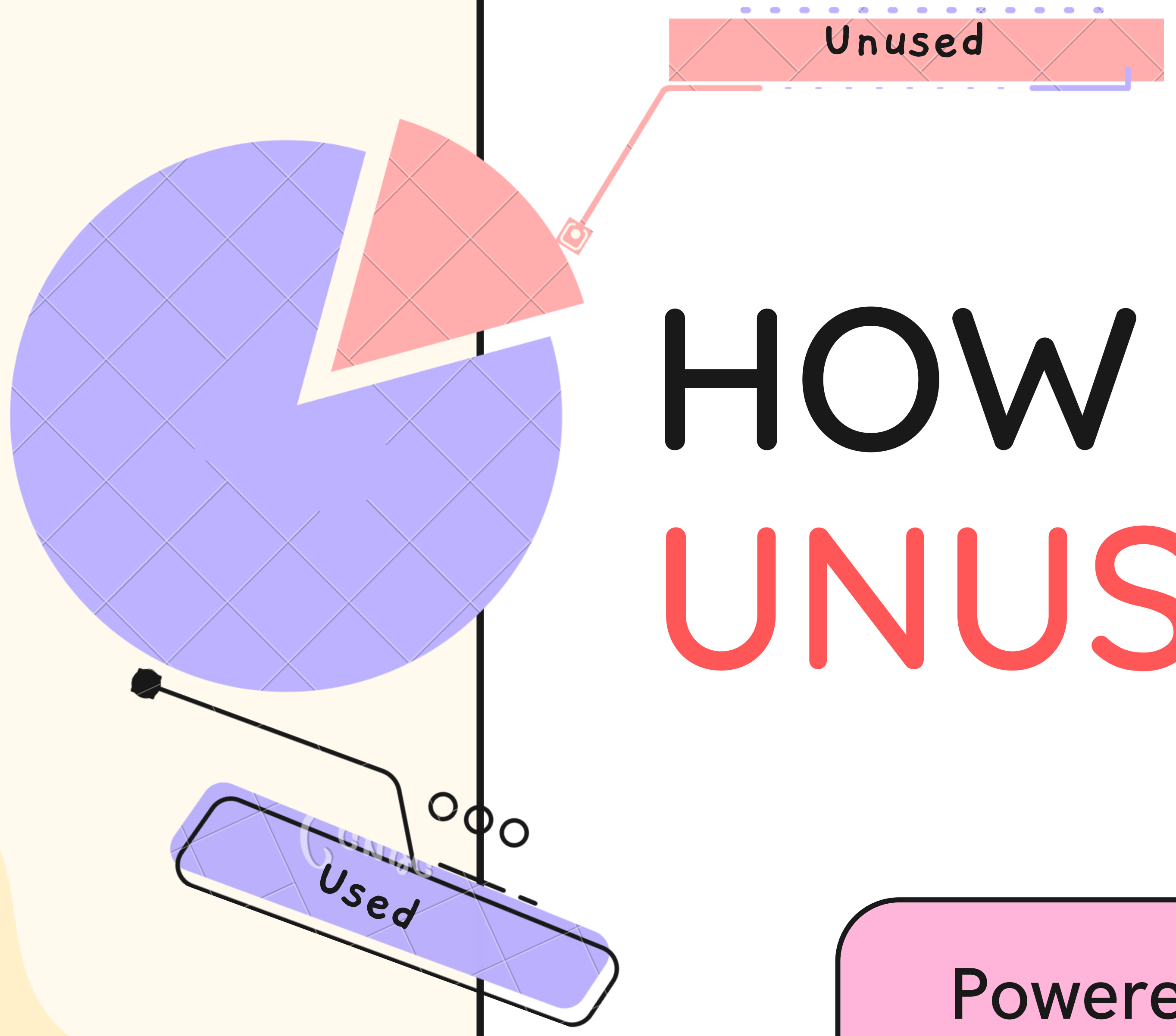
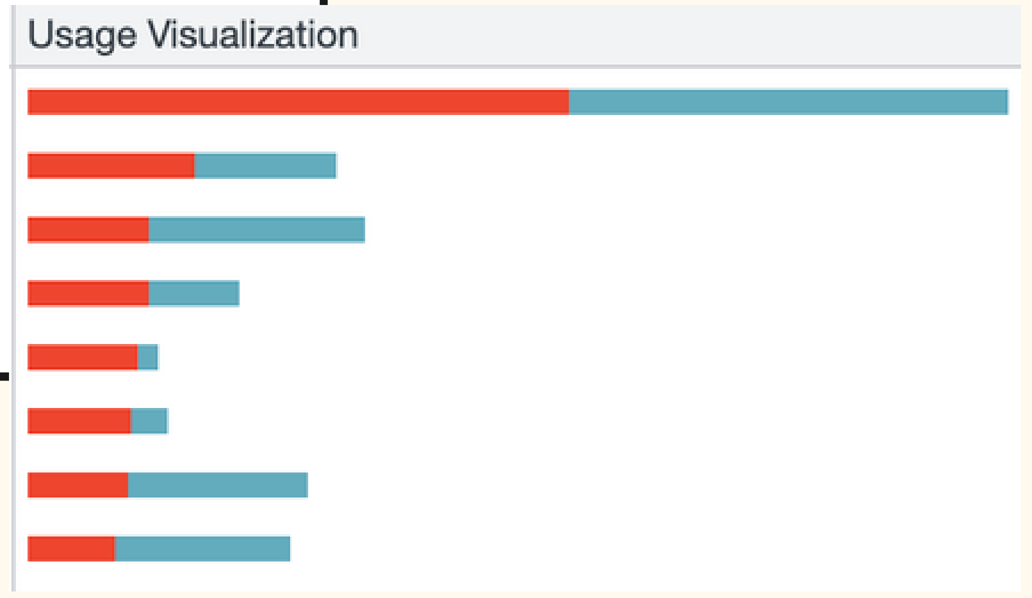


HOW TO FIND OUT UNUSED CSS & JS?

Powered by Google Chrome



WHY SHOULD YOU CARE ?

IMPROVED PERFORMANCE

By removing unused resources such as CSS and JS we can reduce the unnecessary overhead of loading and rendering those resources which helps optimize the critical rendering path of a web page resulting in improved performance and faster load times

REDUCED BANDWIDTH

Every byte of data transferred over the network consumes bandwidth, by eliminating unused resources we can reduce the file size and the number of network requests resulting in lower bandwidth usage which can be a huge benefit for users on slower networks or limited/expensive bandwidth availability

