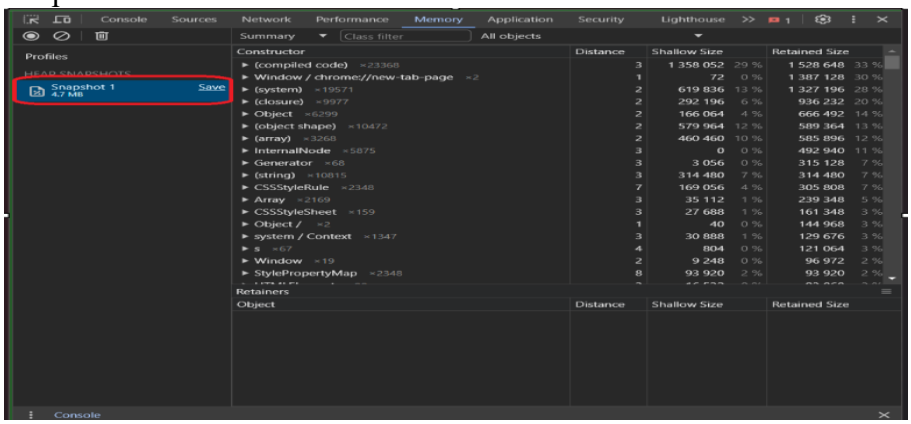
i. Heap snapshot

The "Memory" panel allows developers to take a heap snapshot, providing a detailed snapshot of the JavaScript heap at a specific point in time.The heap snapshot is a representation of all objects in memory, including their sizes and references. This can help identify memory leaks and inefficient memory usage.

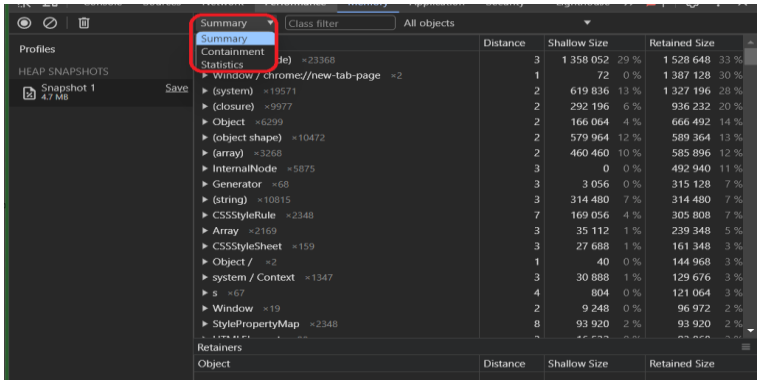
Open chrome inspect using shortcut or going with devtool setting→click on memory panel→select ptofilr type→click take snapshot.



Snapshot created.

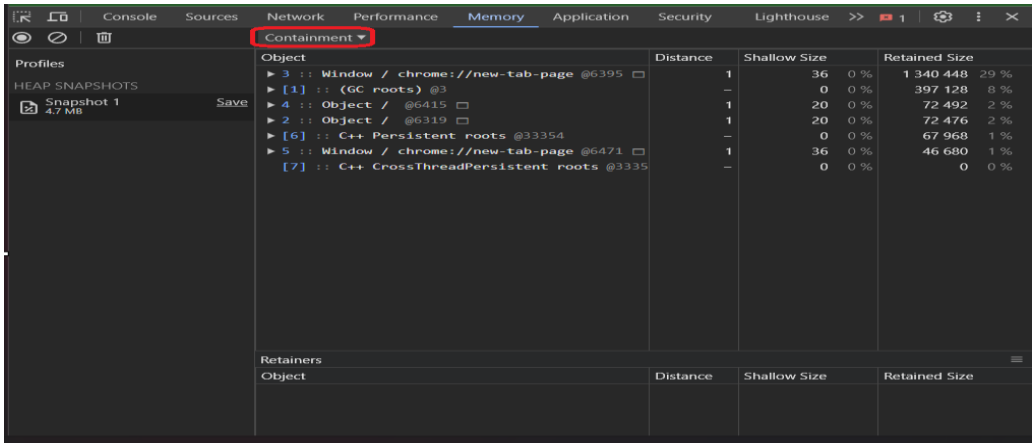


- To view created snapshots in different way→click on summary→click→view.



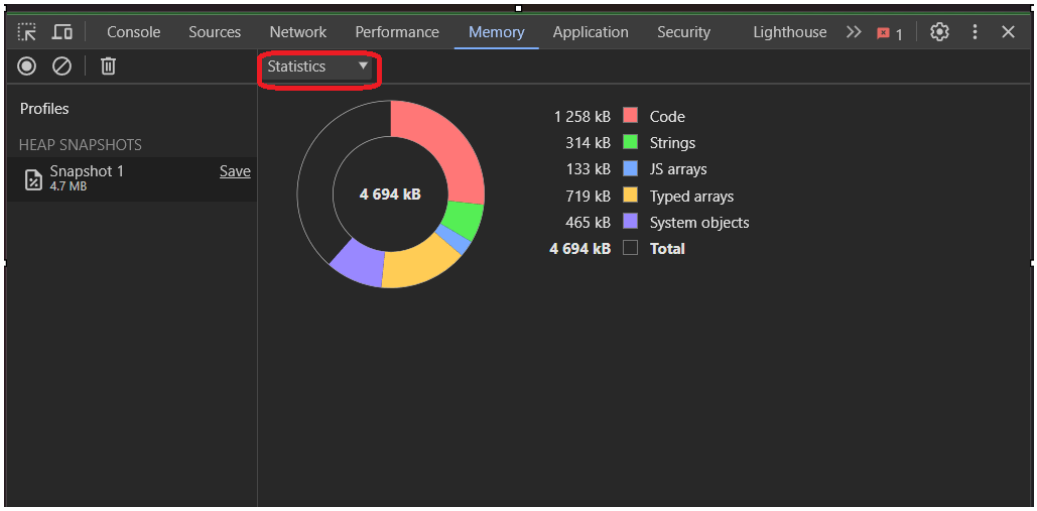
Containment view

Containment view allows exploration of heap contents. It provides a better view of object structure, helping analyse objects referenced in the global namespace (window) to find out what is keeping them around. Use it to analyse closures and dive into your objects at a low level



