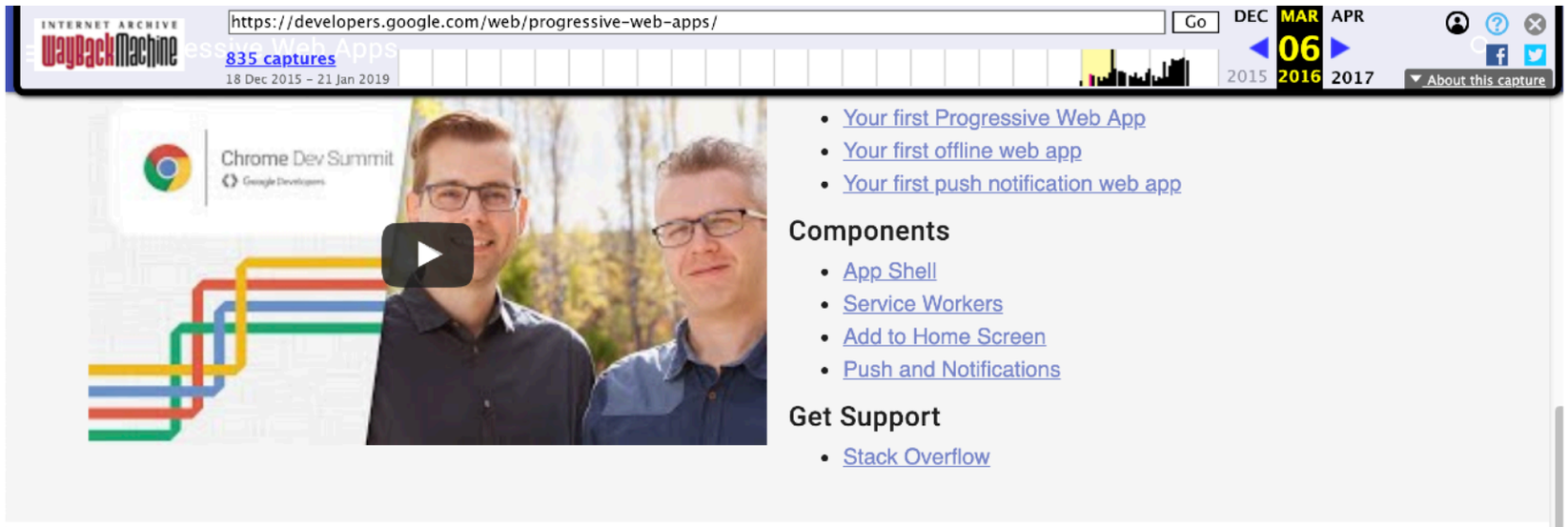
De Voorhoede

Progressive web apps

Declan Rek

WHAT IS A PWA

2016



What is a Progressive Web App?

Progressive Web Apps are experiences that combine the best of the web and the best of apps. They are useful to users from the very first visit in a browser tab, no install required. As the user progressively builds a relationship with the App over time, it becomes more and more powerful. It loads quickly, even on flaky networks, sends relevant push notifications, has an icon on the home screen and loads as top-level, full screen experience.

Progressive Web Apps are:

- **Progressive** - Work for every user, regardless of browser choice because they're built with progressive enhancement as a core tenet.
- **Responsive** - Fit any form factor: desktop, mobile, tablet, or whatever is next.
- **Connectivity independent** - Enhanced with service workers to work offline or on low quality networks.
- **App-like** - Feel like an app to the user with app-style interactions and navigation because it's built on the app shell model.
- **Fresh** - Always up-to-date thanks to the service worker update process.
- **Safe** - Served via HTTPS to prevent snooping and ensure content hasn't been tampered with.
- **Discoverable** - Are identifiable as "applications" thanks to W3C manifests and service worker registration scope allowing search engines to find them.
- **Re-engageable** - Make re-engagement easy through features like push notifications.
- **Installable** - Allow users to "keep" apps they find most useful on their home screen without the hassle of an app store.
- **Linkable** - Easily share via URL and not require complex installation.

2016... LATER

that performs as well as our native app. We now feel we have the best of both worlds.
- Amar Nagaram, an Engineering Director at Flipkart.

Instant Loading

Service workers allow your apps to load nearly instantly and reliably, no matter what kind of network connection your user is on.

Add to Home Screen

Web app install banners give you the ability to let your users quickly and seamlessly add your web app to their home screen, making it easy to launch and return to your app.

Push Notifications

Web push notifications makes it easy to re-engage with users by showing relevant, timely, and contextual notifications, even when the browser is closed.

Fast

Smooth animations, scrolling, and navigations keep the experience silky smooth.

Secure

HTTPS secures the connection between you and your users, ensuring your users information is protected and isn't tampered with.

Responsive

Modern users live on phones, tablets and laptops; your apps and websites should do the same. Learn how to structure and code your content to look great on screens of any size.

NOW

Progressive Web Apps

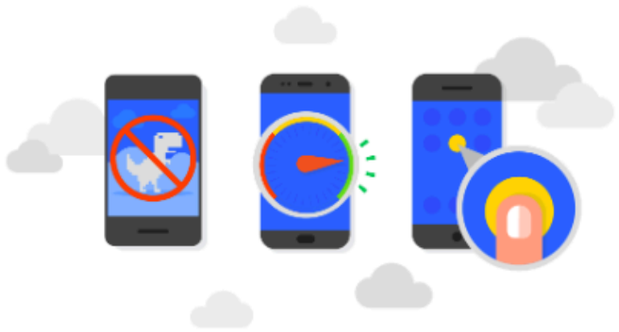
A new way to deliver amazing user experiences on the web.

Don't miss out on the action at this year's Chrome Dev Summit, watch live now!

Progressive Web Apps are user experiences that have the reach of the web, and are:

- **Reliable** - Load instantly and never show the downasaur, even in uncertain network conditions.
- **Fast** - Respond quickly to user interactions with silky smooth animations and no janky scrolling.
- **Engaging** - Feel like a natural app on the device, with an immersive user experience.

This new level of of quality allows Progressive Web Apps to earn a place on the users home screen.



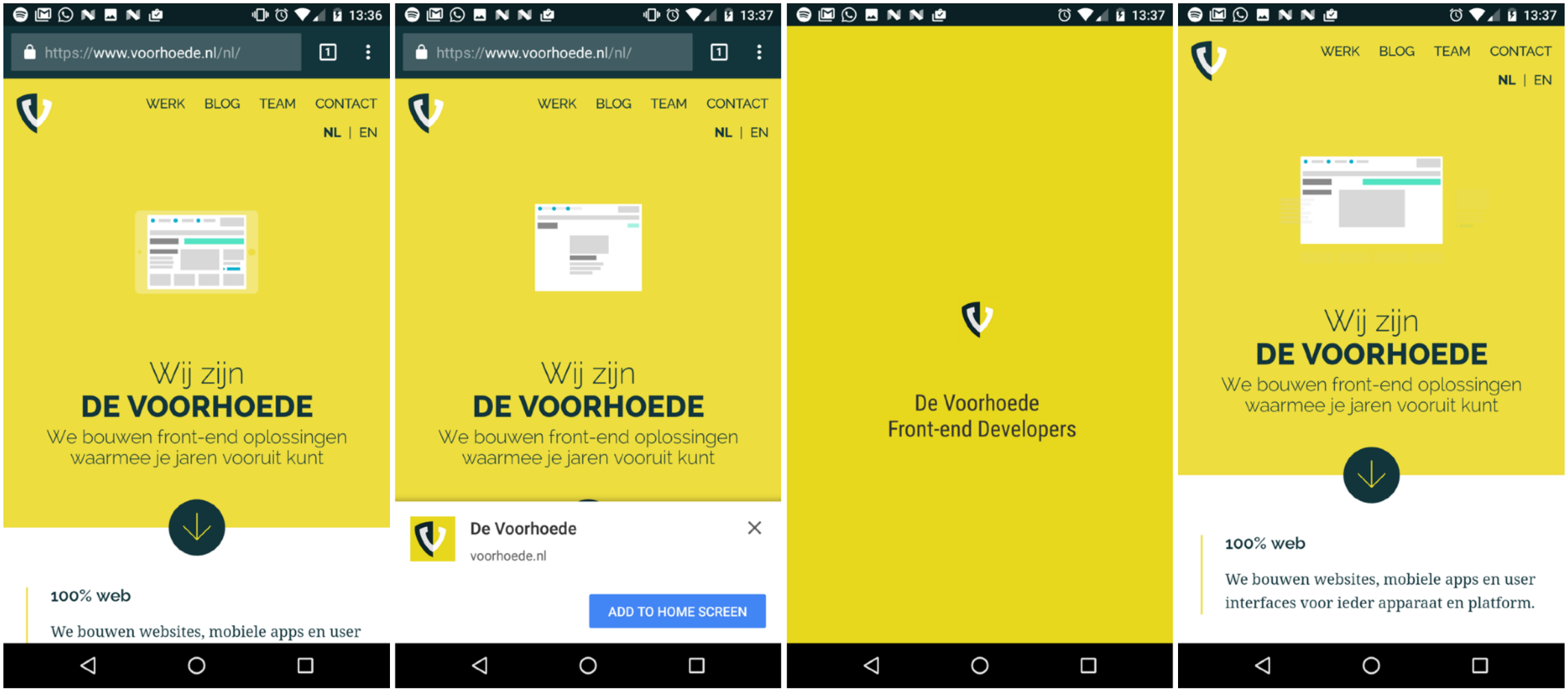
Progressive Web Apps are user experiences that have the **reach of the web**