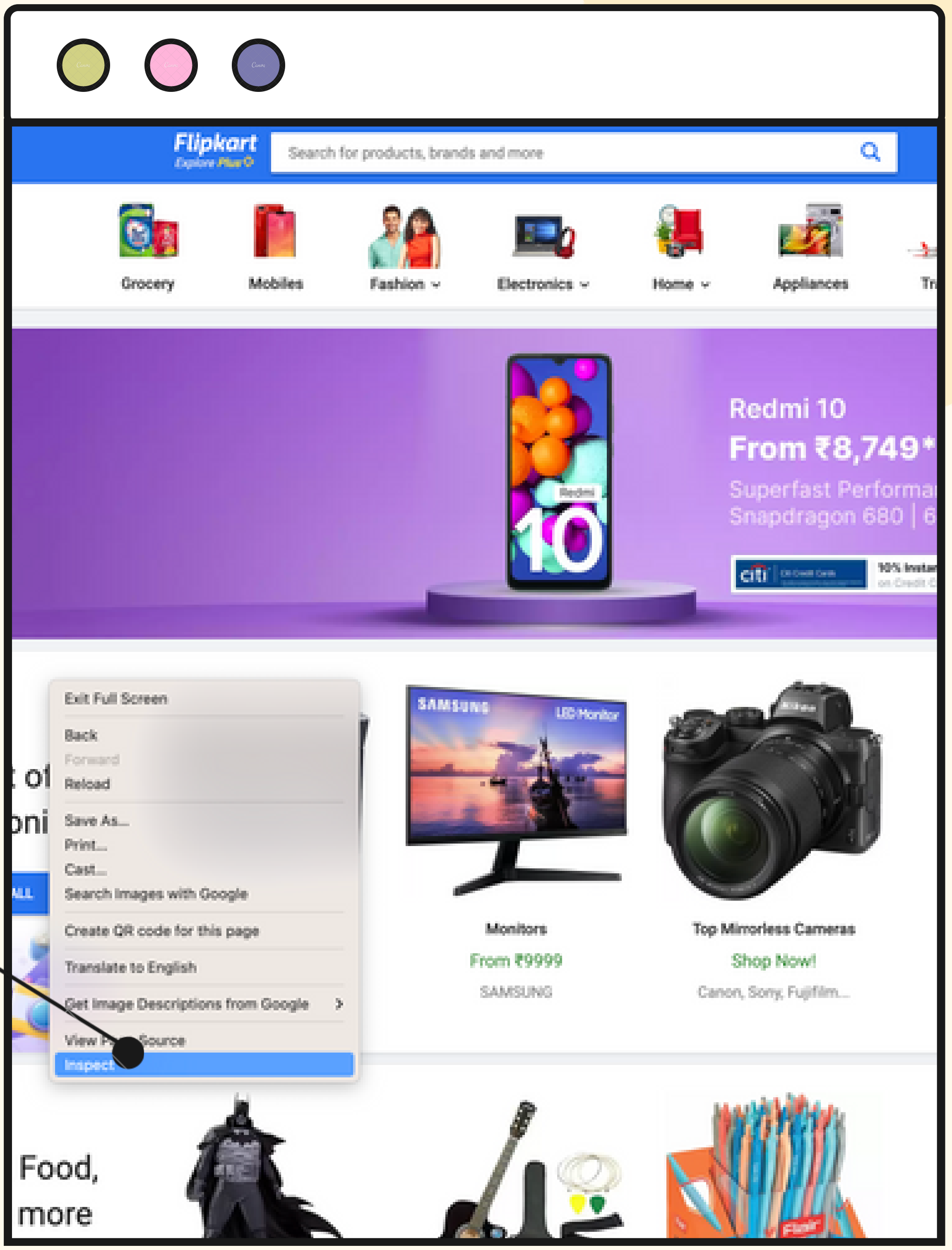
EASIER/FASTER DEBUGGING

Getting rid of unused code from production makes debugging any prod issue much easier and faster. With fewer lines of code to analyse developers can identify and resolve issues faster resulting in a more stable and reliable application.

HOW TO FIND OUT UNUSED RESOURCES USING CHROME DEVTOOLS ?