Statistics view

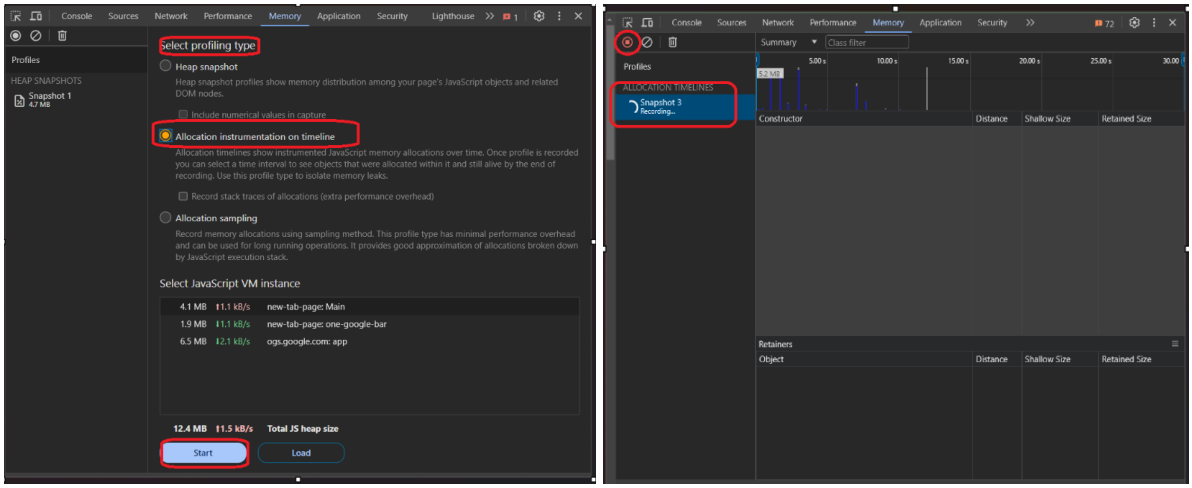
Statistics View displays a circular chart, which shows how much memory each type of object that is present takes up overall.



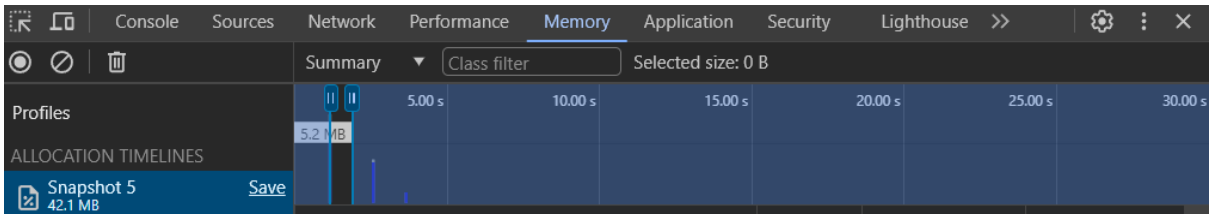
ii. Allocation Timeline:

Allocation timeline that visually represents memory allocations and deallocations over time. Developers can use the timeline to identify patterns and trends in memory usage, making it easier to correlate memory changes with specific actions or events on the webpage.

- Click on memory panel→click profiles→select type as **'allocation timeline'**→click to start→click on recording icon to stop recoding.



The height of each bar corresponds to the size of the recently allocated objects,s Blue bars indicate objects that are still live at the end of the timeline, gray bars indicate objects that were allocated during the timeline.



iii. Allocation sampling

The Allocation Sampling view maps memory usage to individual page components like documents, frames, web workers, and graphics layers. This reveals the source of any high usage. This combines the detailed snapshot information of the heap profiler with the incremental updating and tracking of the Performance panel.

- Select the Allocation sampling radio button→click the "**Start**" button.

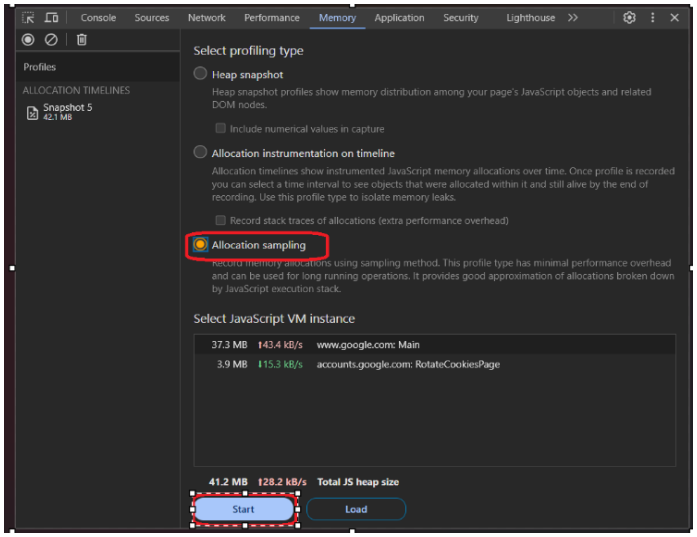
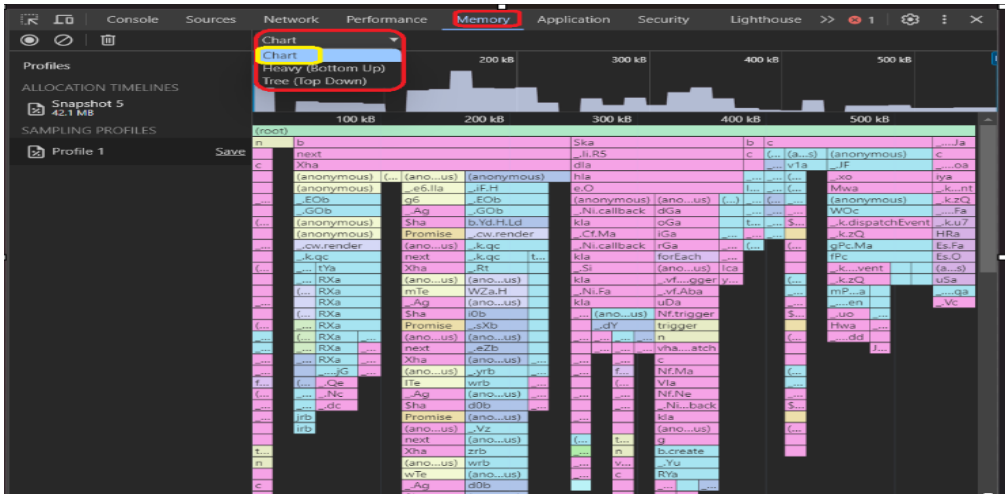
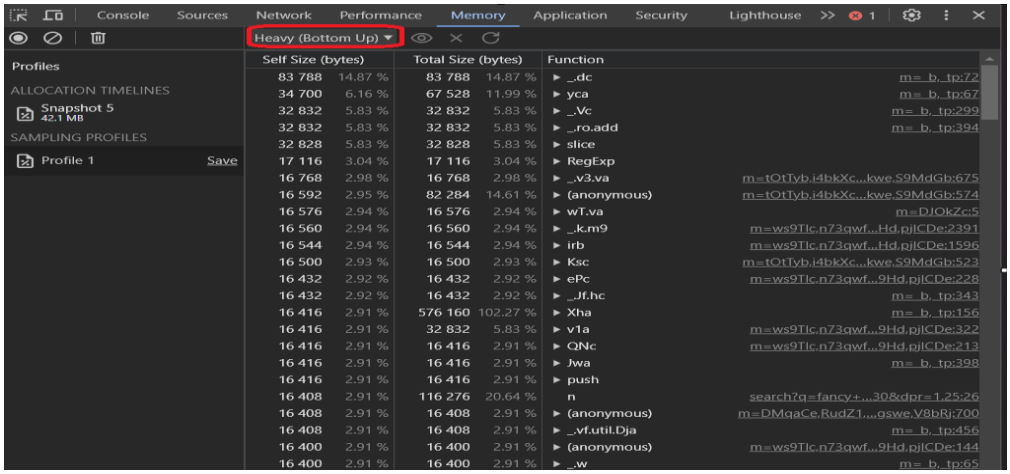