HTTPS

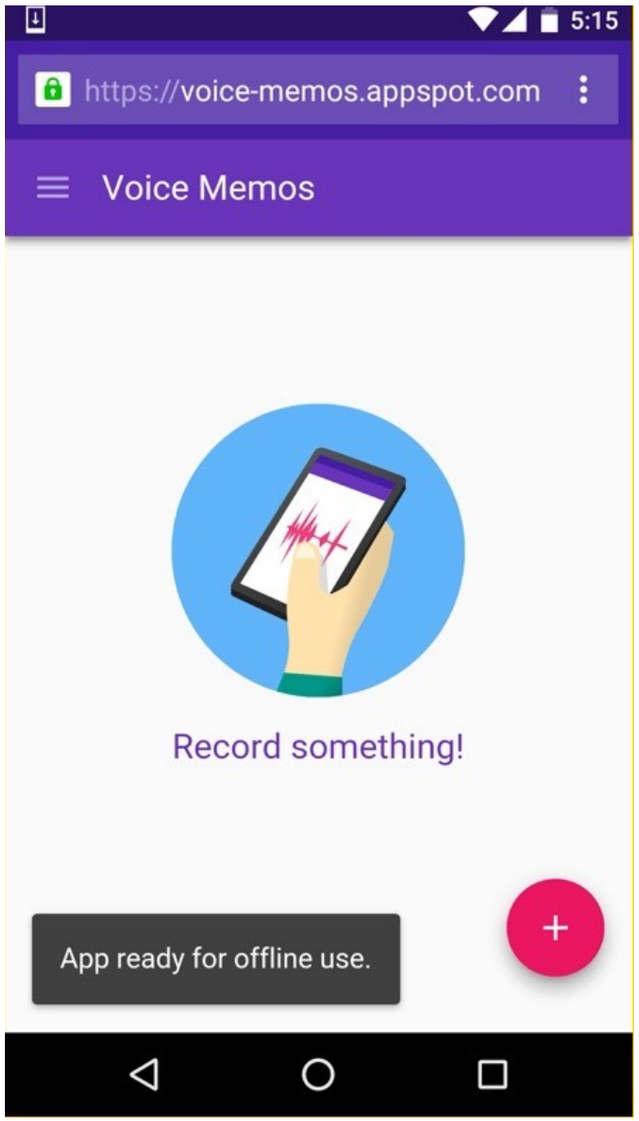
Service worker

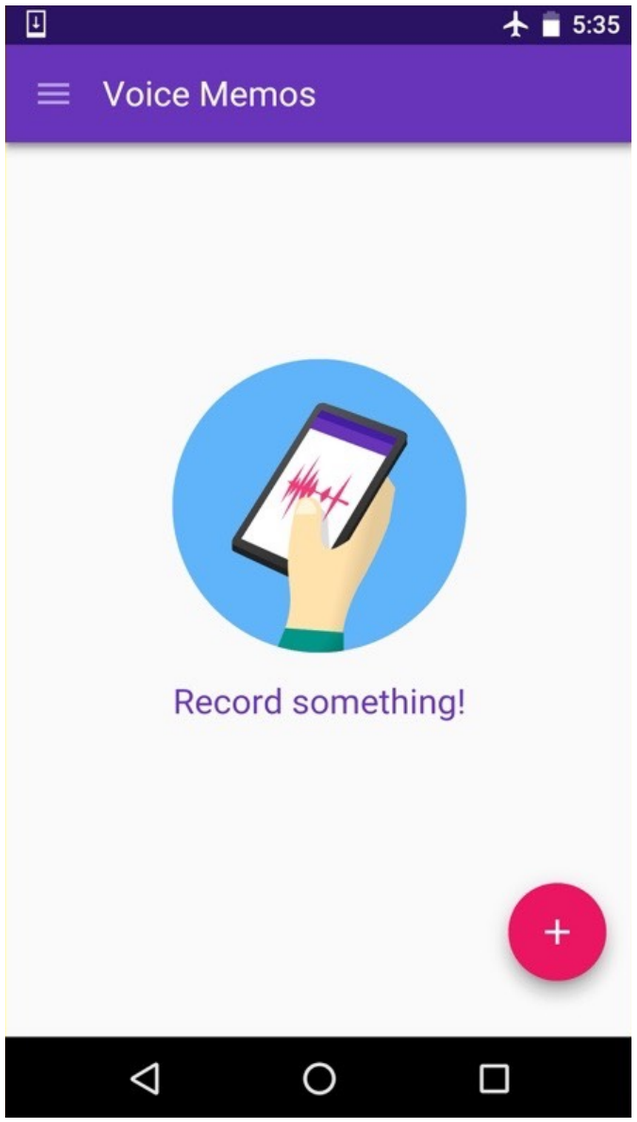
Manifest file

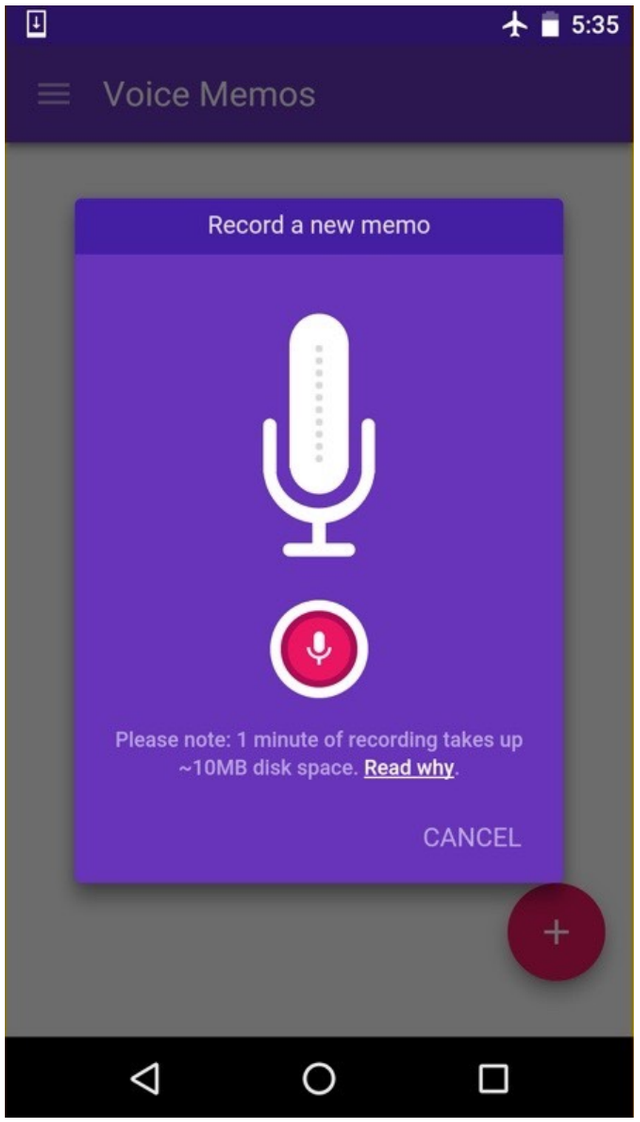
ADD TO HOMESTEEN



OFFLINE SUPPORT

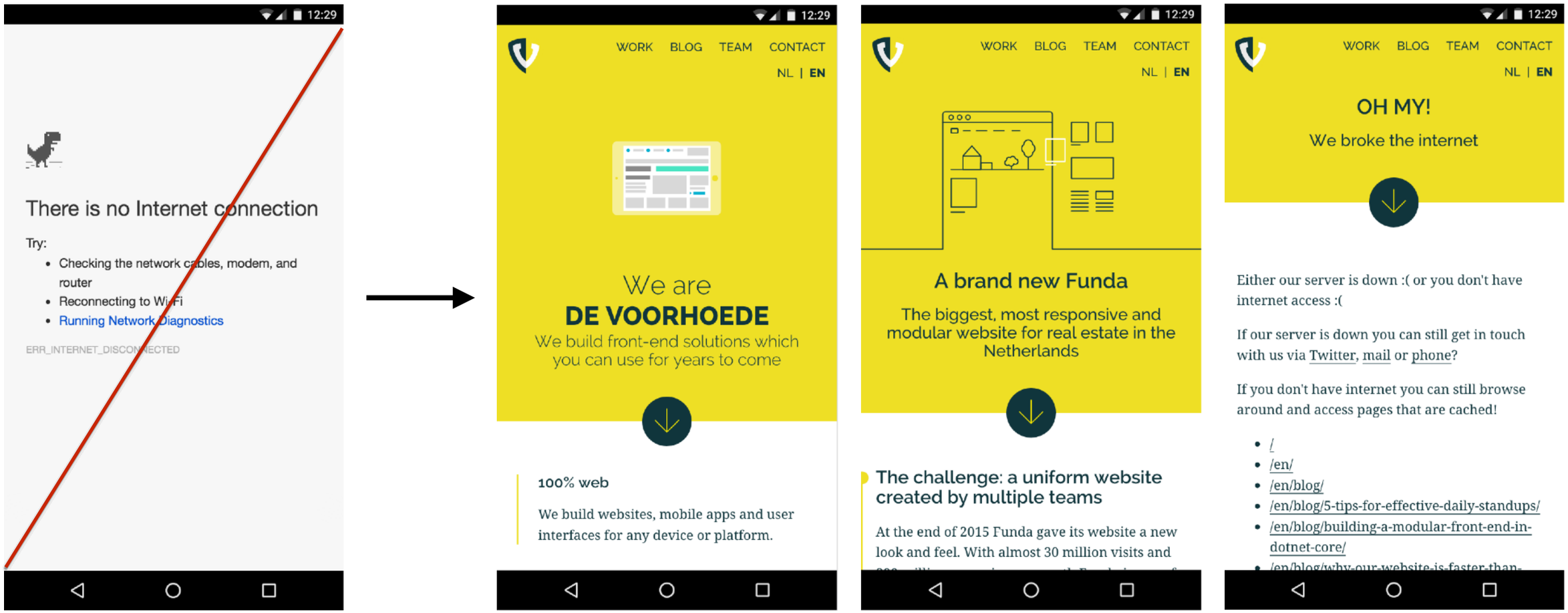




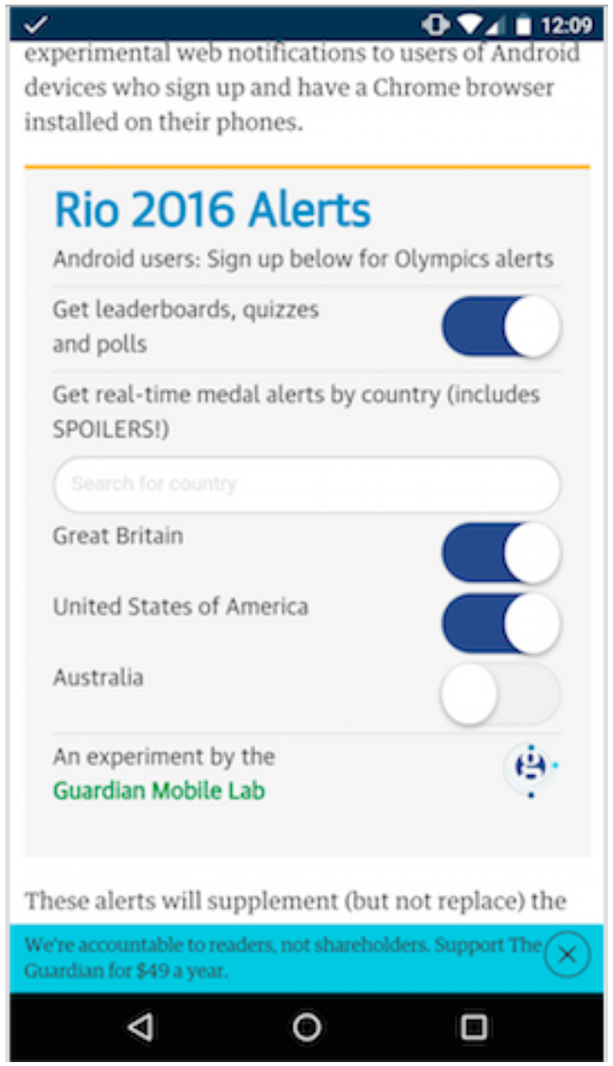


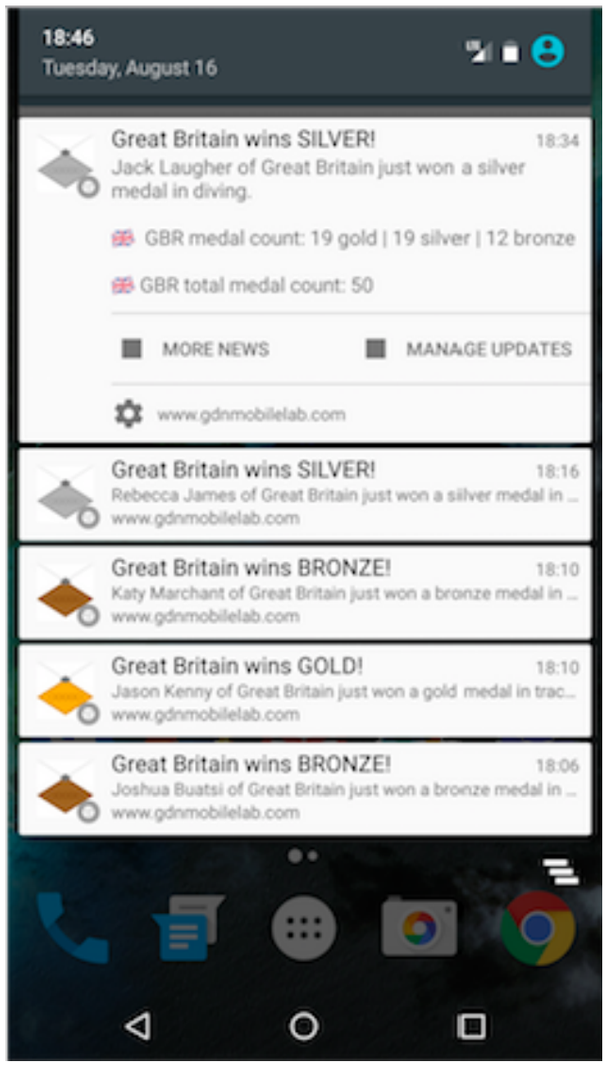
developers.google.com/web

OFFLINE SUPPORT