1 OPEN CHROME DEVELOPER TOOLS

1.1 Right-click on the webpage and select "Inspect."

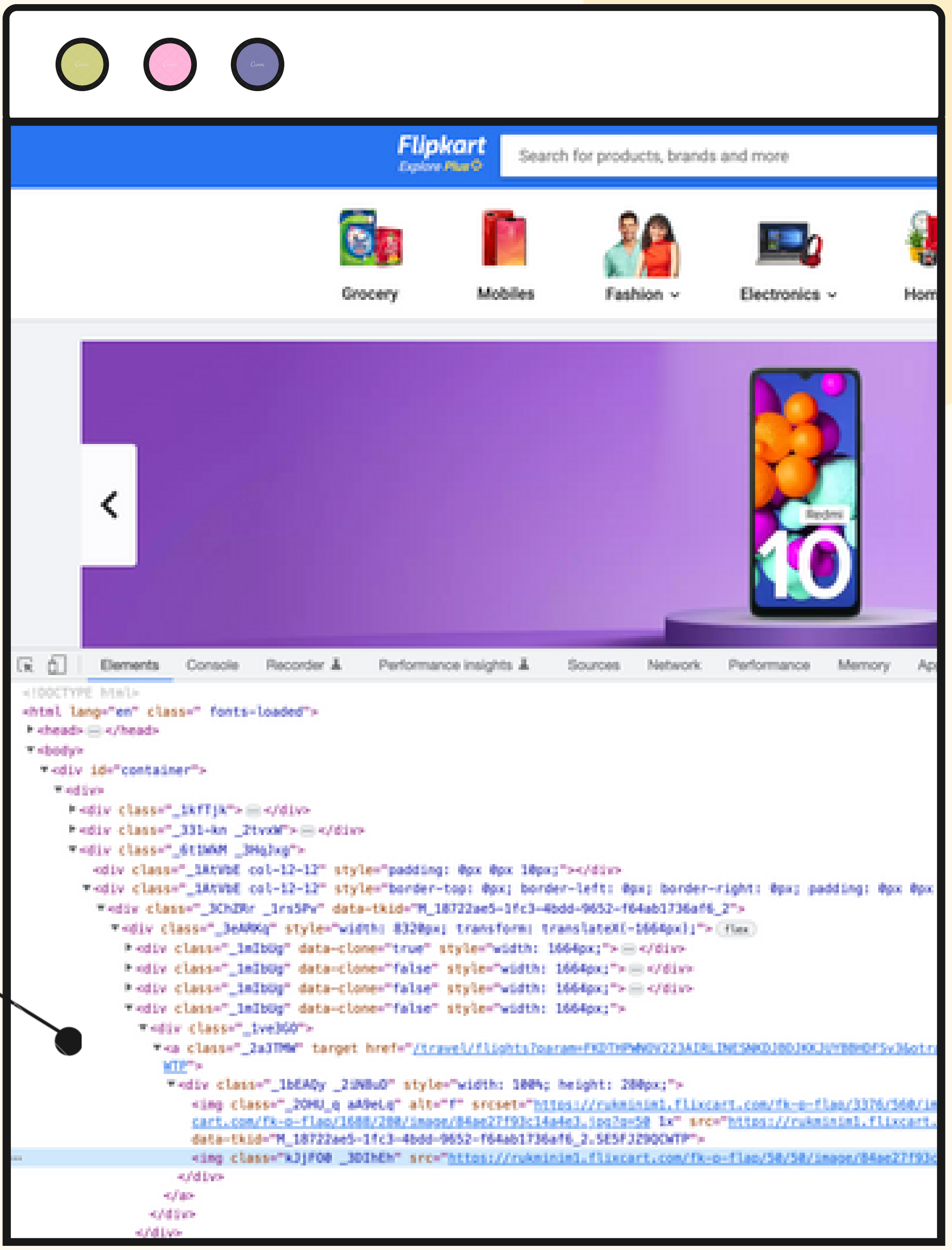


1

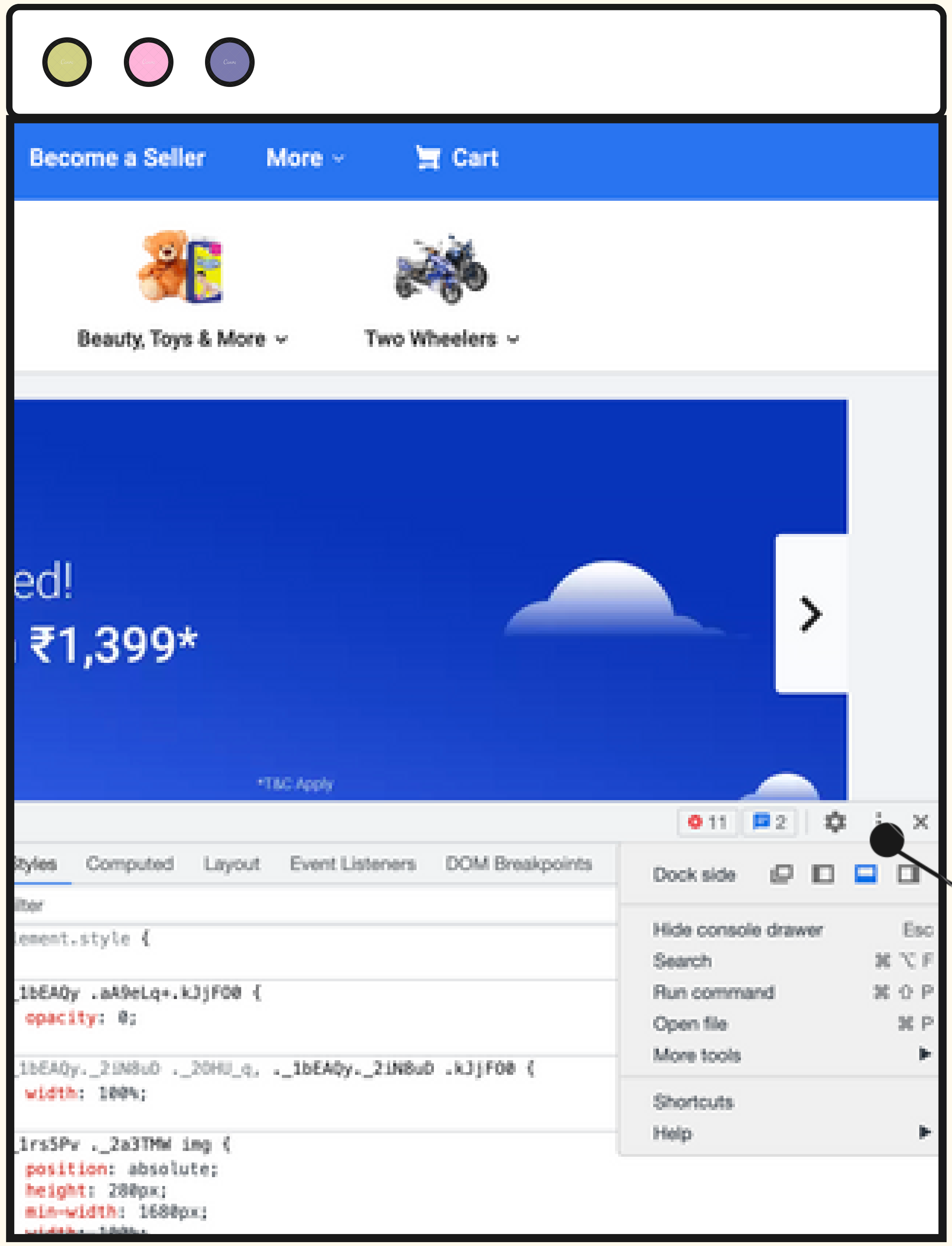
OPEN CHROME DEVELOPER TOOLS (CONT.)

1.2

The Chrome Developer Tools panel will open at the bottom or right side of the browser.

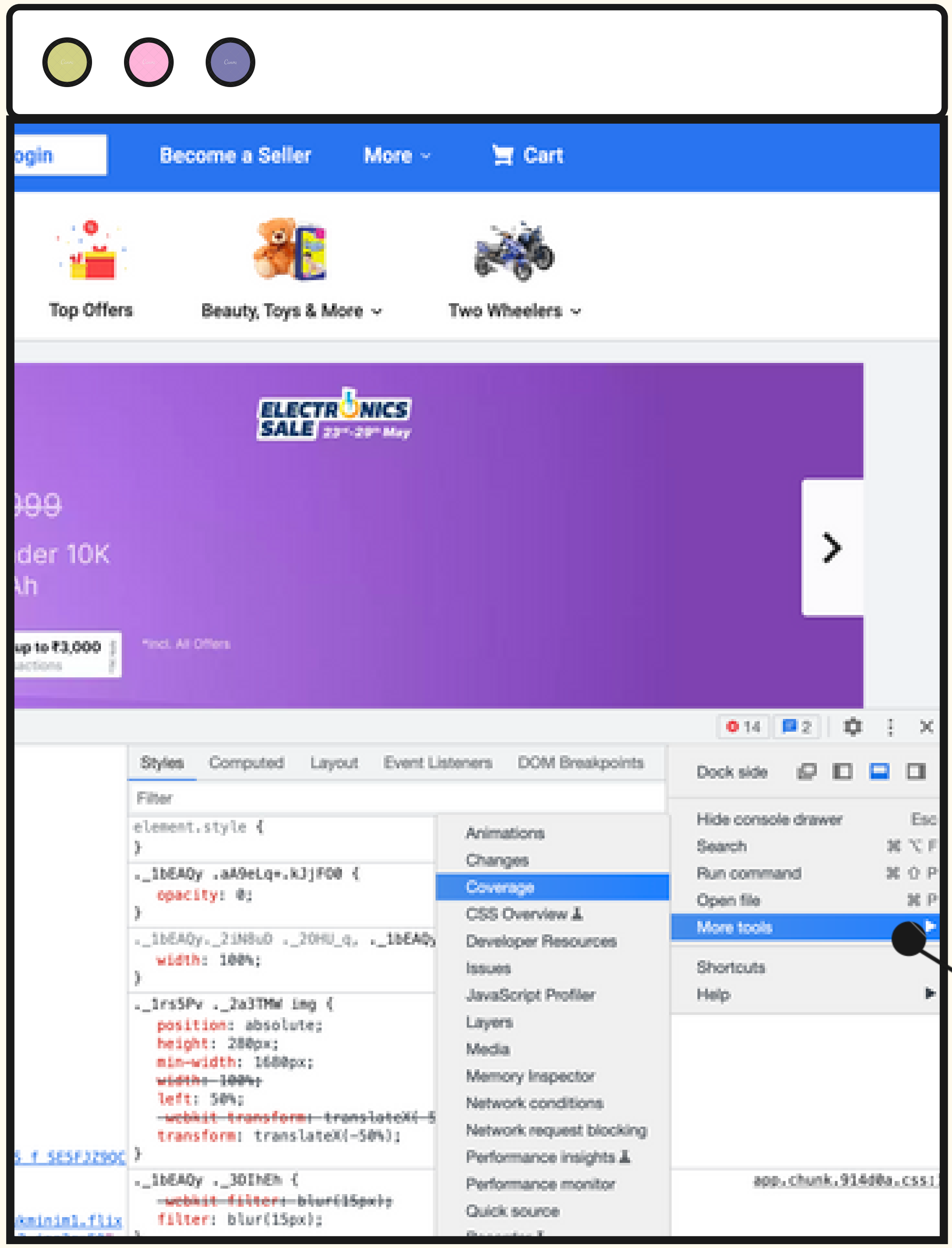


```
<!DOCTYPE html>
<html lang="en" class=" fonts-loaded">
 ▸<head> … </head>
 ▾<body>
   ▾<div id="container">
     ▾<div>
       ▸<div class="_1kfTjk"> … </div>
       ▸<div class="_331-kn _2tvxW"> … </div>
       ▾<div class="_6t1WkM _3HqJxg">
          <div class="_1AtVbE col-12-12" style="padding: 0px 0px 10px;"></div>
         ▾<div class="_1AtVbE col-12-12" style="border-top: 0px; border-left: 0px; border-right: 0px; padding: 0px 0px
           ▾<div class="_3ChZRr _1rs5Pv" data-tkid="M_18722ae5-1fc3-4bdd-9652-f64ab1736af6_2">
             ▾<div class="_3eARKq" style="width: 8320px; transform: translateX(-1664px);"> flex
               ▸<div class="_1mIbUg" data-clone="true" style="width: 1664px;"> … </div>
               ▸<div class="_1mIbUg" data-clone="false" style="width: 1664px;"> … </div>
               ▸<div class="_1mIbUg" data-clone="false" style="width: 1664px;"> … </div>
               ▾<div class="_1mIbUg" data-clone="false" style="width: 1664px;">
                 ▾<div class="_1ve3G0">
                   ▾<a class="_2a3TMW" target href="/travel/flights?param=[illegible]
                    WTP">
                     ▾<div class="_1bEAQy _2iNBuO" style="width: 100%; height: 280px;">
                        <img class="_2OHU_q aA9eLq" alt="f" srcset="https://rukminim1.flixcart.com/fk-p-flap/3376/560/im
                        cart.com/fk-p-flap/1688/280/image/84ae27f93c14a4e1.jpg?q=50 1x" src="https://rukminim1.flixcart.
                        data-tkid="M_18722ae5-1fc3-4bdd-9652-f64ab1736af6_2.5E5FJZ9QCWTP">
                        <img class="kJjFO0 _3DIhEh" src="https://rukminim1.flixcart.com/fk-p-flap/50/50/image/84ae27f93c
                      </div>
                    </a>
                  </div>
                </div>
```