Chart view

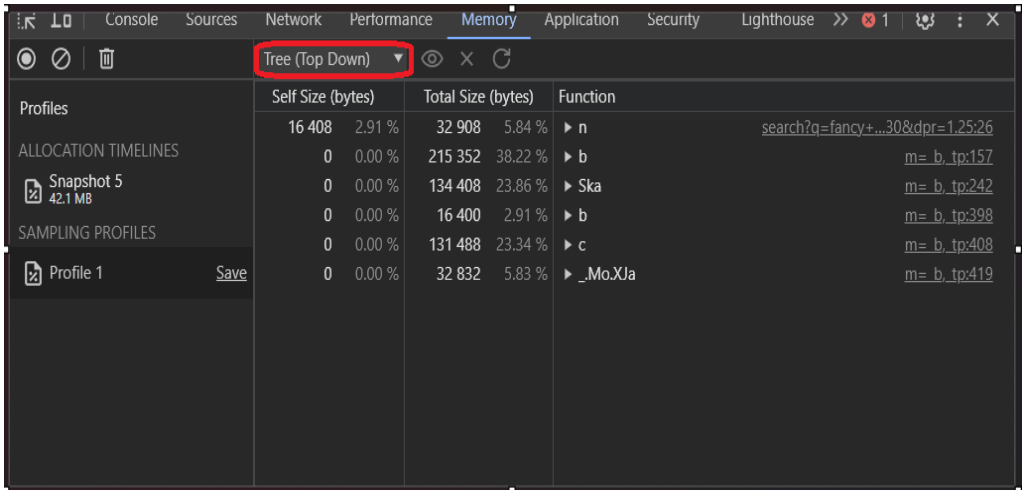


Heavy bottom up view



Tree top down view

Submitted by: Manaswi M. Patil

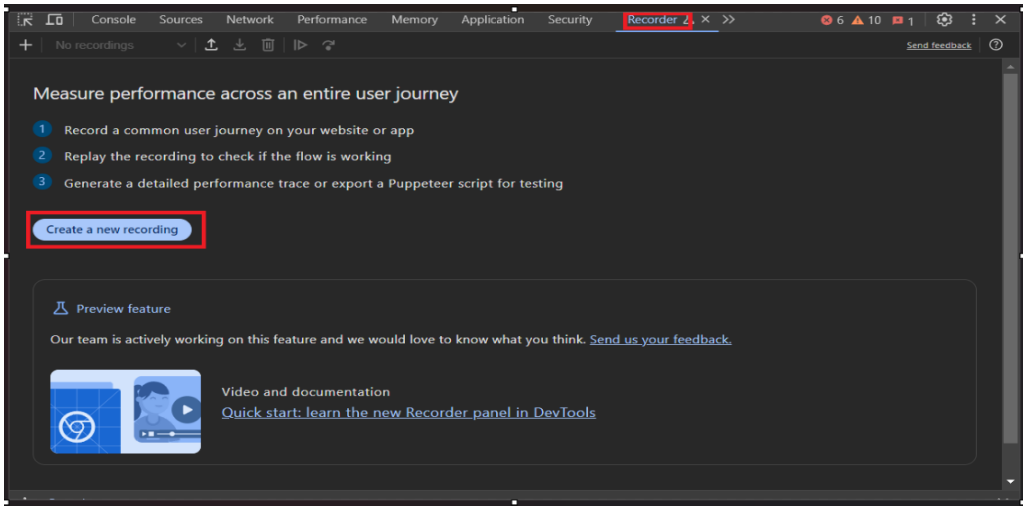


Submitted by: Manaswi M. Patil

J. Recorder panel

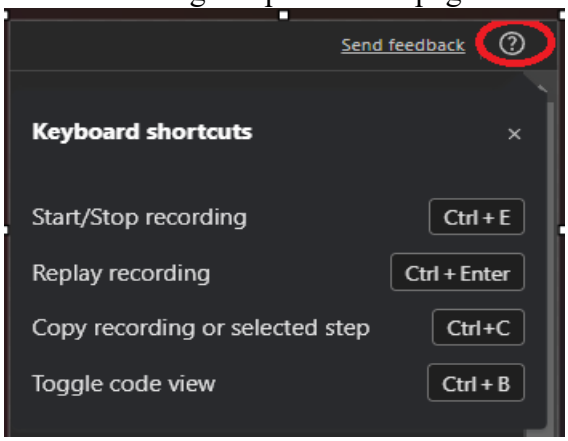
Recorder API is a preview feature that allows user to extend the Recorder panel in Chrome Developer Tools.