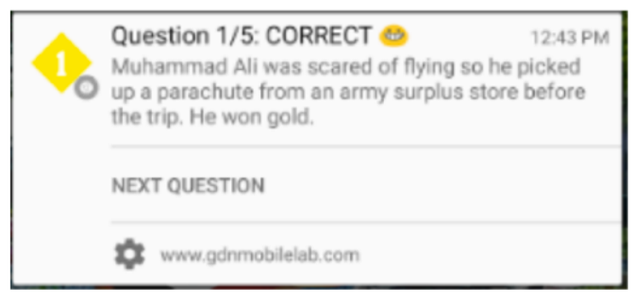
PUSH NOTIFICATIONS





Users tapped the action button to start the quiz

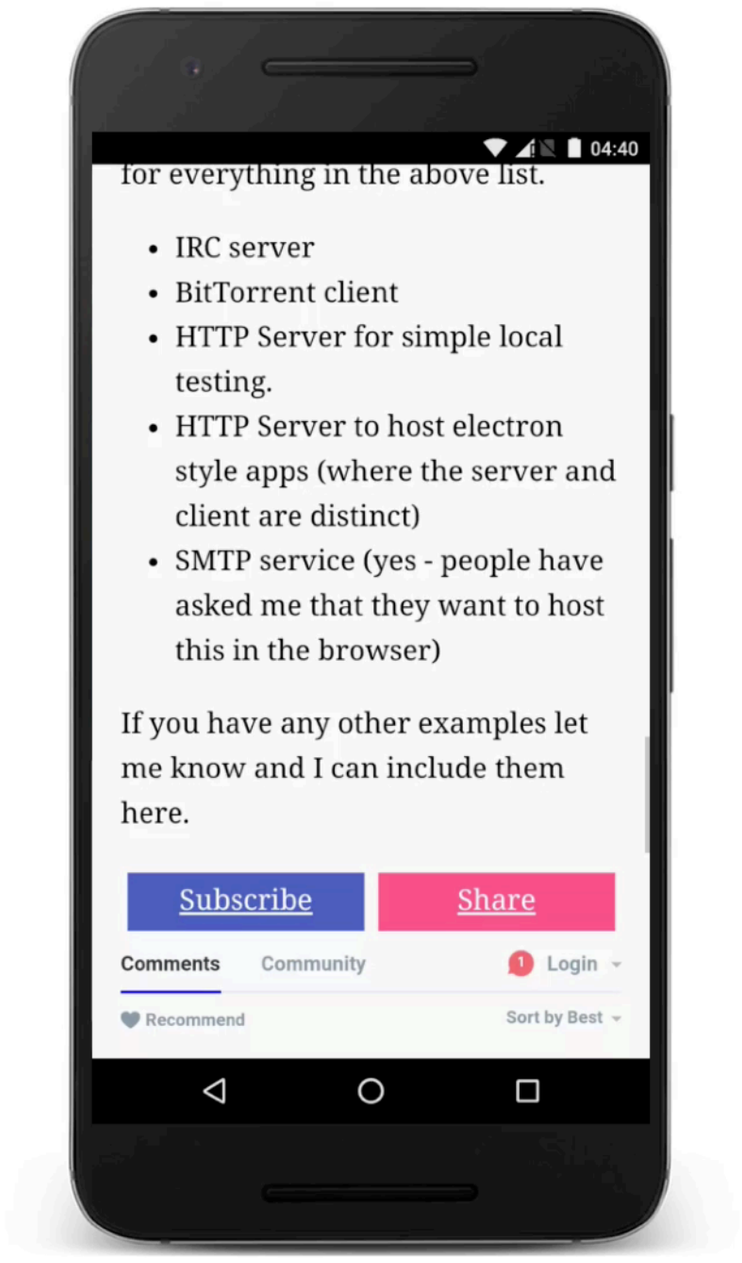
Users tapped on the action button with their answer



The answer was revealed before going to the next question

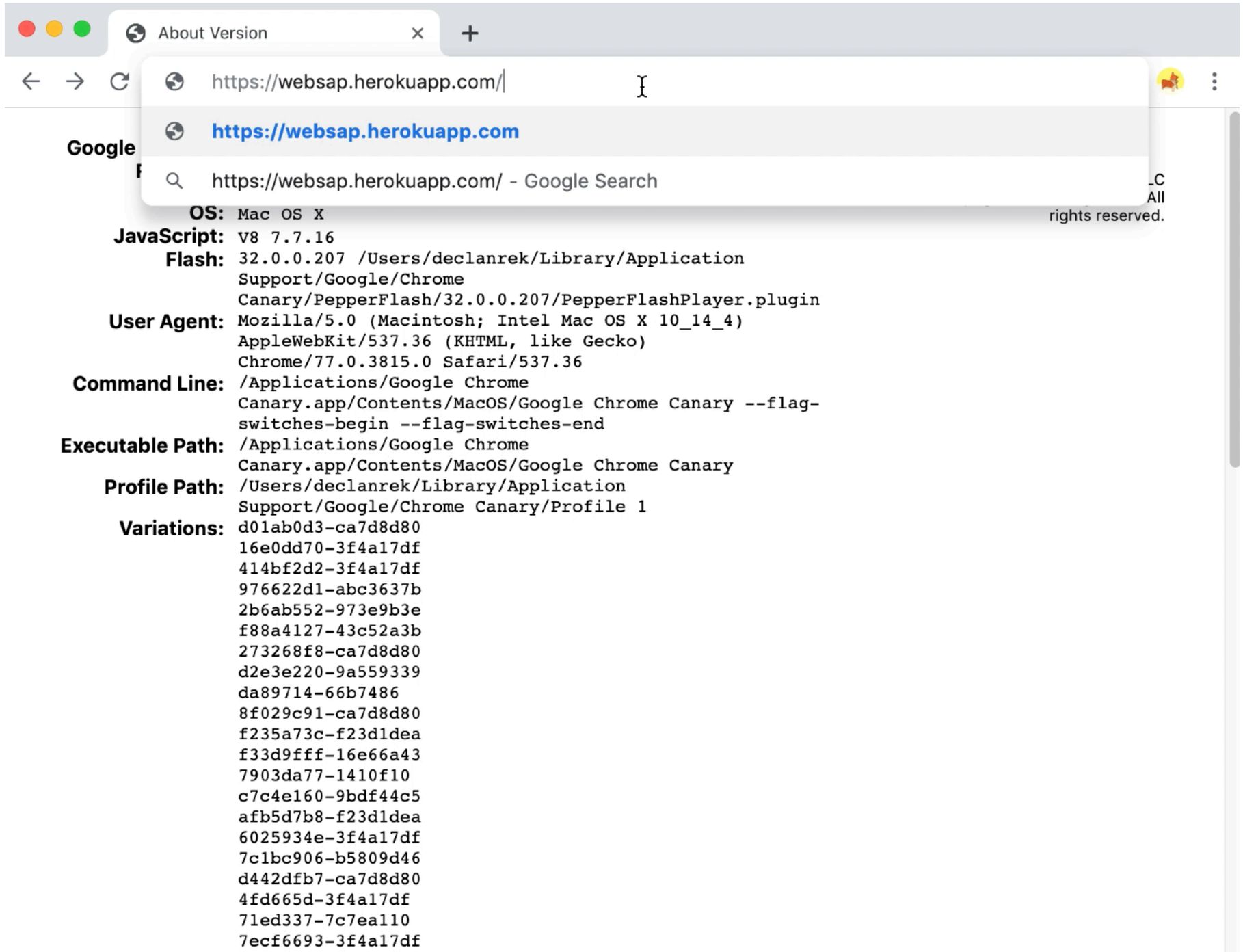
medium.com/the-guardian-mobile-innovation-lab/