2 OPEN THE COVERAGE TAB

2.1 Click on the 3-dot menu icon on Developer tools, a dropdown menu will open up.



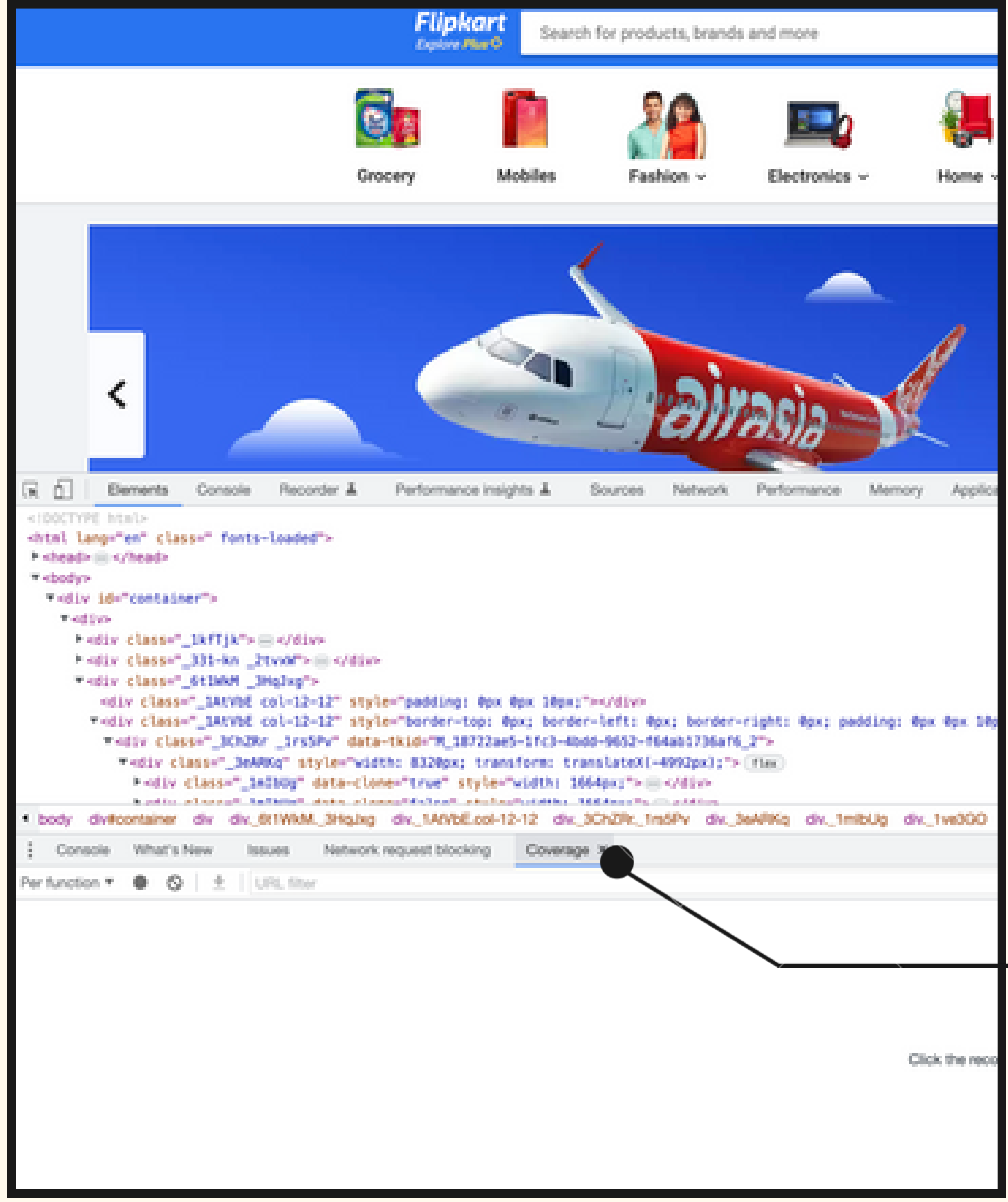
2

OPEN THE COVERAGE TAB (CONT.)

2.2

Click on the "More tools" option from the menu, another menu will open up, where we need to click on the "Coverage" option

2 OPEN THE COVERAGE TAB (CONT.)



```
<!DOCTYPE html>
<html lang="en" class=" fonts-loaded">
▸<head> … </head>
▾<body>
  ▾<div id="container">
    ▾<div>
      ▸<div class="_1kfTjk"> … </div>
      ▸<div class="_331-kn _2tvxW"> … </div>
      ▾<div class="_6t1WkM _3HqJxg">
         <div class="_1AtVbE col-12-12" style="padding: 0px 0px 10px;"></div>
        ▾<div class="_1AtVbE col-12-12" style="border-top: 0px; border-left: 0px; border-right: 0px; padding: 0px 0px 10
          ▾<div class="_3ChZRr _1rs5Pv" data-tkid="M_18722ae5-1fc3-4bdd-9652-f64ab1736af6_2">
            ▾<div class="_3eARKq" style="width: 8320px; transform: translateX(-4992px);"> flex
              ▸<div class="_1mIbUg" data-clone="true" style="width: 1664px;"> … </div>
```

body div#container div div._6t1WkM._3HqJxg div._1AtVbE.col-12-12 div._3ChZRr._1rs5Pv div._3eARKq div._1mIbUg div._1ve3GO