- Click chrome inspect→click **recorder panel**→click to create new recording.



Shortcuts to recorder faster

- Click right top corner of page→click on **'? '** this symbol to add shortcuts.

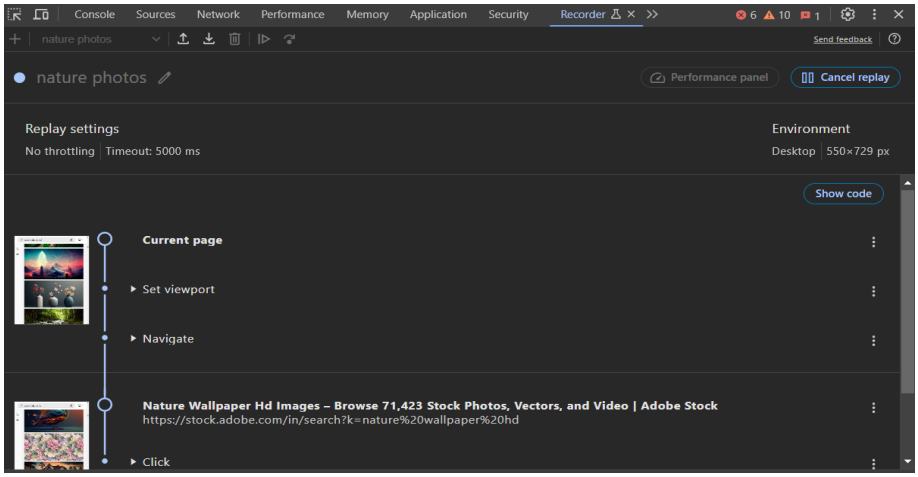


Recording a execution flow of a website

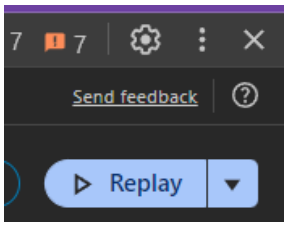
How to record user flow?

- Click on 'https://stock.adobe.com/in/search?k=nature%20wallpaper%20hd' this link →open in chrome browser→go to record panel→start recording→set recording name→start recording→click to end recording.

Submitted by: Manaswi M. Patil

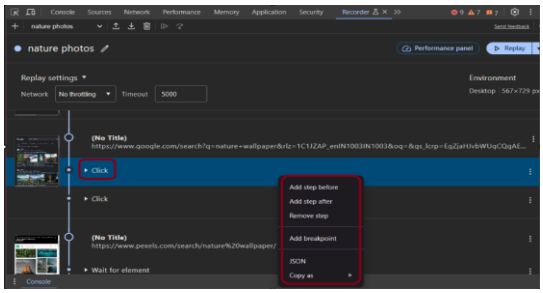


➢ Replay the recording using recoder.

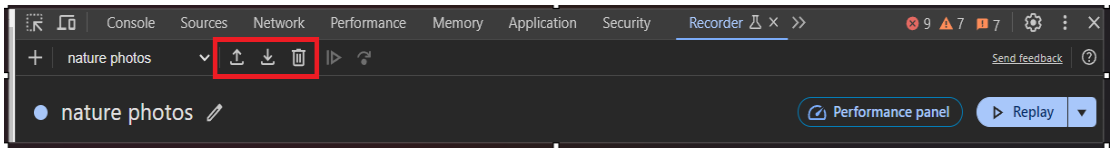


Add or remove content just clicking on element

➢ Click on text→where you want to edit→right click on text→select option you need.

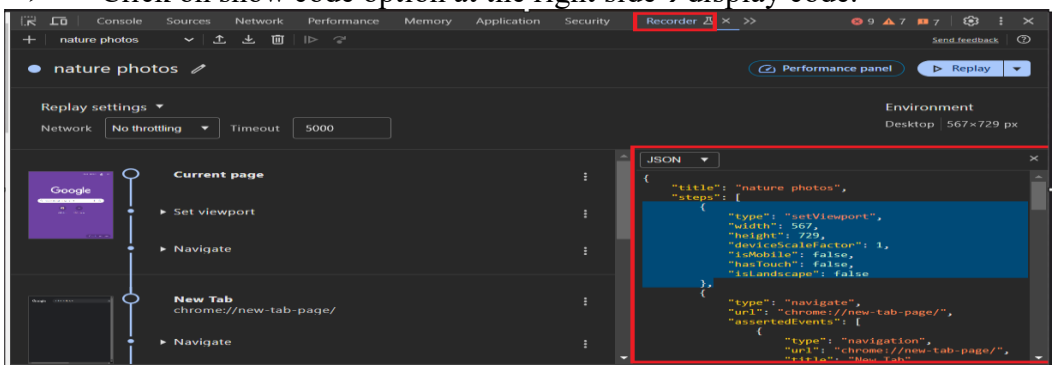


How to delete code?



Code Inspect

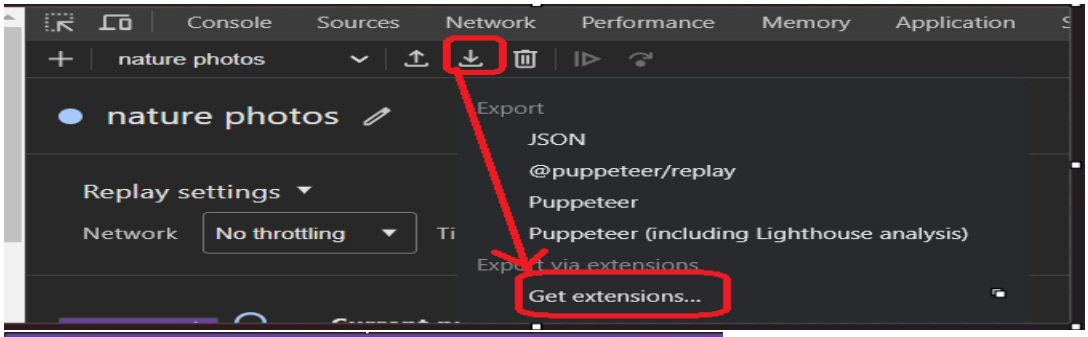
➢ Click on show code option at the right side→display code.

