

YES, YOUR BROWSER CAN DO THAT(*probably*)

A Look At Modern Web
APIs You Might Not Know

PROGRESSIVE ENHANCEMENT

Progressive enhancement is a design philosophy that provides a baseline of essential content and functionality to as many users as possible while delivering the best possible experience only to users of the most modern browsers

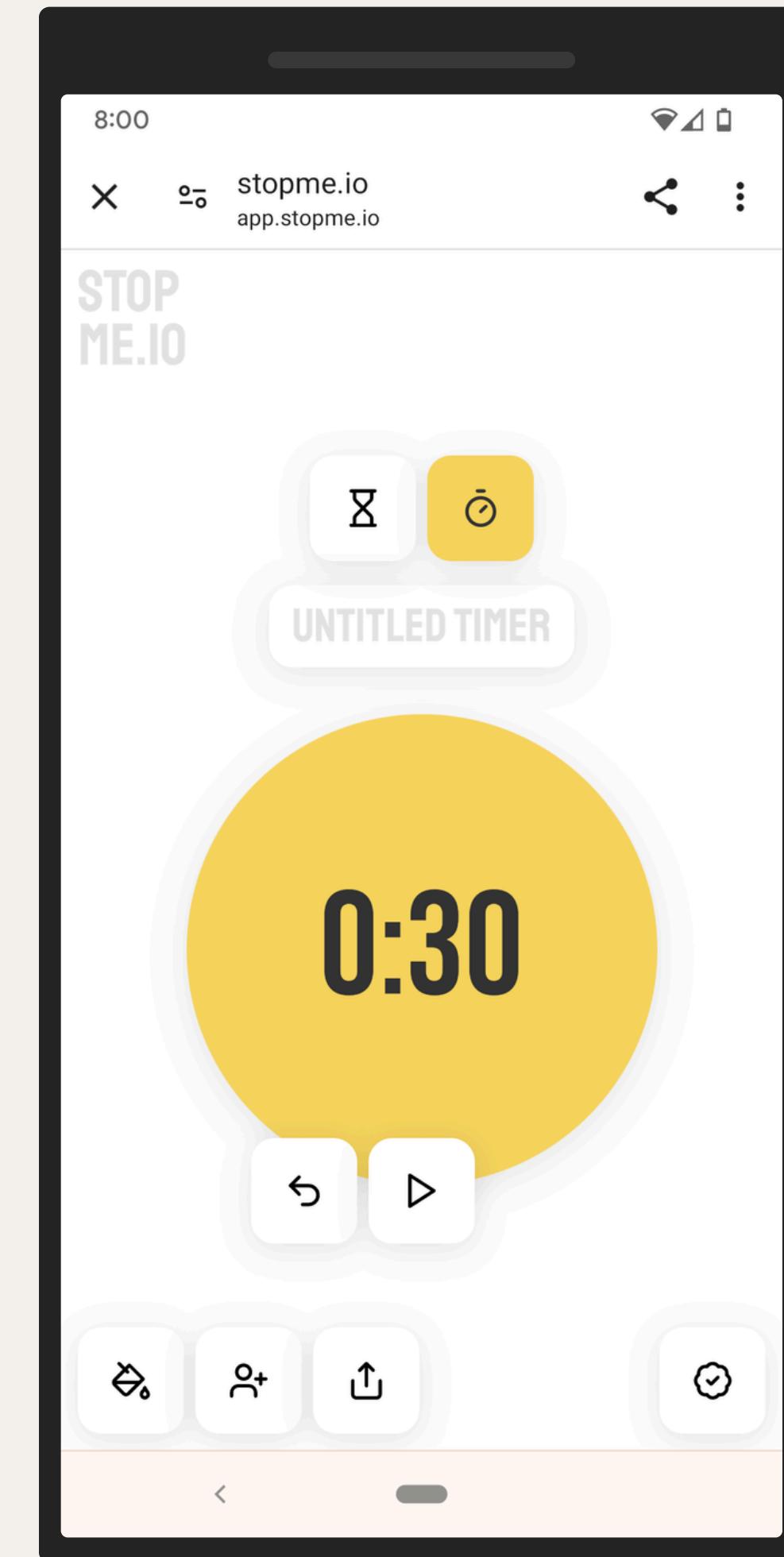
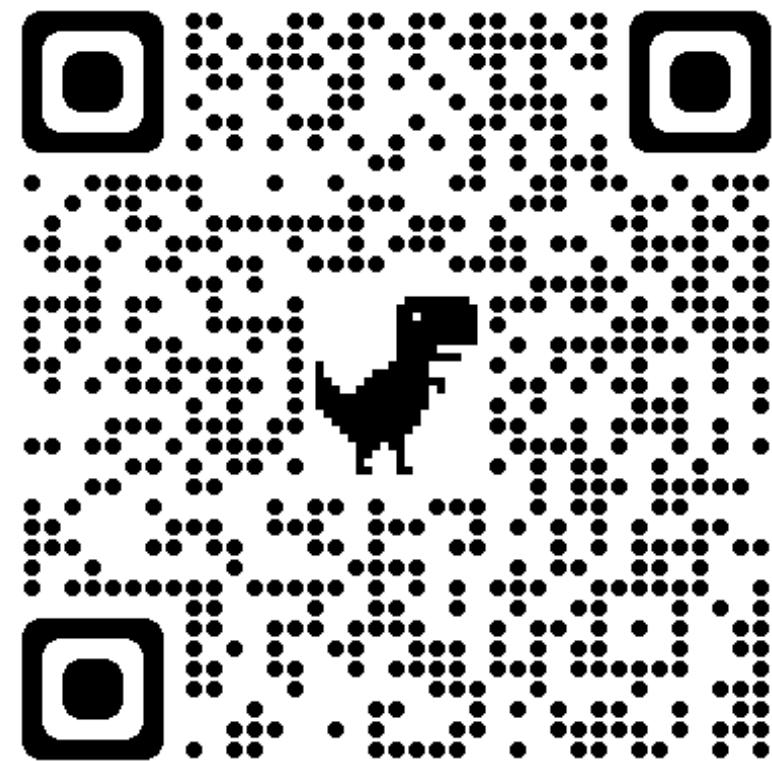
MOVING THE NEEDLE & PHONE GAP