Click the reco

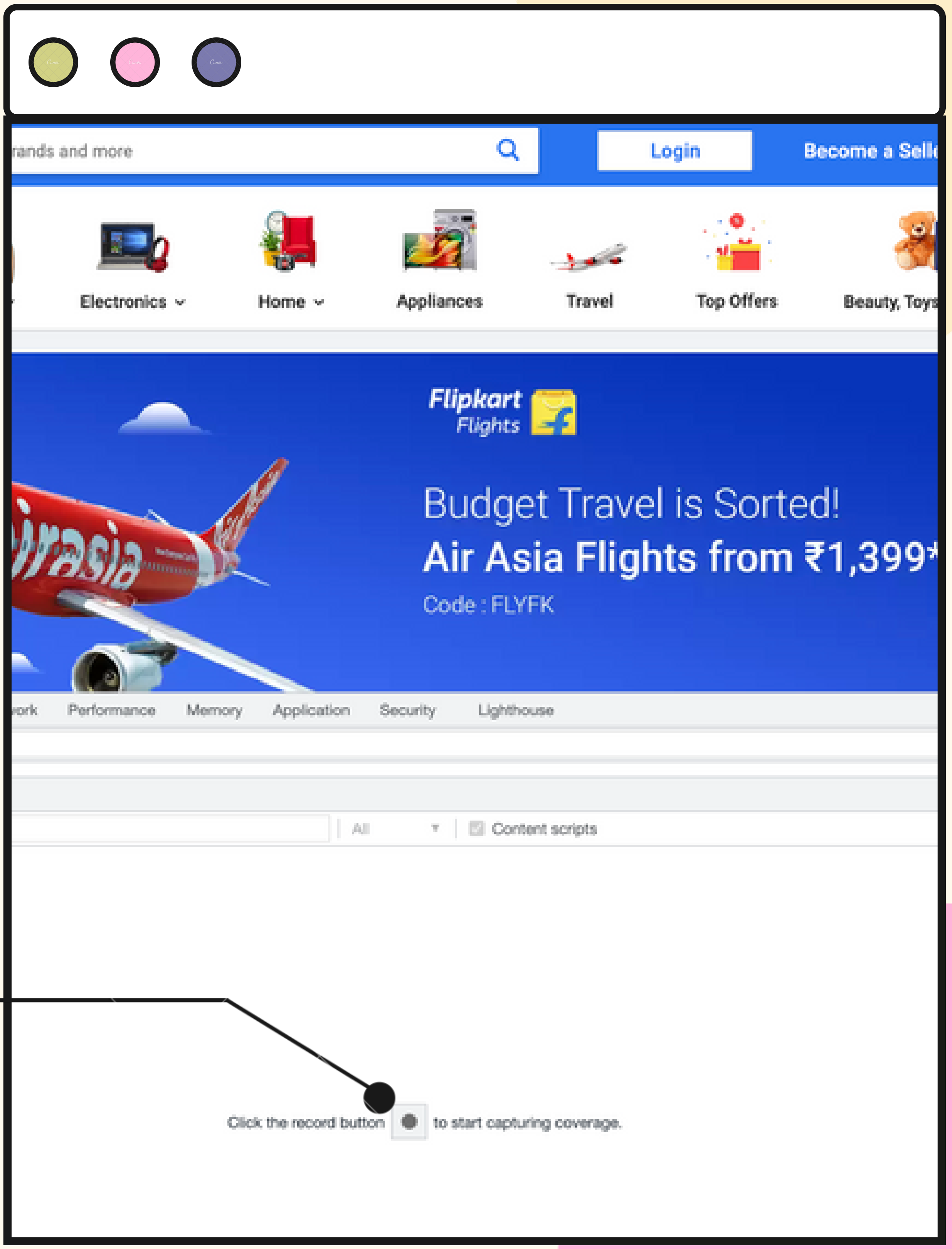
2.3

A new tab for Coverage will show up in the Developers tools panel

3 ANALYSE CODE COVERAGE

3.1

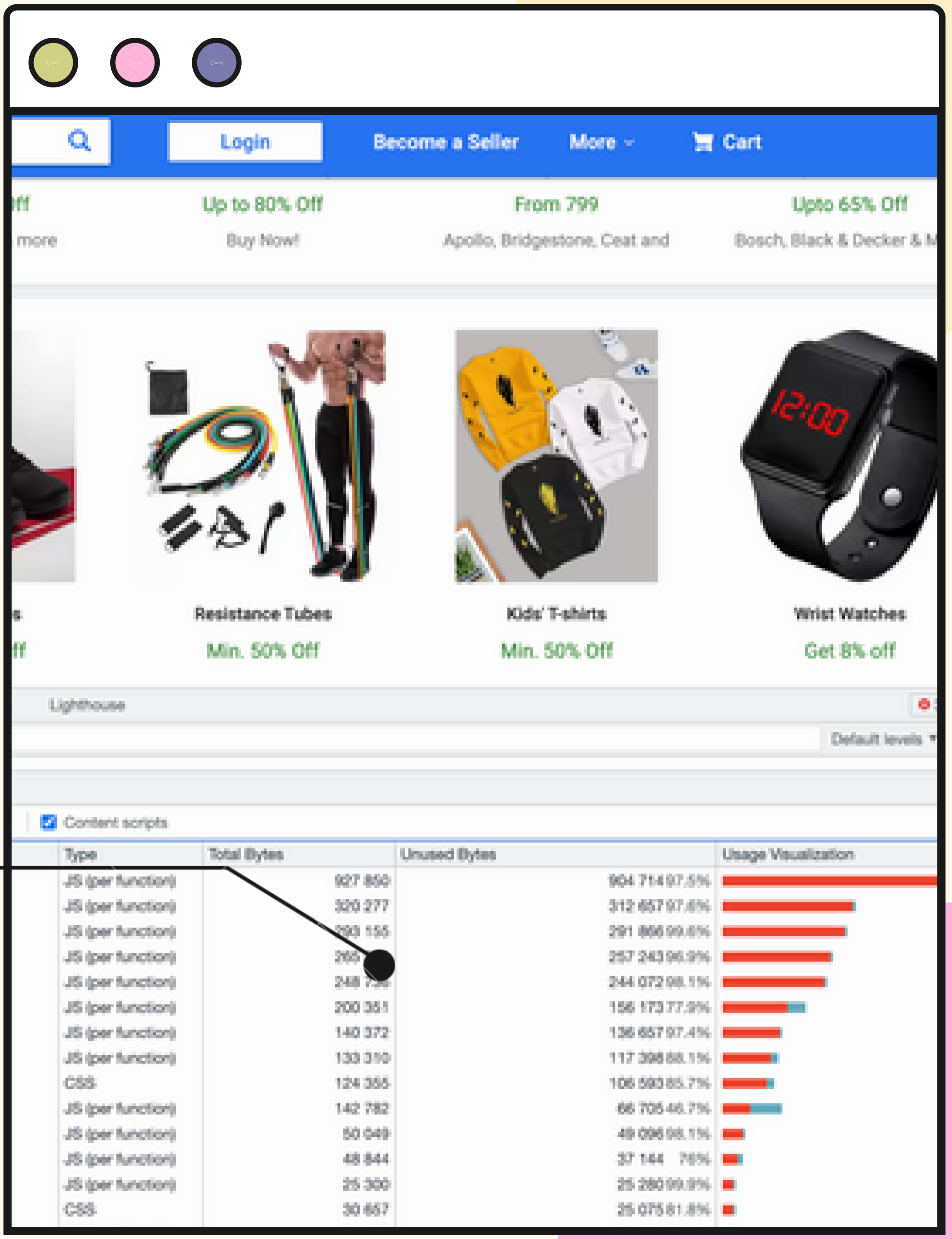
We can now start the analysis for unused code by clicking on the record button on coverage tab



Click the record button ● to start capturing coverage.