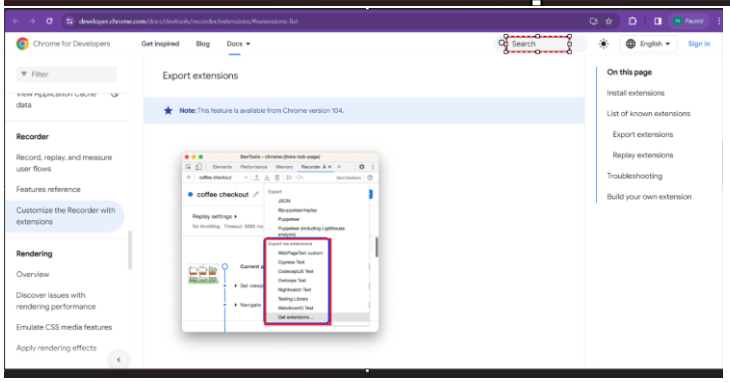


Submitted by: Manaswi M. Patil

How to install extension?

Click down arrow from the panel which located besides of delete icon→click→click to get extension,





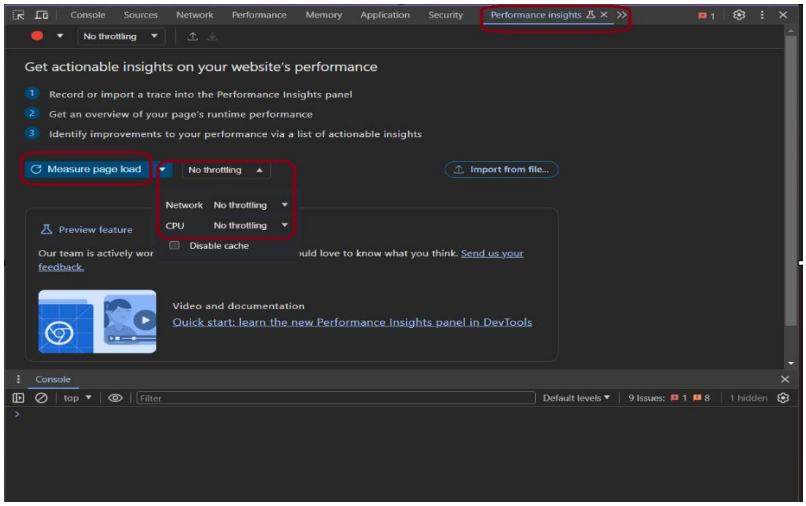
- Search for cypress extension , nightwatch extension and many more .
- **Cypress** is a front end testing tool built for the modern web.
- **Nightwatch** extension lets you export JSON user flows as Nightwatch .
- **CodeceptJS extension** lets you export JSON user flows as CodeceptJS test script.
- **Owloops extension** lets you export recordings as Owloops tests.
- **Testing Library extension** lets you export JSON user flows as Testing Library script. Testing Library has simple and complete testing utilities that encourage good testing practices.
- **WebdriverIO extension** lets you export JSON user flows as WebdriverIO test script.
- **WebPageTest extension** lets you export user flows from the Recorder directly as WebPageTest Custom scripts to measure site's performance.

K. Performance insights

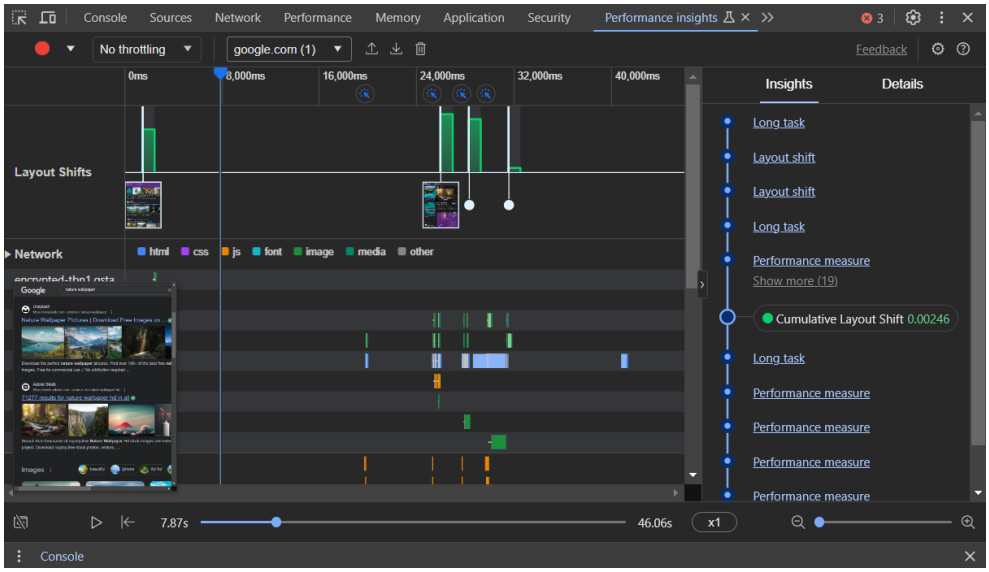
The new Performance insights panel is an experiment to address these three developer pain points when working with the existing Performance panel:

- Too much information.
- Hard to distinguish between use cases.
- Requires deep expertise of how browsers work to use effectively.

➢ Click chrome inspect→used shortcut key **'ctrl+shift+I'**→click performance insights panel→click measure page load →click to disable cache→type something in search bar of window which you want to record→or click this https://stock.adobe.com/in/search?k=nature%20wallpaper%20hd →start recording→click to stop.



➢ Output window display like following.