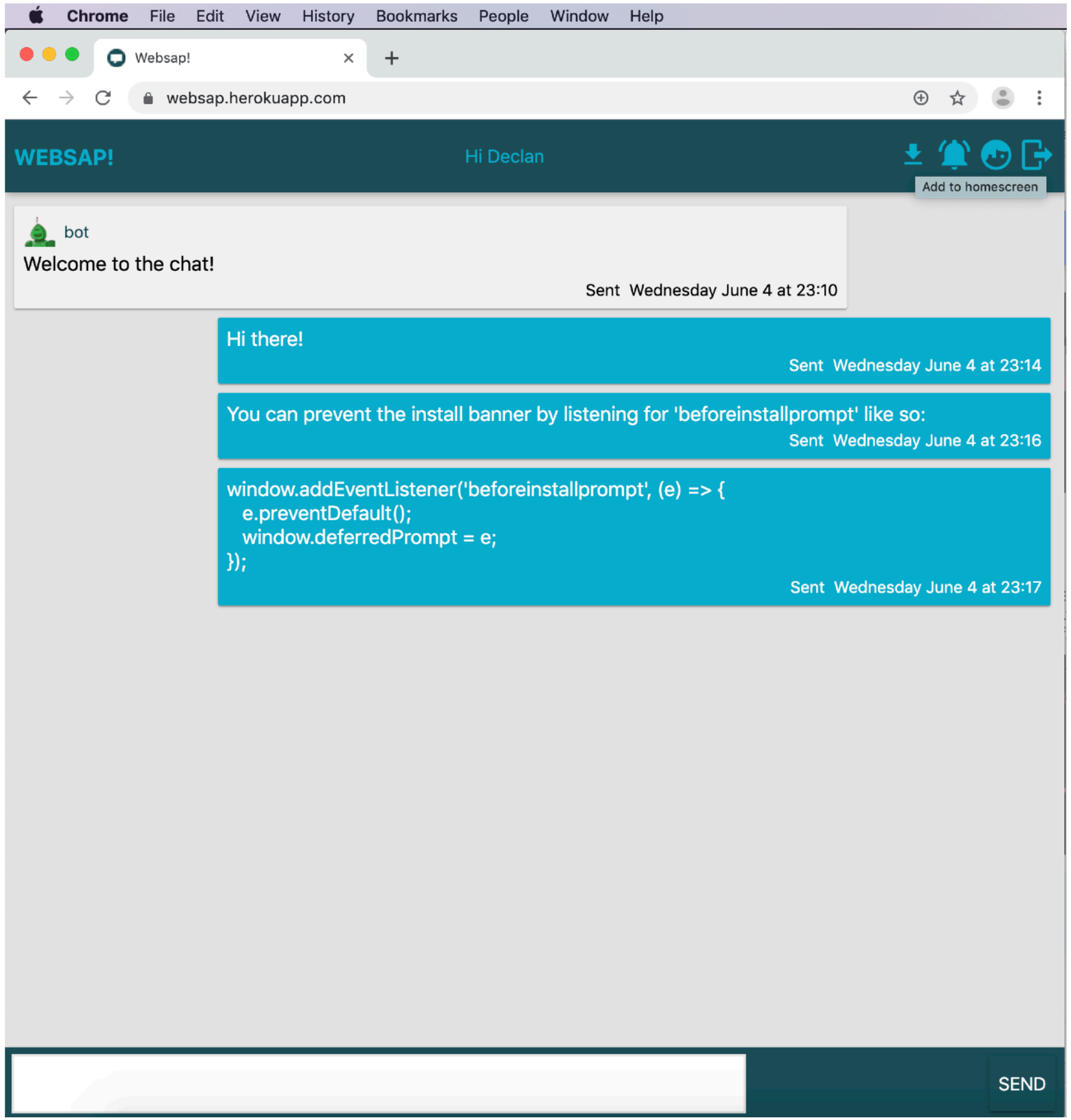
NATIVE SHARE

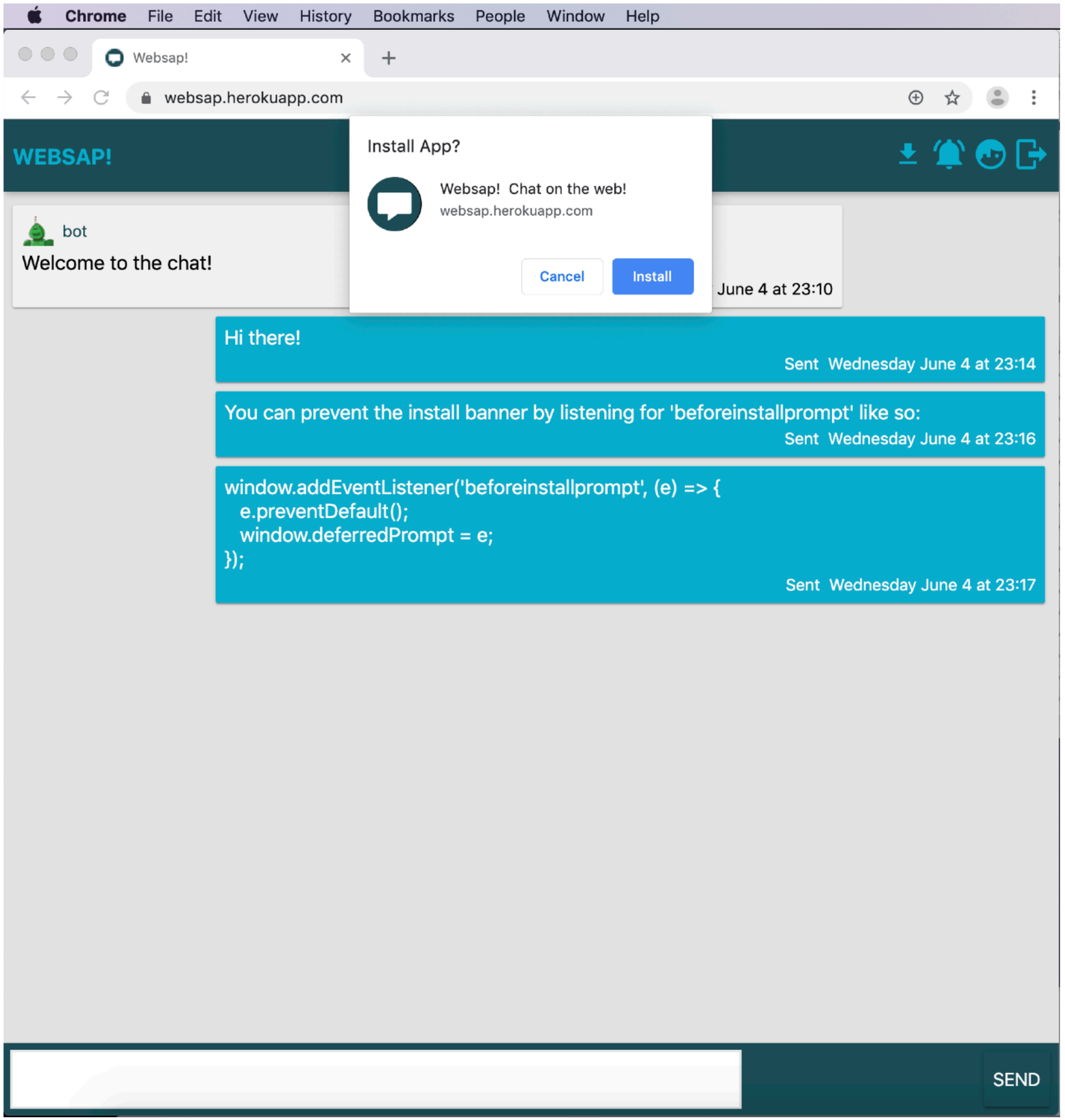


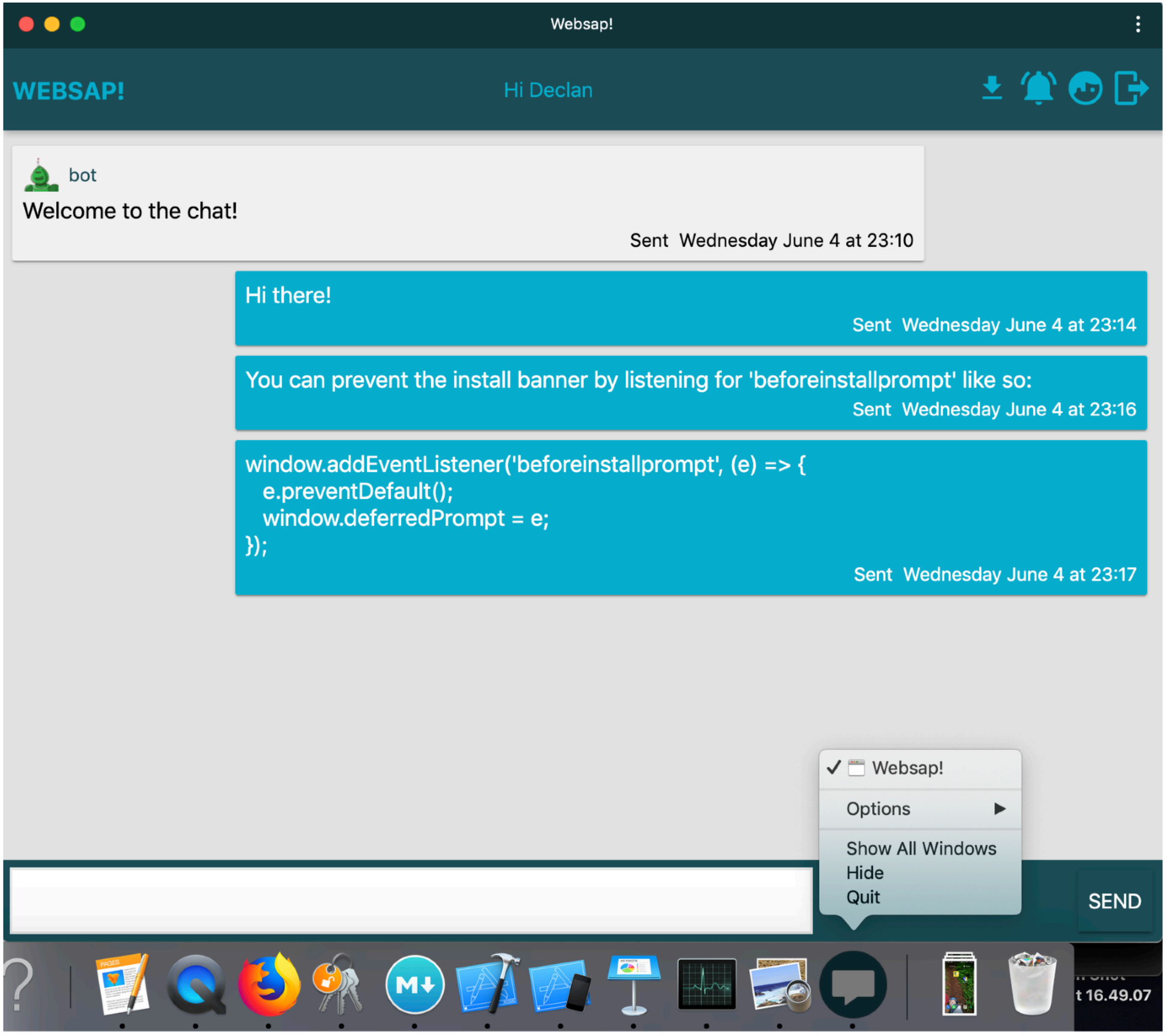
paul.kinlan.me/navigator.share

DESKTOP APPS




Chrome File Edit View History Bookmarks People Window Help
Websap!
websap.herokuapp.com
WEBSAP!
Hi Declan
Add to homescreen
bot
Welcome to the chat!
Sent Wednesday June 4 at 23:10
Hi there!
Sent Wednesday June 4 at 23:14
You can prevent the install banner by listening for 'beforeinstallprompt' like so:
Sent Wednesday June 4 at 23:16
window.addEventListener('beforeinstallprompt', (e) => {
e.preventDefault();
window.deferredPrompt = e;
});
Sent Wednesday June 4 at 23:17
SEND


Chrome File Edit View History Bookmarks People Window Help
Websap!
websap.herokuapp.com
WEBSAP!
Install App?
Websap! Chat on the web!
websap.herokuapp.com
Cancel
Install
bot
Welcome to the chat!
June 4 at 23:10
Hi there!
Sent Wednesday June 4 at 23:14
You can prevent the install banner by listening for 'beforeinstallprompt' like so:
Sent Wednesday June 4 at 23:16
window.addEventListener('beforeinstallprompt', (e) => {
e.preventDefault();
window.deferredPrompt = e;
});
Sent Wednesday June 4 at 23:17
SEND


Websap!
WEBSAP!
Hi Declan
bot
Welcome to the chat!
Sent Wednesday June 4 at 23:10
Hi there!
Sent Wednesday June 4 at 23:14
You can prevent the install banner by listening for 'beforeinstallprompt' like so:
Sent Wednesday June 4 at 23:16
window.addEventListener('beforeinstallprompt', (e) => {
e.preventDefault();
window.deferredPrompt = e;
});
Sent Wednesday June 4 at 23:17
Websap!
Options
Show All Windows
Hide
Quit
SEND

PLAY STORE

Google Play

Search

De

Apps

Categories

Home

Top Charts

New Releases

My apps

Shop

Games

Family

Editors' Choice

Account

Payment methods

My subscriptions

Redeem

Buy gift card

My wishlist

My Play activity

Parent Guide

PICK·UP·10

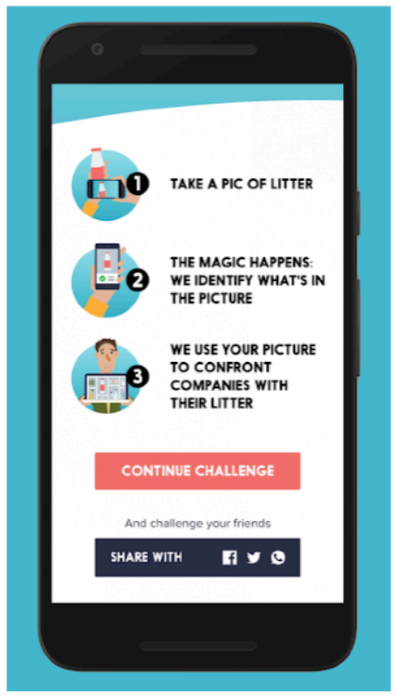
Pick Up 10

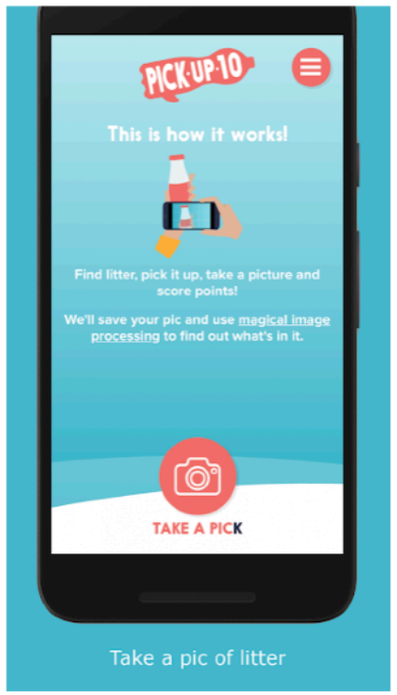
De Voorhoede Lifestyle

3 PEGI 3

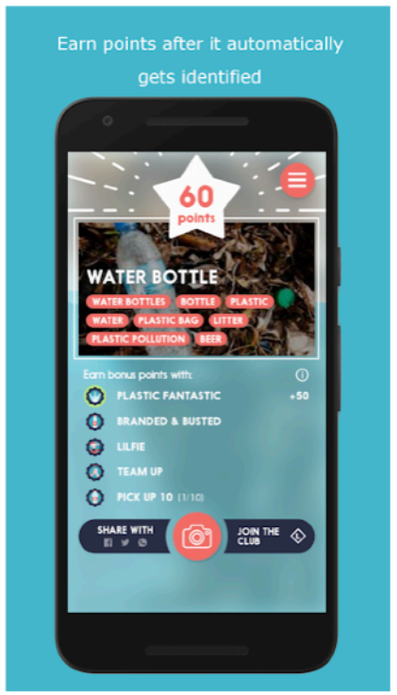
This app is compatible with your device.