3 ANALYSE CODE COVERAGE (CONT.)

3.2 Once the recording has started, we can start interacting with the application and the coverage tab will start showing up the resources along with its recorded usage

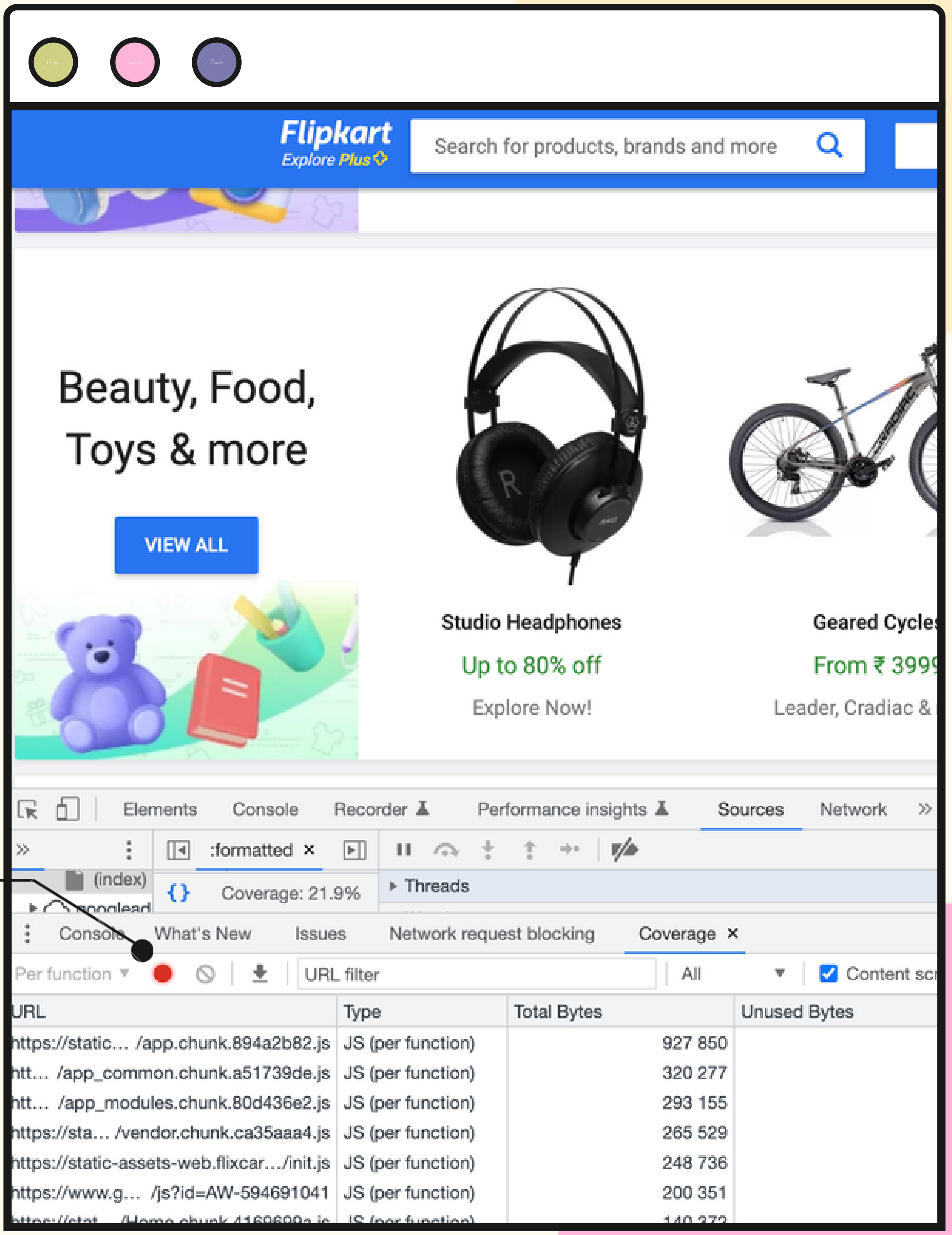


Type	Total Bytes	Unused Bytes	Usage Visualization
JS (per function)	927 850	904 714 97.5%	🟥
JS (per function)	320 277	312 657 97.6%	🟥
JS (per function)	293 155	291 866 99.6%	🟥
JS (per function)	265 [illegible]	257 243 96.9%	🟥
JS (per function)	248 [illegible]	244 072 98.1%	🟥
JS (per function)	200 351	156 173 77.9%	🟥
JS (per function)	140 372	136 657 97.4%	🟥
JS (per function)	133 310	117 398 88.1%	🟥
CSS	124 355	106 593 85.7%	🟥
JS (per function)	142 782	66 705 46.7%	🟥
JS (per function)	50 049	49 096 98.1%	🟥
JS (per function)	48 844	37 144 76%	🟥
JS (per function)	25 300	25 280 99.9%	🟥
CSS	30 657	25 075 81.8%	🟥

3 ANALYSE CODE COVERAGE (CONT.)

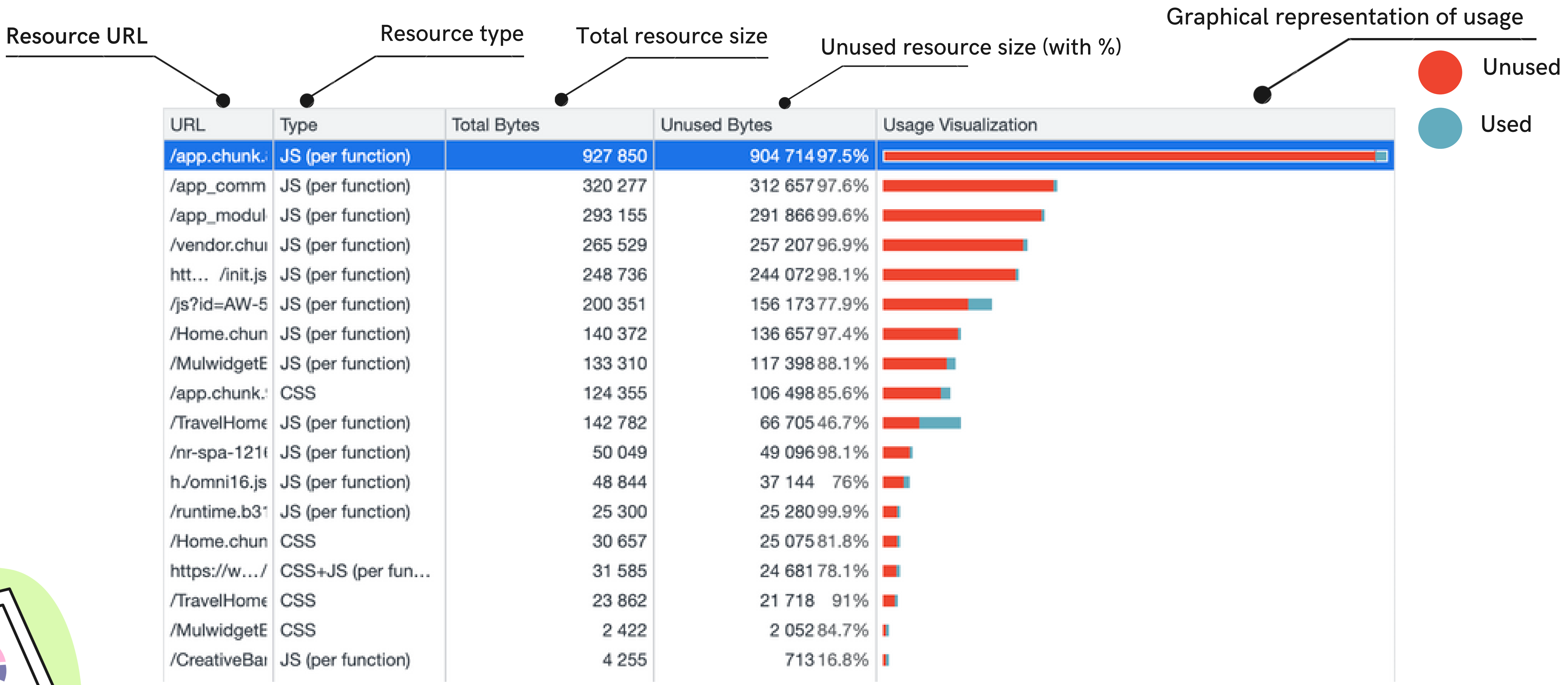
3.3

Click on the Stop button in the Coverage tab to stop tracking code coverage



URL	Type	Total Bytes	Unused Bytes
https://static… /app.chunk.894a2b82.js	JS (per function)	927 850	
htt… /app_common.chunk.a51739de.js	JS (per function)	320 277	
htt… /app_modules.chunk.80d436e2.js	JS (per function)	293 155	
https://sta… /vendor.chunk.ca35aaa4.js	JS (per function)	265 529	
https://static-assets-web.flixcar…/init.js	JS (per function)	248 736	
https://www.g… /js?id=AW-594691041	JS (per function)	200 351	