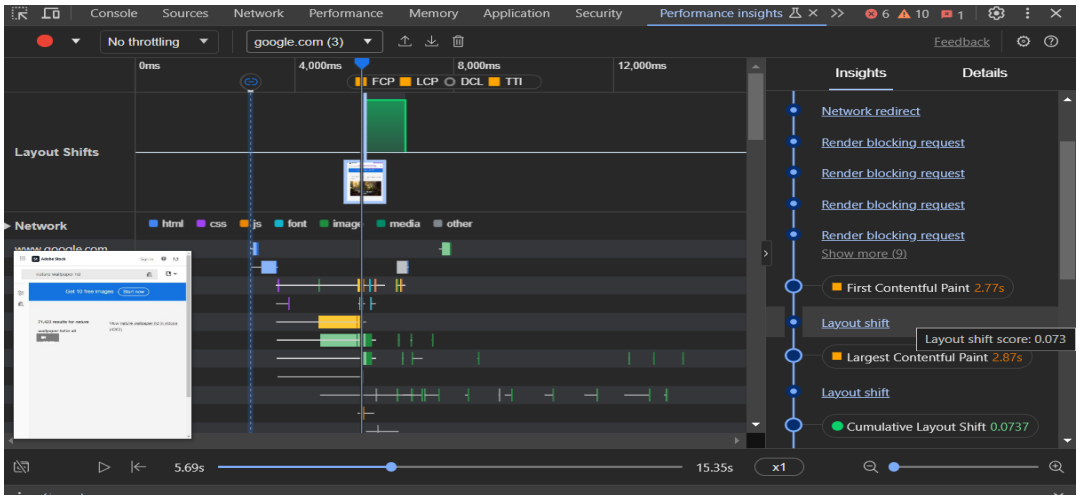


➢ Use control from bottom to replay your recording.

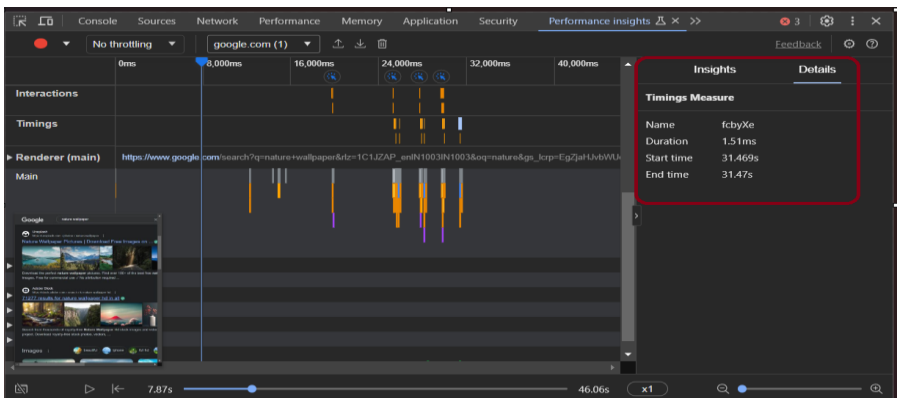
View of Performance Insights

First Contentful Paint (FCP): FCP measures the time it takes for the first piece of content to be rendered on the screen.FCP provides insights into how quickly users see any part of the page. It includes text, images, and non-white canvas elements. A faster FCP indicates a quicker perceived page load.

Largest Contentful Paint (LCP): LCP measures the time it takes for the largest content element (image or text block) to be rendered on the screen. LCP focuses on the main content that users are likely interested in. It is a key metric for assessing a user's overall experience, especially when dealing with media-rich websites. Improving LCP contributes to a better user experience.



Details of Performance Insights

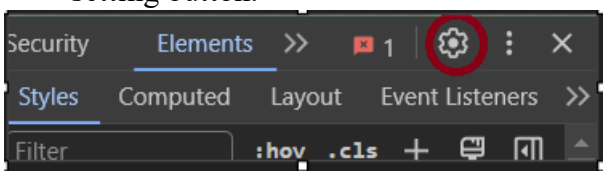


Submitted by: Manaswi M. Patil

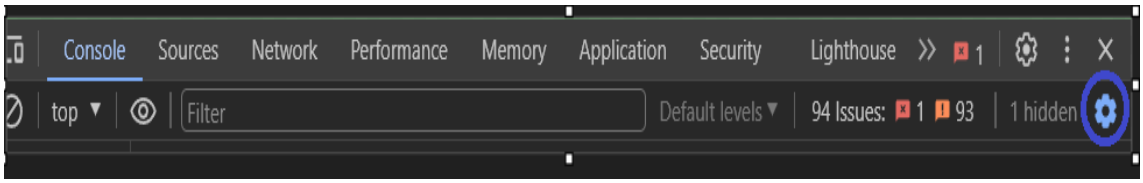
DEVELOPER SETTING

Chrome DevTools provides a set of settings that allow you to customize and configure various aspects of the developer tools.

- Click on right top corner in inspect window→to open '**developer tool setting** '→click on setting button.

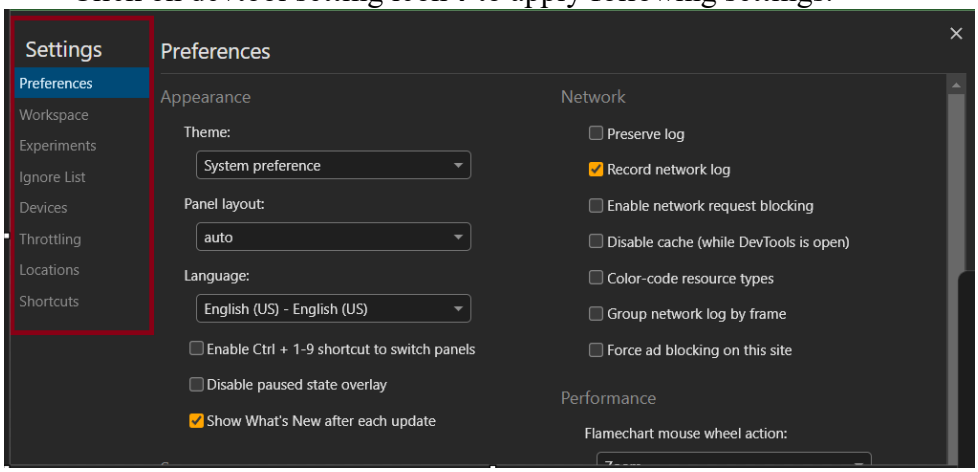


- Click on right top corner in inspect window→to open **'panel setting'** →click on setting button.



List of devtool settings:

- Click on devtool setting icon→to apply following settings.