Installed

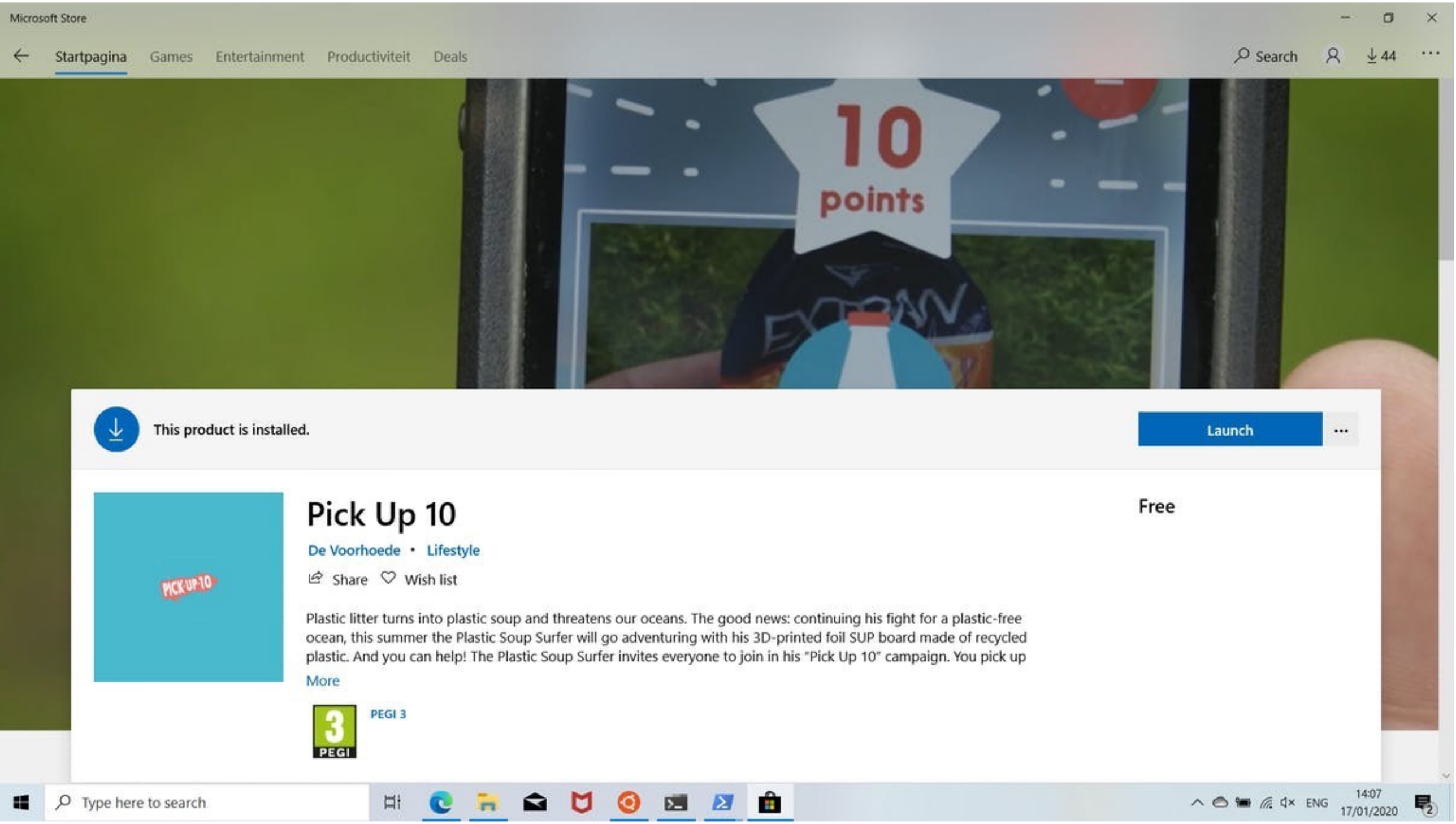




Take a pic of litter



MICROSOFT STORE

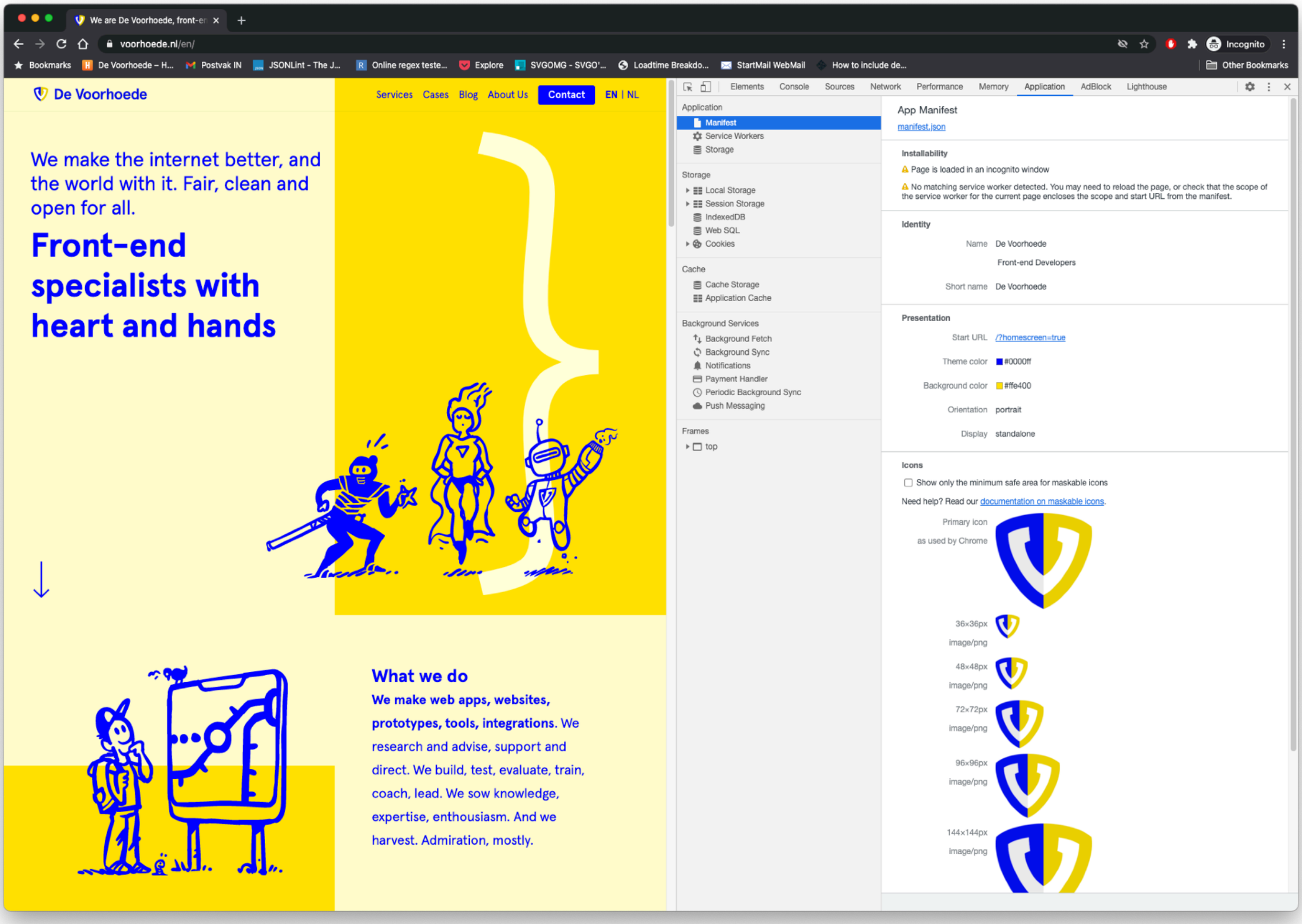


HTTPS

Service worker

Manifest file

MANIFEST.JSON



MANIFEST.JSON

<link rel="manifest" href="/manifest.json">

manifest.json ×

```
{
  "name": "De Voorhoede \n Front-end Developers",
  "short_name": "De Voorhoede",
  "start_url": "/?homescreen=true",
  "display": "standalone",
  "orientation": "portrait",
  "gcm_sender_id": "482941778795",
  "theme_color": "#0000ff",
  "background_color": "#ffe400",
  "icons": [
      {
          "src": "/images/social/android-36x36.png",
          "sizes": "36x36",
          "type": "image/png"
      },
      {
          "src": "/images/social/android-48x48.png",
          "sizes": "48x48",
          "type": "image/png"
      },
      {
          "src": "/images/social/android-72x72.png",
          "sizes": "72x72",
          "type": "image/png"
      },
      {
          "src": "/images/social/android-96x96.png",
          "sizes": "96x96",
          "type": "image/png"
      },
      {
          "src": "/images/social/android-144x144.png",
          "sizes": "144x144",
          "type": "image/png"
      },
      {
          "src": "/images/social/android-192x192.png",
          "sizes": "192x192",
          "type": "image/png"
      }
  ]
}
```

SERVICE WORKER

Service workers essentially act as proxy servers that sit between web applications, the browser, and the network (when available). They are intended, among other things, to enable the creation of effective offline experiences, intercept network requests and take appropriate action based on whether the network is available. They will also allow access to push notifications and background sync APIs.

MDN - Service worker API

SERVICE WORKER

JavaScript

Web worker

Runs in the background

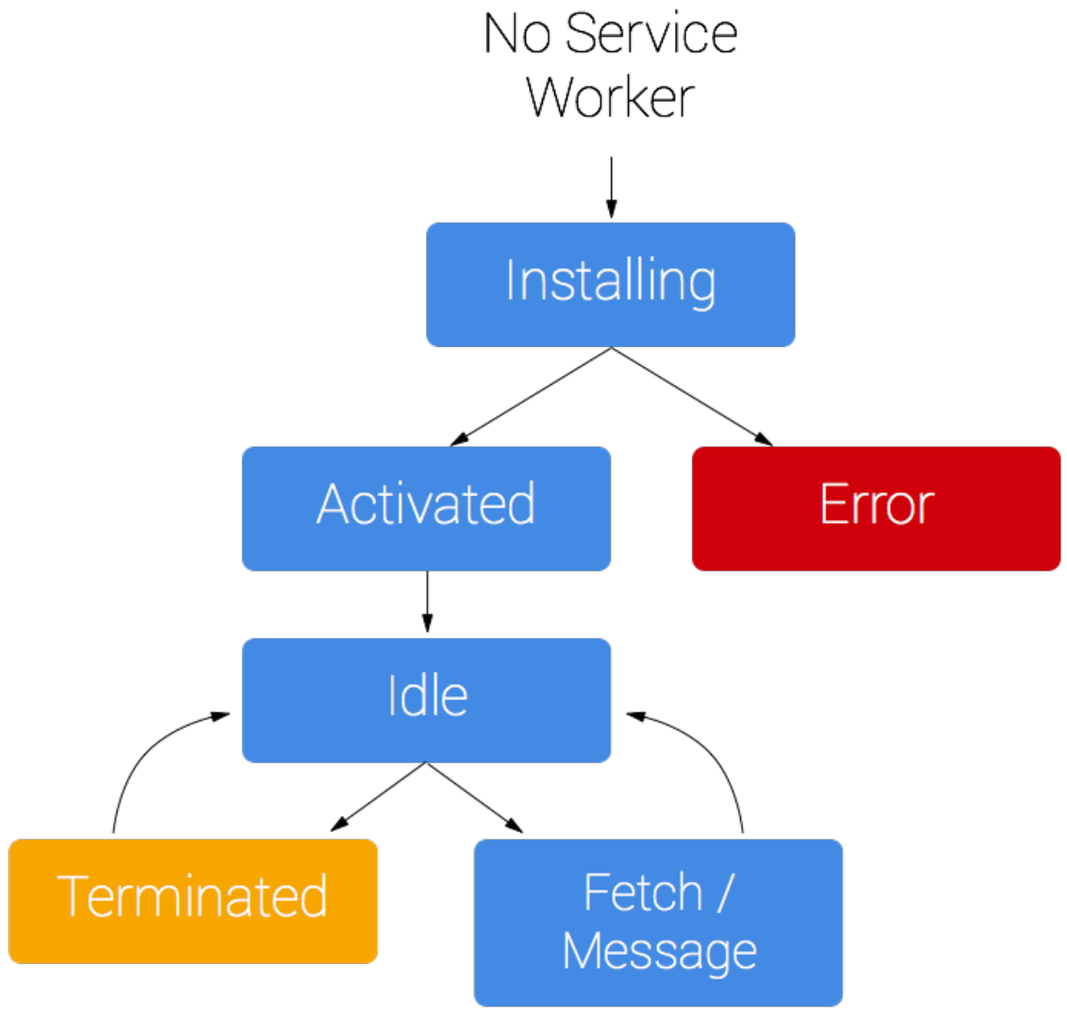
Separate thread

Event based

No access to DOM

No global state

SERVICE WORKER



developers.google.com/web/fundamentals/primers/service-workers

REGISTER SW

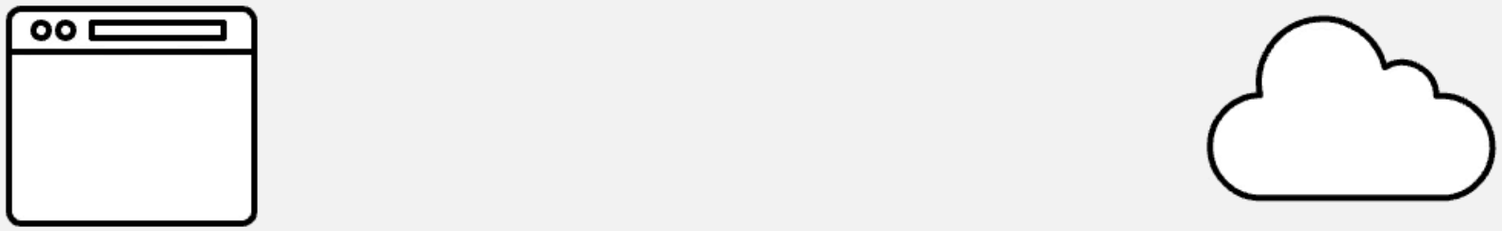
```
if ('serviceWorker' in navigator) {
    window.addEventListener(‘load’, function() {
        navigator.serviceWorker.register('/service-worker.js')
            .then(function(registration) {
                return registration.update();
            })
    });
}
```