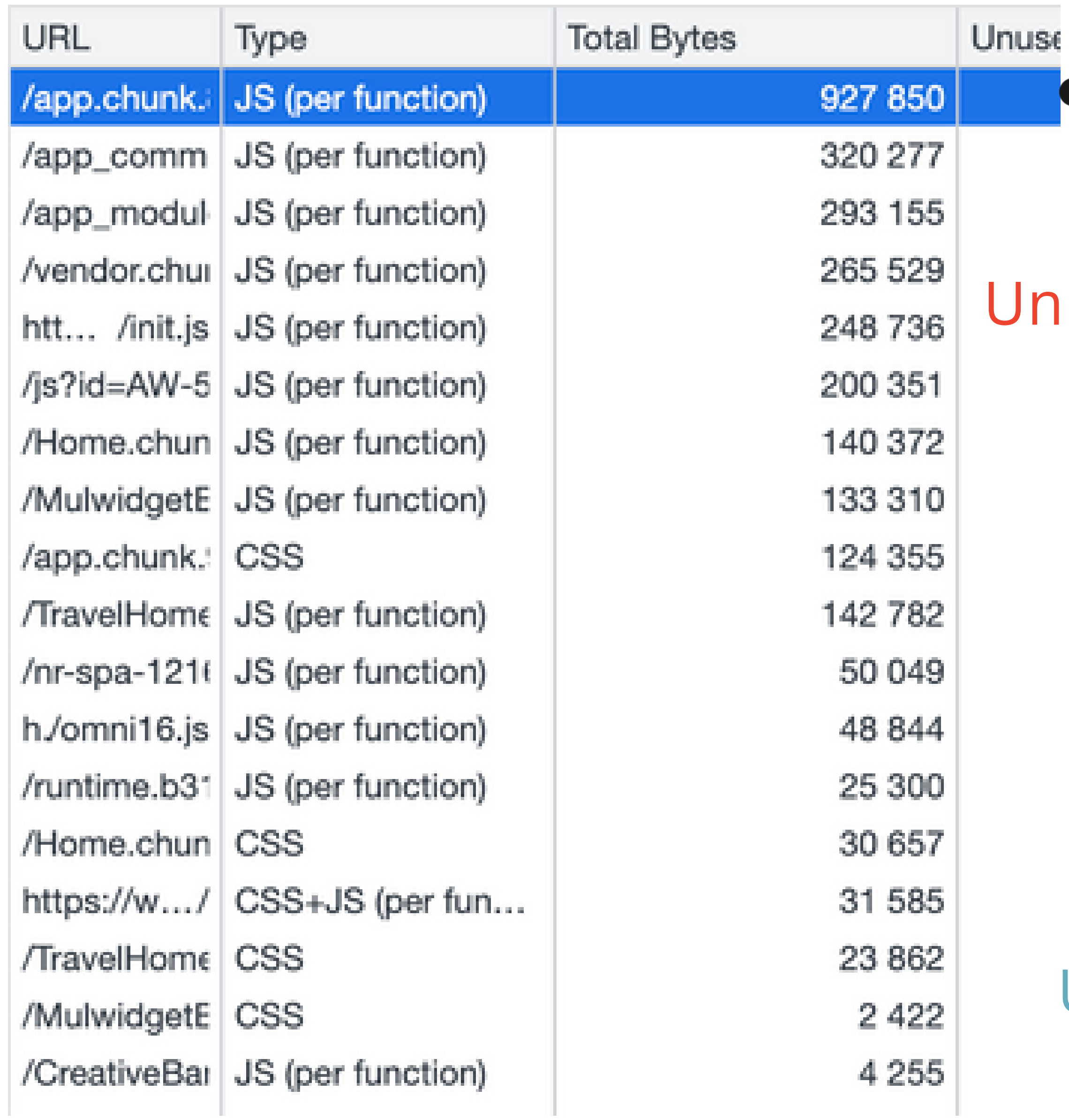
UNDERSTANDING THE COVERAGE TABLE


Resource URL
Resource type
Total resource size
Unused resource size (with %)
Graphical representation of usage
● Unused
● Used