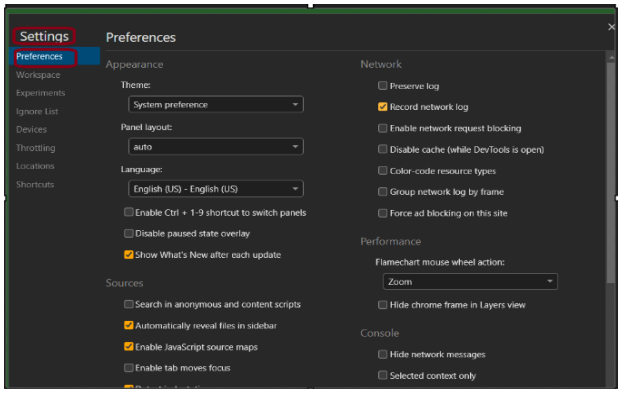


1. Preferences

This tab lists both general customization options and panel specific ones.

- Click on devtool setting icon→go preferences to apply changes.

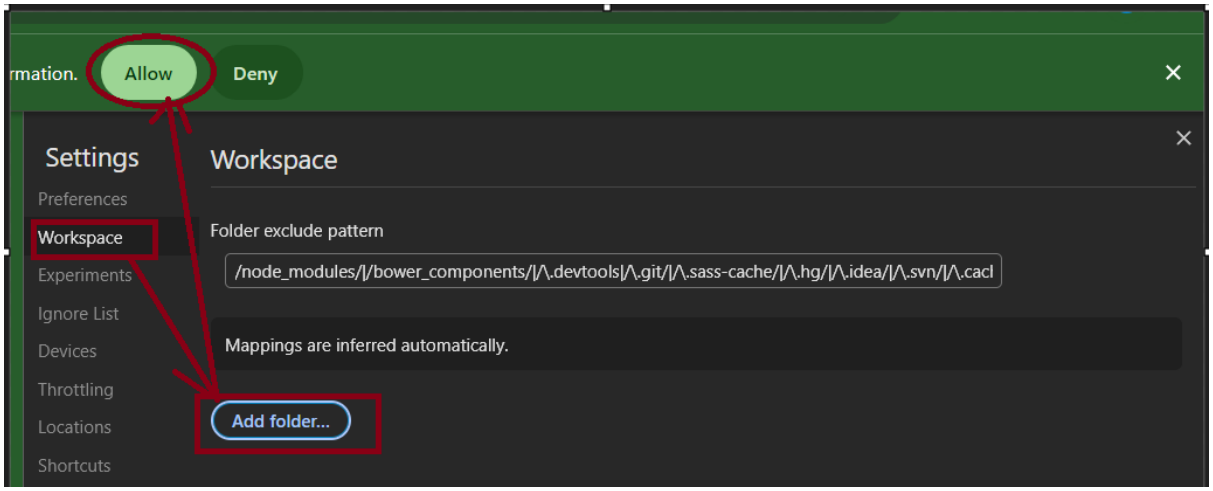
Submitted by: Manaswi M. Patil



2. Workspace

Allows you to make changes to local files directly from DevTools.

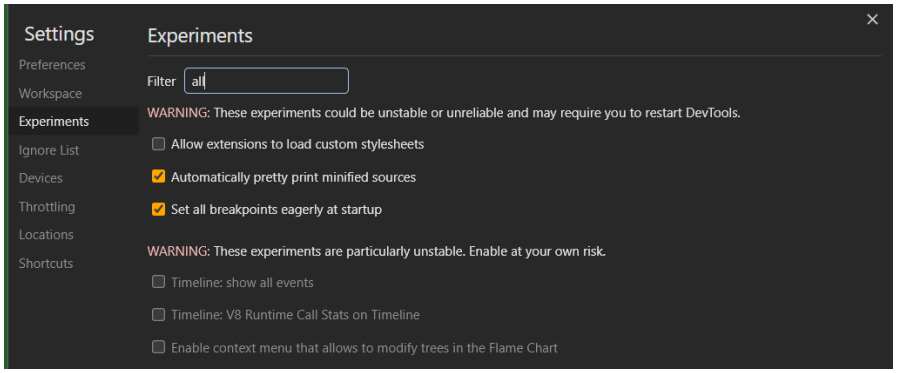
- Click on Workspace tab→click Add folder→Select the folder from your sources→ Click Allow in the prompt at the top→to let DevTools make changes to source.



3. Experiments

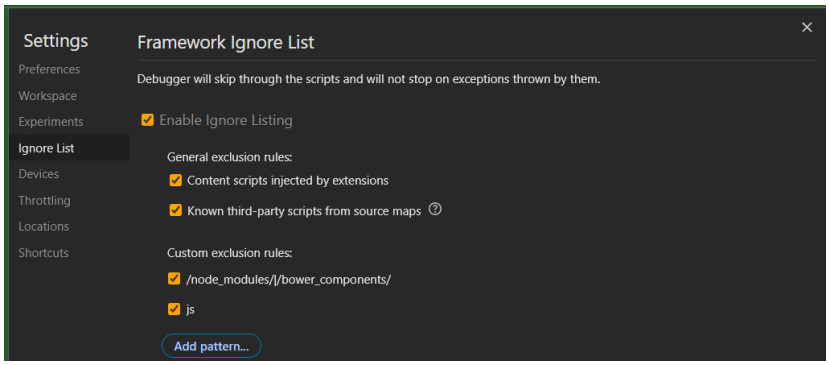
Enabling experimental features allows developers to test upcoming functionalities, improvements, or changes to the DevTools interface. It's important to note that these features may be unstable or subject to change, and they are not recommended for production use.

- Click dev setting→click Experiments button→search for the experiment which would you like to try in the Filter textbox→ Enable the checkbox→ next to the experiment→ Close Settings→ If need then click Reload DevTools in the prompt at the top.