SERVICE WORKER LIFECYCLE

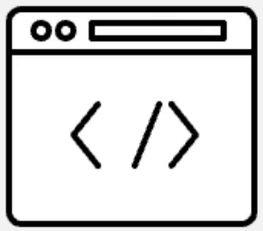
```
self.addEventListener('install', event => {
    // do I even exist?
});
```

```
self.addEventListener('activate', event => {
    // am I even active?
});

self.addEventListener('fetch', event => {
    // can I even do cool stuff?
});
```

INITIAL PAGE LOAD



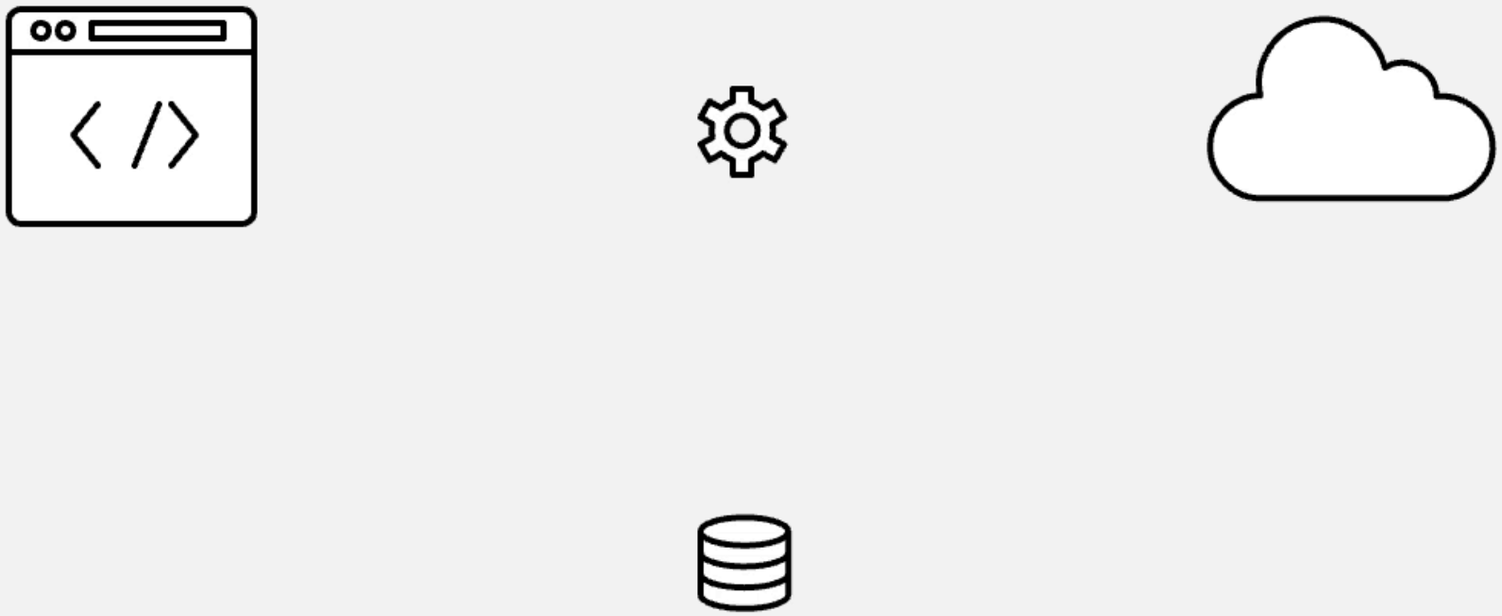
REGISTER SERVICE WORKER



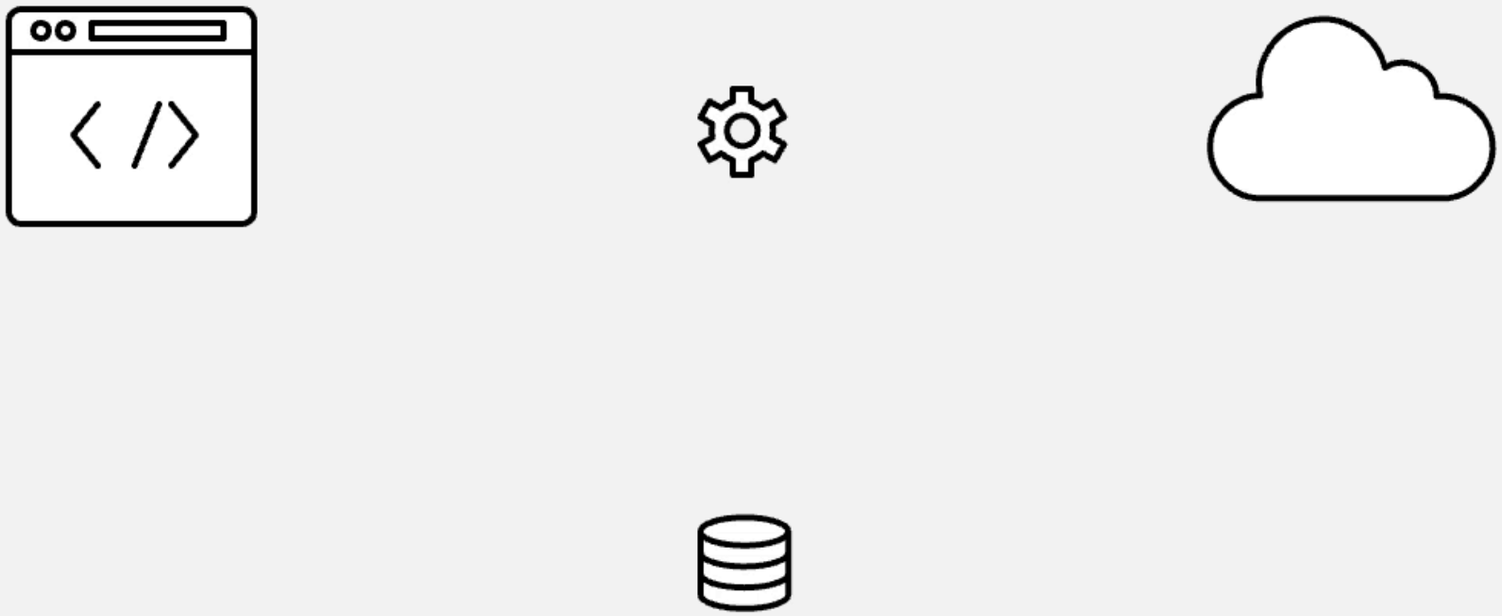
SERVICE WORKER AS PROXY

HIJACK FETCH

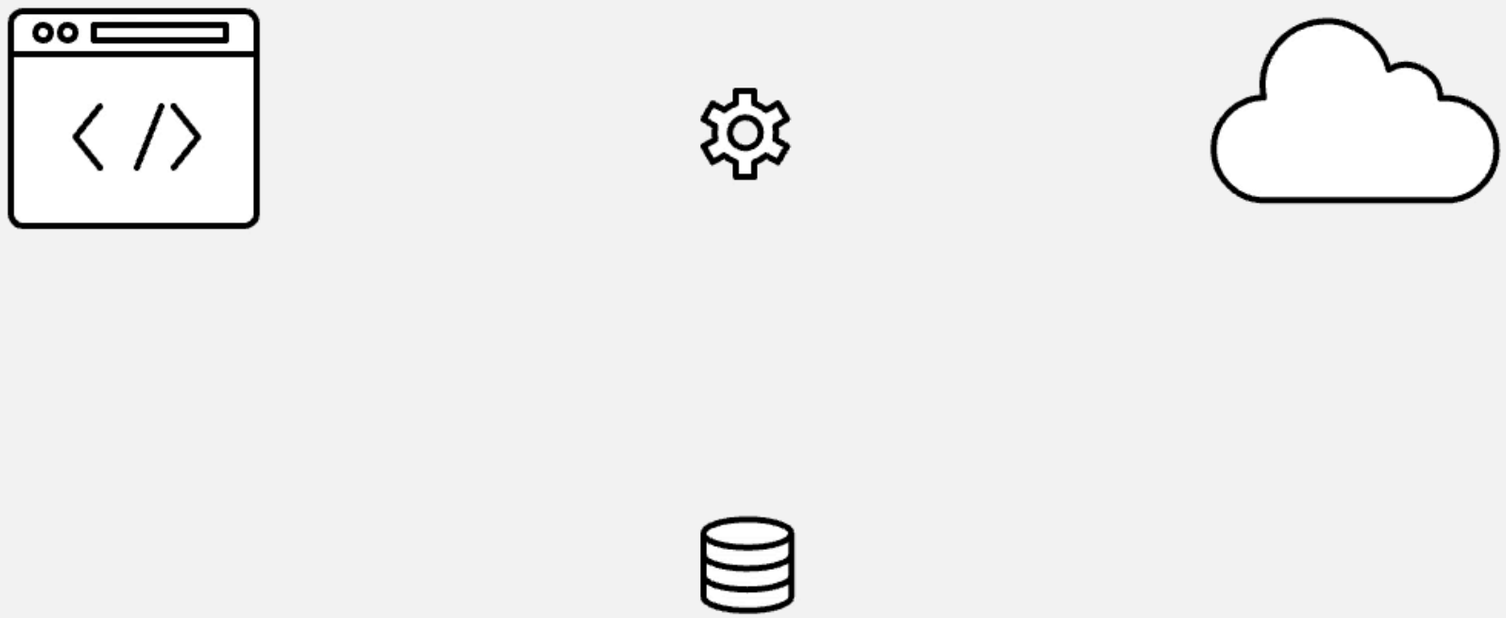
CACHE ON INSTALL



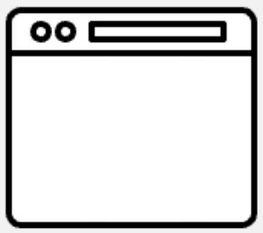
CACHE ON FETCH



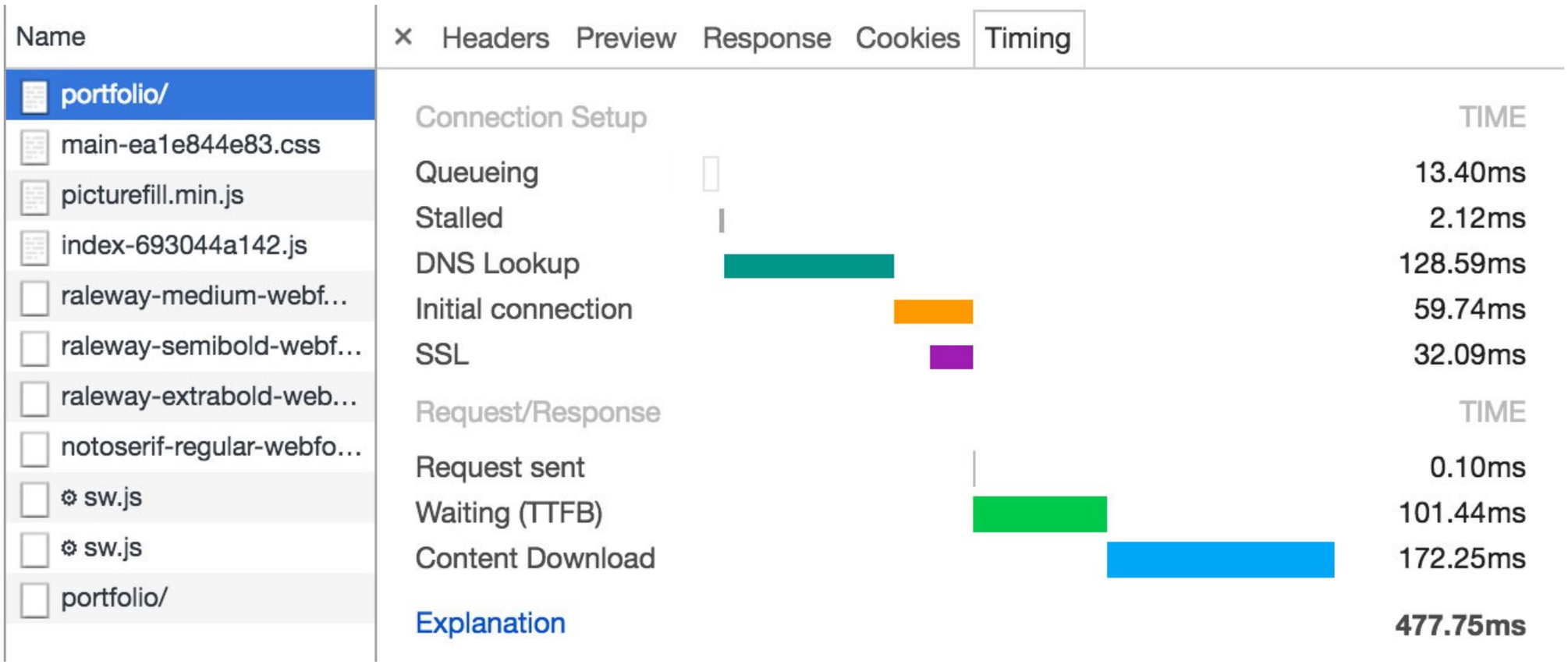
SERVE FROM CACHE



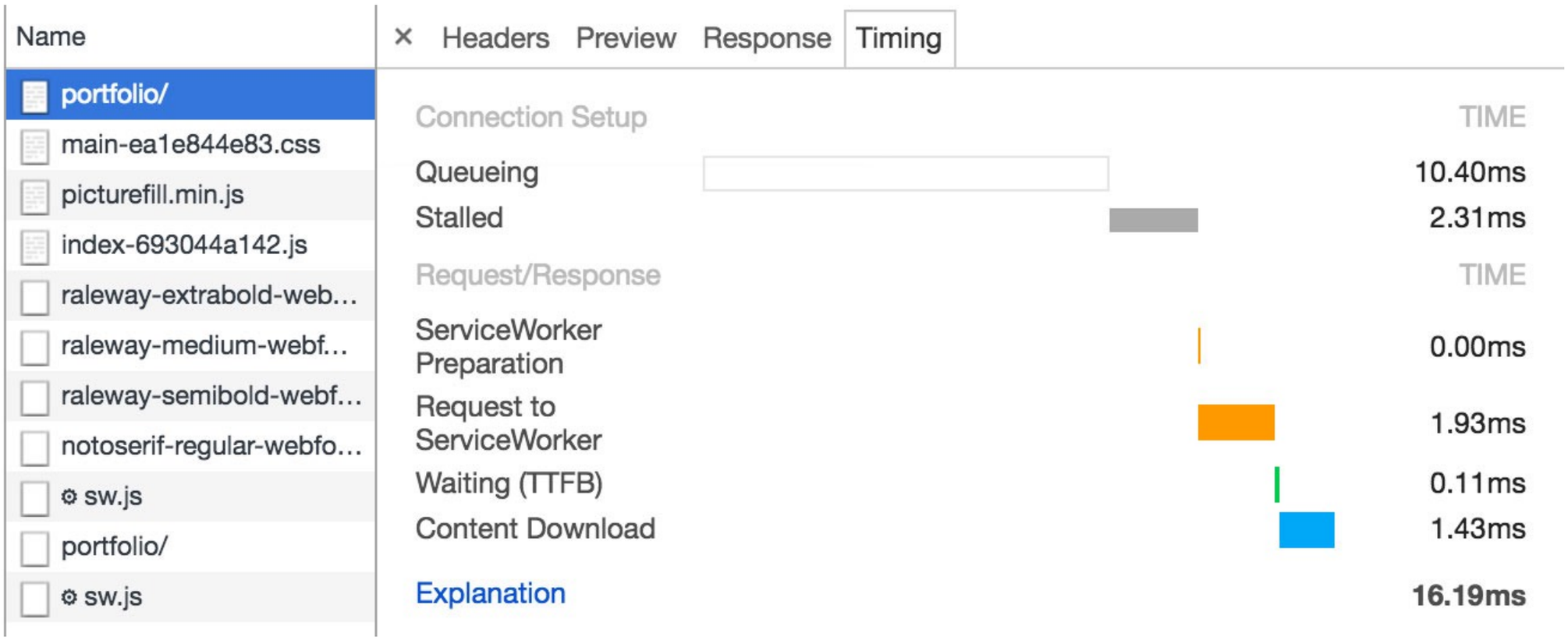
UPDATE SERVICE WORKER



FROM


Name
Headers
Preview
Response
Cookies
Timing
portfolio/
main-ea1e844e83.css
picturefill.min.js
index-693044a142.js
raleway-medium-webf…
raleway-semibold-webf…
raleway-extrabold-web…
notoserif-regular-webfo…
sw.js
sw.js
portfolio/
Connection Setup
TIME
Queueing
13.40ms
Stalled
2.12ms
DNS Lookup
128.59ms
Initial connection
59.74ms
SSL
32.09ms
Request/Response
TIME
Request sent
0.10ms
Waiting (TTFB)
101.44ms
Content Download
172.25ms
Explanation
477.75ms

TO


Name
portfolio/
main-ea1e844e83.css
picturefill.min.js
index-693044a142.js
raleway-extrabold-web…
raleway-medium-webf…
raleway-semibold-webf…
notoserif-regular-webfo…
sw.js
portfolio/
sw.js
Headers
Preview
Response
Timing
Connection Setup
TIME
Queueing
10.40ms
Stalled
2.31ms
Request/Response
TIME
ServiceWorker Preparation
0.00ms
Request to ServiceWorker
1.93ms
Waiting (TTFB)
0.11ms
Content Download
1.43ms
Explanation
16.19ms

CACHING THINGS

PRECACHING

```
const CORE_CACHE_NAME = ‘core-cache’;