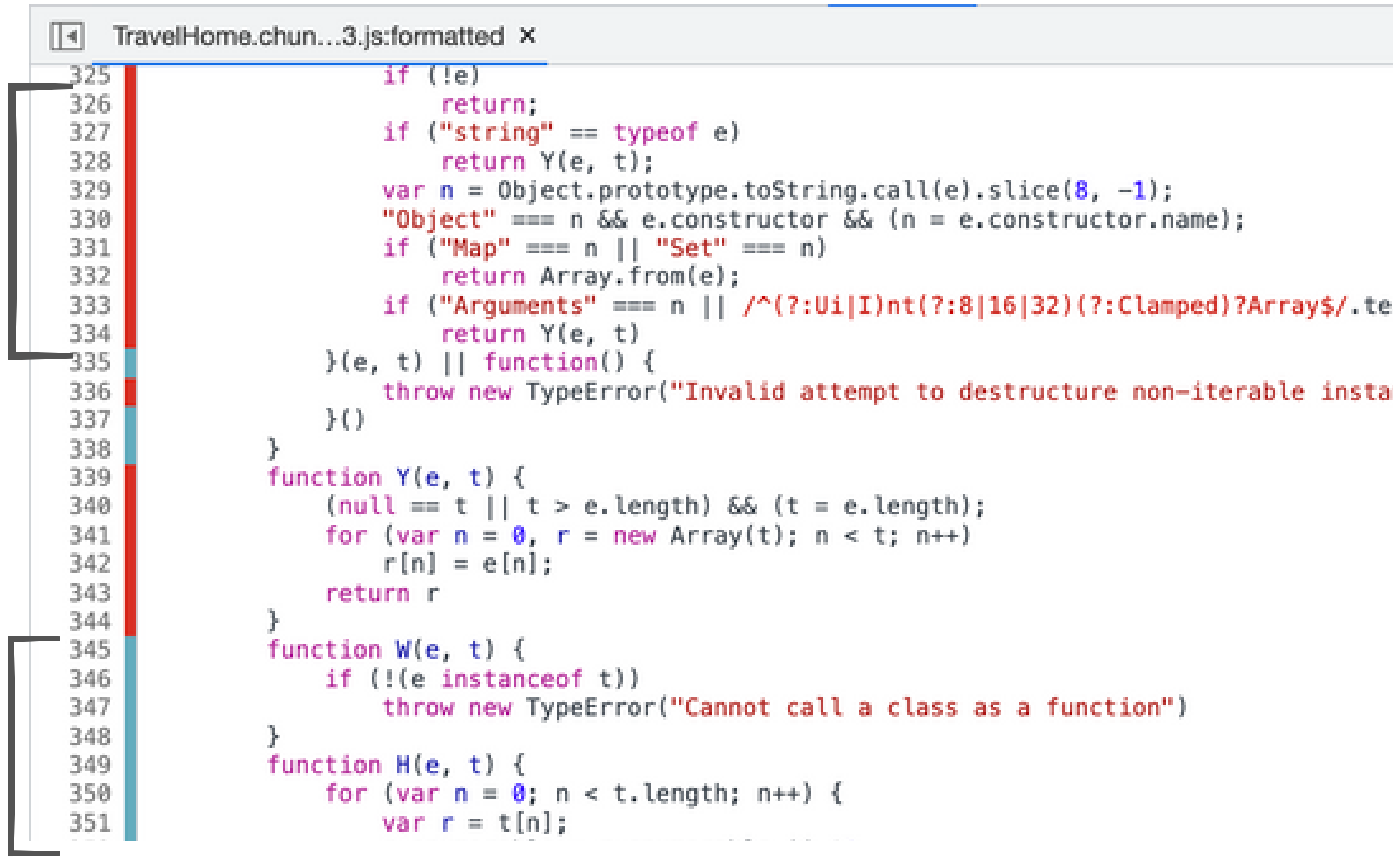
URL	Type	Total Bytes	Unused Bytes		Usage Visualization
/app.chunk.:	JS (per function)	927 850	904 714	97.5%	
/app_comm	JS (per function)	320 277	312 657	97.6%	
/app_modul	JS (per function)	293 155	291 866	99.6%	
/vendor.chu	JS (per function)	265 529	257 207	96.9%	
htt… /init.js	JS (per function)	248 736	244 072	98.1%	
/js?id=AW-5	JS (per function)	200 351	156 173	77.9%	
/Home.chun	JS (per function)	140 372	136 657	97.4%	
/MulwidgetE	JS (per function)	133 310	117 398	88.1%	
/app.chunk.:	CSS	124 355	106 498	85.6%	
/TravelHome	JS (per function)	142 782	66 705	46.7%	
/nr-spa-121(	JS (per function)	50 049	49 096	98.1%	
h./omni16.js	JS (per function)	48 844	37 144	76%	
/runtime.b3'	JS (per function)	25 300	25 280	99.9%	
/Home.chun	CSS	30 657	25 075	81.8%	
https://w…/	CSS+JS (per fun…	31 585	24 681	78.1%	
/TravelHome	CSS	23 862	21 718	91%	
/MulwidgetE	CSS	2 422	2 052	84.7%	
/CreativeBar	JS (per function)	4 255	713	16.8%	

UNDERSTANDING THE COVERAGE TABLE

Clicking on any resource will show the line-by-line coverage details

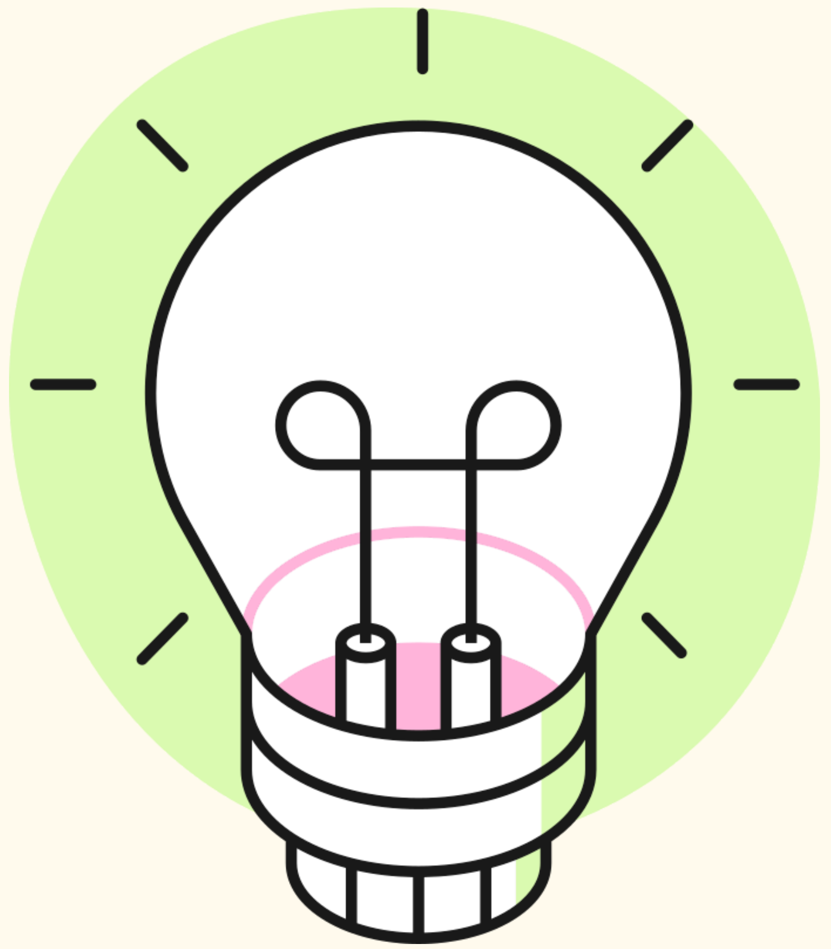


URL	Type	Total Bytes	Unuse
/app.chunk.	JS (per function)	927 850	
/app_comm	JS (per function)	320 277	
/app_modul	JS (per function)	293 155	
/vendor.chu	JS (per function)	265 529	
htt... /init.js	JS (per function)	248 736	
/js?id=AW-5	JS (per function)	200 351	
/Home.chun	JS (per function)	140 372	
/MulwidgetE	JS (per function)	133 310	
/app.chunk.	CSS	124 355	
/TravelHome	JS (per function)	142 782	
/nr-spa-121	JS (per function)	50 049	
h./omni16.js	JS (per function)	48 844	
/runtime.b3	JS (per function)	25 300	
/Home.chun	CSS	30 657	
https://w.../	CSS+JS (per fun...	31 585	
/TravelHome	CSS	23 862	
/MulwidgetE	CSS	2 422	
/CreativeBa	JS (per function)	4 255	



```
    if (!e)
      return;
    if ("string" == typeof e)
      return Y(e, t);
    var n = Object.prototype.toString.call(e).slice(8, -1);
    "Object" === n && e.constructor && (n = e.constructor.name);
    if ("Map" === n || "Set" === n)
      return Array.from(e);
    if ("Arguments" === n || /^(?:Ui|I)nt(?:8|16|32)(?:Clamped)?Array$/.te
      return Y(e, t)
  }(e, t) || function() {
    throw new TypeError("Invalid attempt to destructure non-iterable insta
  }()
}
function Y(e, t) {
  (null == t || t > e.length) && (t = e.length);
  for (var n = 0, r = new Array(t); n < t; n++)
    r[n] = e[n];
  return r
}
function W(e, t) {
  if (!(e instanceof t))
    throw new TypeError("Cannot call a class as a function")
}
function H(e, t) {
  for (var n = 0; n < t.length; n++) {
    var r = t[n];
```

NOTE



It's important to note that when code is flagged as unused during coverage recording, it doesn't necessarily mean it will never be used. It simply indicates that it was not utilized during the specific period of coverage analysis. Consider an if-else block as an example: Only the block with a true condition gets executed, resulting in the remaining blocks being marked as unused. However, this doesn't render the other blocks unnecessary throughout the entire lifespan of the application.

Thank you!

learn,
share &
follow.