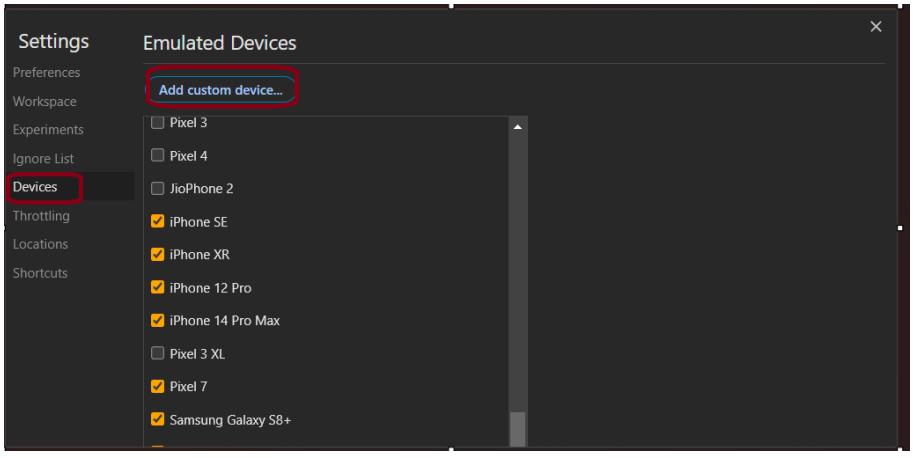
4. Ignore list

- Open dev Settings→click on ignore list button→ check or clear Settings→Enable Ignore Listing→This is the main switch for all ignore-listing capabilities→you can add pattern using 'add pattern' button.



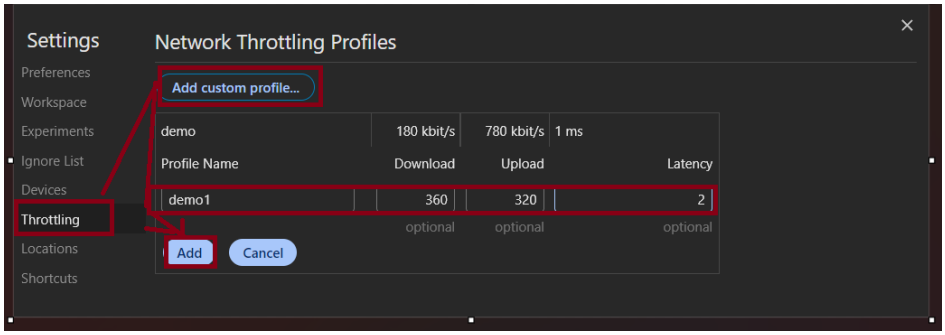
5. Devices

- Click dev Settings→click **device**→ enable the checkbox next to a device you want to add→You can add custom devices.



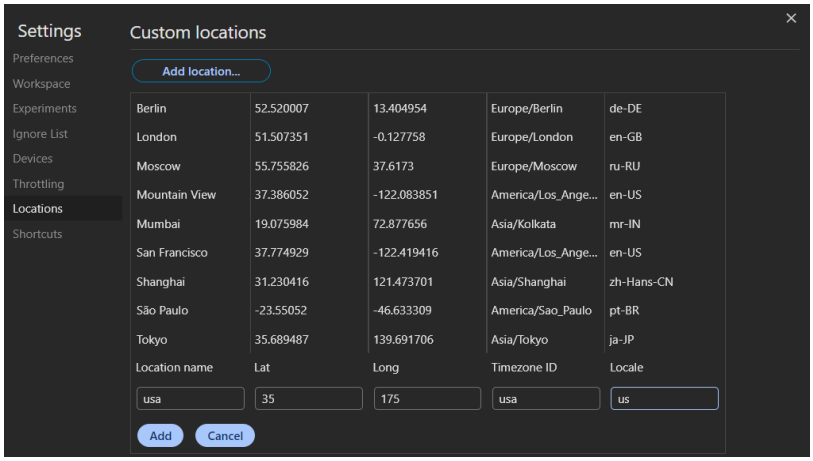
6. Throttling

- Click dev setting→click **throttling** tab→add custom profiles →click to add button→profile created.



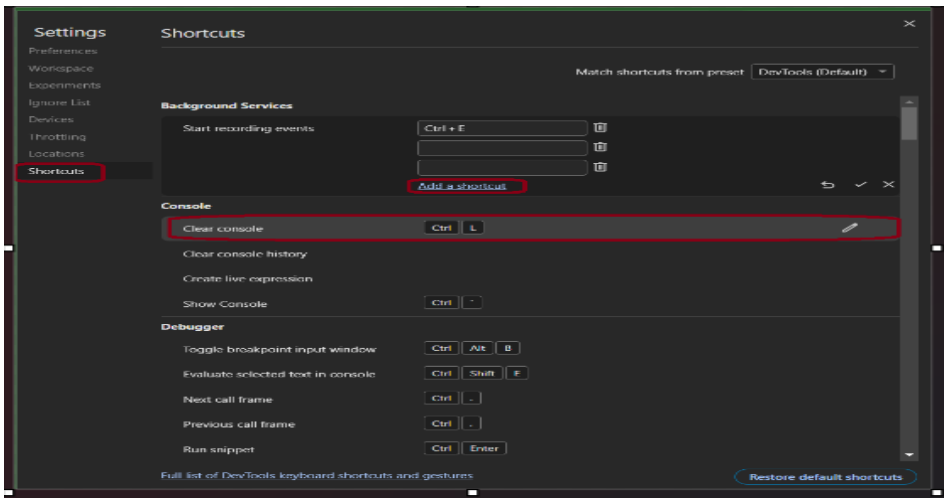
7. Locations

- Click dev setting→click on **location** tab→click add location button→enter the details→click add.



8. Shortcuts

- Click dev setting→**shortcuts**→you can add shortcuts using add shortcut button→OR default shortcuts you can use while focused in DevTools to speed up your workflow.



Submitted by: Manaswi M. Patil

CONCLUSION

Proficiency in using Chrome's Inspect feature for web development and debugging. Understanding of web performance metrics and optimization techniques. Skills in diagnosing and resolving common web development issues. Enhanced ability to work collaboratively in web development and debugging tasks. Chrome DevTools is an invaluable set of tools for web developers, offering a robust environment to inspect, debug, and optimize web pages. It provides a wide array of features, from real-time editing of HTML and CSS to comprehensive performance analysis. Developers can efficiently troubleshoot issues, test different scenarios, and enhance the overall quality of web applications. With its intuitive interface and powerful capabilities, DevTools empowers developers to create, maintain, and improve web experiences effectively. Regularly exploring and leveraging the functionalities within Chrome DevTools can significantly contribute to a smoother and more productive web development process.

Submitted by: Manaswi M. Patil