const CORE_ASSETS = [ /* css, bundle.js, offline page */ ];

self.addEventListener(‘install’, event => {
    event.waitUntil(
        caches.open(CORE_CACHE_NAME)
            .then(cache => cache.addAll(CORE_ASSETS))
            .then(() => self.skipWaiting())
    )
});
```

SERVE FROM CACHE

```
self.addEventListener('fetch', event => {
    const request = event.request;
    if (isInCoreCache(request)) {
        event.respondWith(
            caches.open(CORE_CACHE_NAME)
                .then(cache => cache.match(request.url))
        )
    }
})
```

RUNTIME CACHING

```
self.addEventListener('fetch', event => {
    const request = event.request;
    if (isSomethingParticular(request) && isNotCached(request)) {
        event.respondWith(
            return fetch(request).then(response => {
                return caches.open('mycache')
                    .then(cache => {
                        cache.put(request, response.clone())
                    })
                    .then(() => response)
            })
        )
    }
})
```

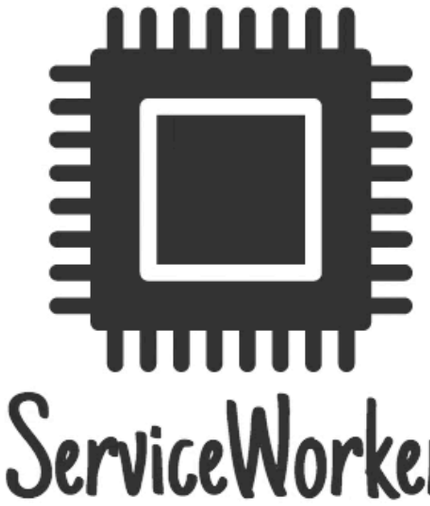
HIJACK FETCH

```
self.addEventListener('fetch', event => {
    const request = event.request;
    event.respondWith(
        ¯\_(ツ)_/¯
    );
});
```

```
self.addEventListener('install', event => {
    // on installation
});

self.addEventListener('activate', event => {
    // on activation
});

self.addEventListener('fetch', event => {
    // on outgoing request
});

self.addEventListener('push', event => {
    // on push message received
});

self.addEventListener('message', event => {
    // on message from app
});

self.addEventListener('sync', event => {
    //
});
```

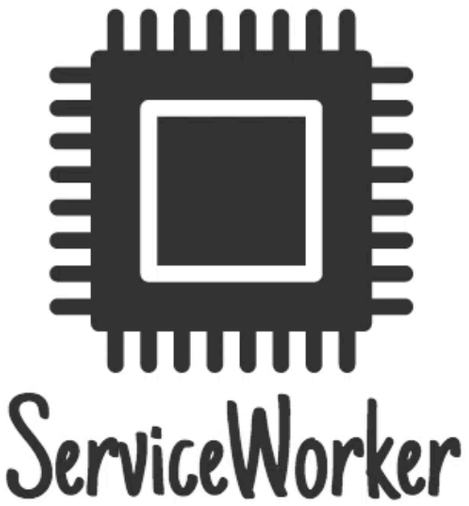
CACHING STRATEGIES

CACHE ONLY



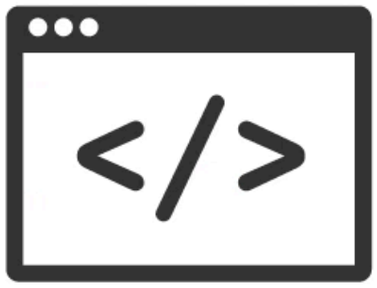
Offline cookbook by Jake Archibald

NETWORK ONLY



Offline cookbook by Jake Archibald

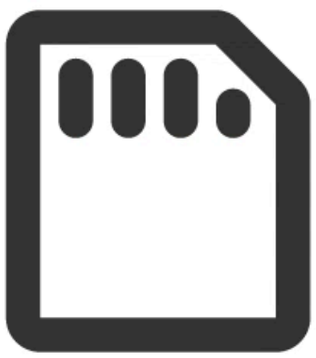
CACHE FIRST



Page

ServiceWorker

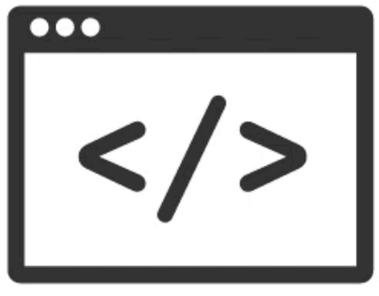
Network



Cache

Offline cookbook by Jake Archibald

NETWORK FIRST



Page

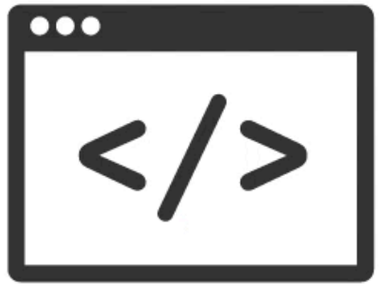
ServiceWorker

Cache

Network

Offline cookbook by Jake Archibald

STALE WHILE REVALIDATE



Page

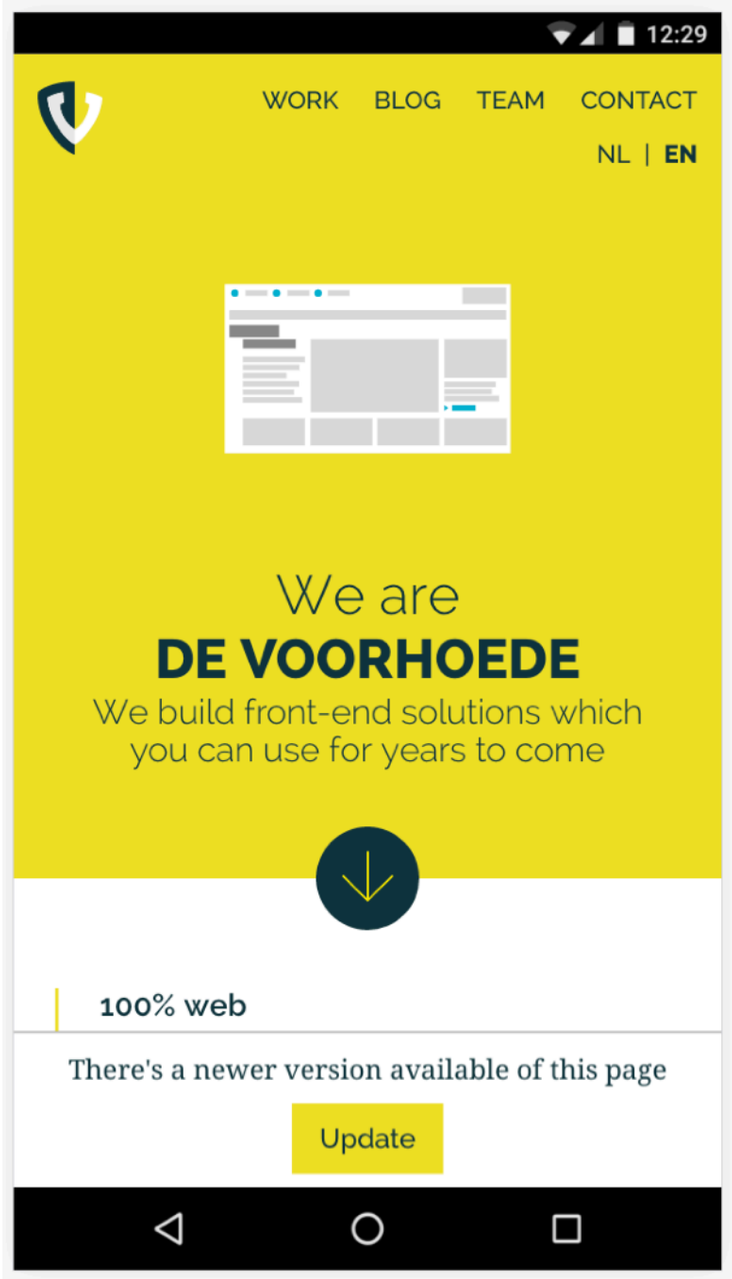
ServiceWorker

Cache

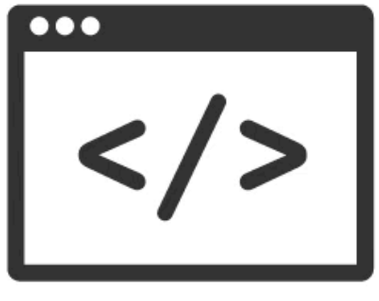
Network

Offline cookbook by Jake Archibald

STALE WHILE REVALIDATE



GENERIC FALLBACK



Page

ServiceWorker

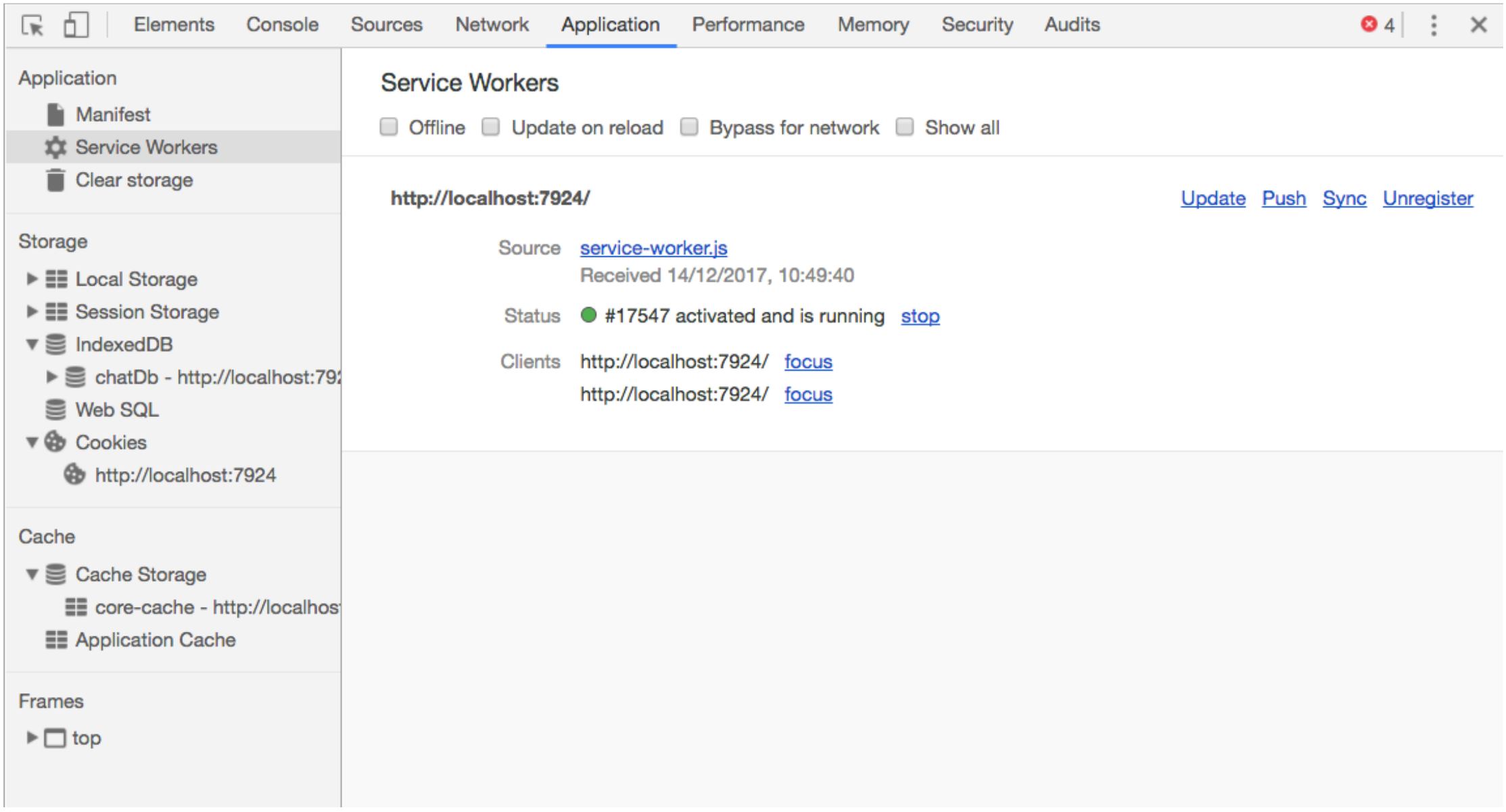
Cache

Network

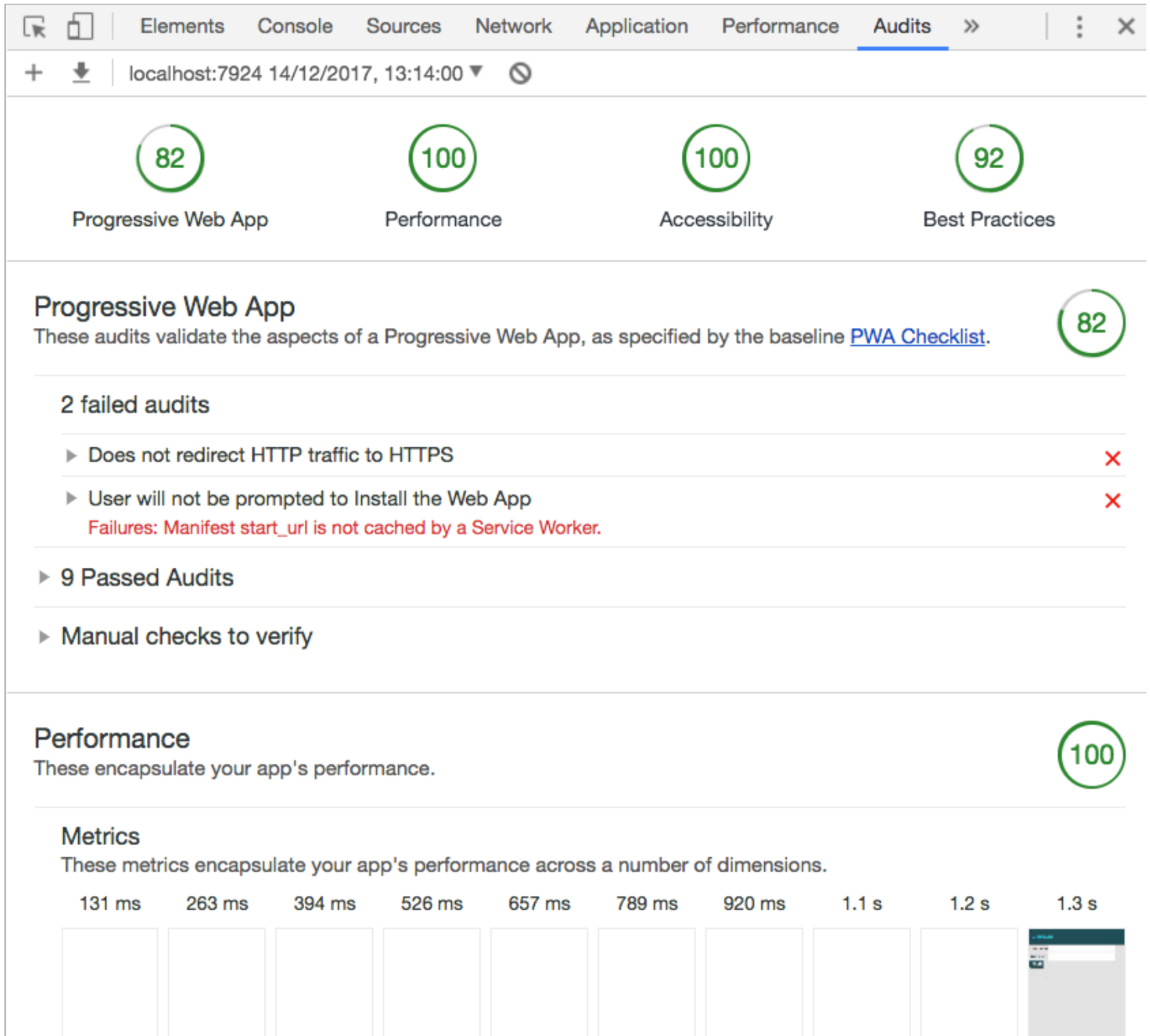
Offline cookbook by Jake Archibald

DEVTOOLS & TESTING

CHROME DEV TOOLS



LIGHTHOUSE


Elements
Console
Sources
Network
Application
Performance
Audits
localhost:7924 14/12/2017, 13:14:00
82
Progressive Web App
100
Performance
100
Accessibility
92
Best Practices
Progressive Web App
These audits validate the aspects of a Progressive Web App, as specified by the baseline PWA Checklist.
82
2 failed audits
Does not redirect HTTP traffic to HTTPS
User will not be prompted to Install the Web App
Failures: Manifest start_url is not cached by a Service Worker.
9 Passed Audits
Manual checks to verify
Performance
These encapsulate your app's performance.
100
Metrics
These metrics encapsulate your app's performance across a number of dimensions.
131 ms
263 ms
394 ms
526 ms
657 ms
789 ms
920 ms
1.1 s
1.2 s
1.3 s