aka Cordova

**LET'S LOOK AT SOME
BROWSER APIS**

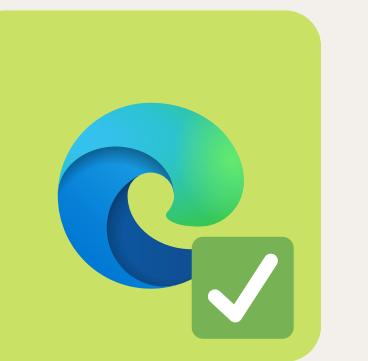
STOPME.IO

The ultimate SaaS (Stopwatch as a Service) product for everyone



**OBSERVE
YOUR APP ...**

RESIZE OBSERVER



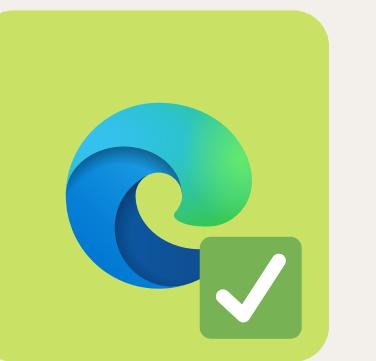
RESIZE OBSERVER



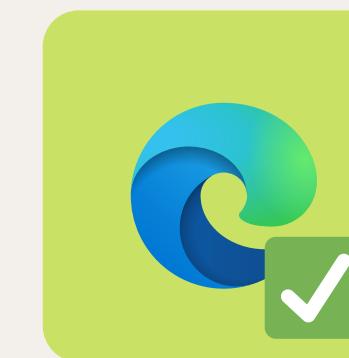
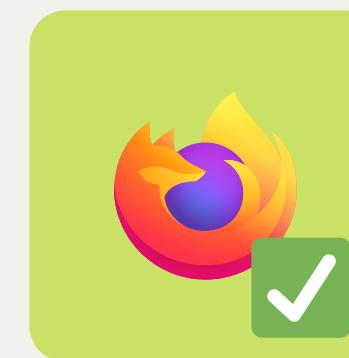
```
const callback = (entries) => {  
  entries.forEach((entry) => {  
    // *.contentBoxSize  
  });  
}  
  
const observer =  
  new ResizeObserver(callback, options);  
  
const el = document.querySelector("element");  
observer.observe(el);
```

02

INTERSECTION OBSERVER



INTERSECTION OBSERVER

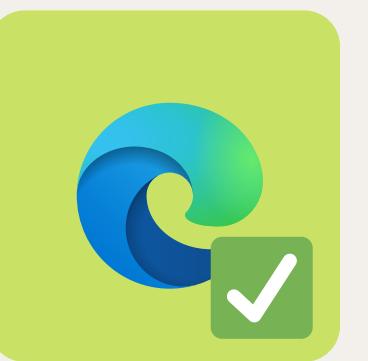


```
const callback = (entries) => {  
  entries.forEach((entry) => {  
    // *.isIntersecting  
    // *.intersectionRect  
    // *.intersectionRatio  
    // ...  
  });  
}  
  
const observer =  
  new IntersectionObserver(callback, options);  
  
const el = document.querySelector("element");  
observer.observe(el);
```

**... AND EVEN THE
USER'S DEVICE**

053

NETWORK INFORMATION API



NETWORK INFORMATION API



```
// Progressive enhancement
const isSupported = "connection" in navigator;

let connection = navigator.connection;
// *.type (e.g. wifi, ...)
// *.effectiveType (e.g. 2g, 3g, ...)
// *.downlink
// *.downlinkMax
// *.rtt
// *.saveData

const callback = () => {
  connection = navigator.connection;
}

navigator.connection.addEventListener(
  "change",
  callback
);
```

NETWORK INFORMATION API



```
// Progressive enhancement
const isSupported = "connection" in navigator;

let connection = navigator.connection;
// *.type (e.g. wifi, ...)
// *.effectiveType (e.g. 2g, 3g, ...)
// *.downlink
// *.downlinkMax
// *.rtt
// *.saveData

const callback = () => {
  connection = navigator.connection;
}

navigator.connection.addEventListener(
  "change",
  callback
);
```

NETWORK INFORMATION API



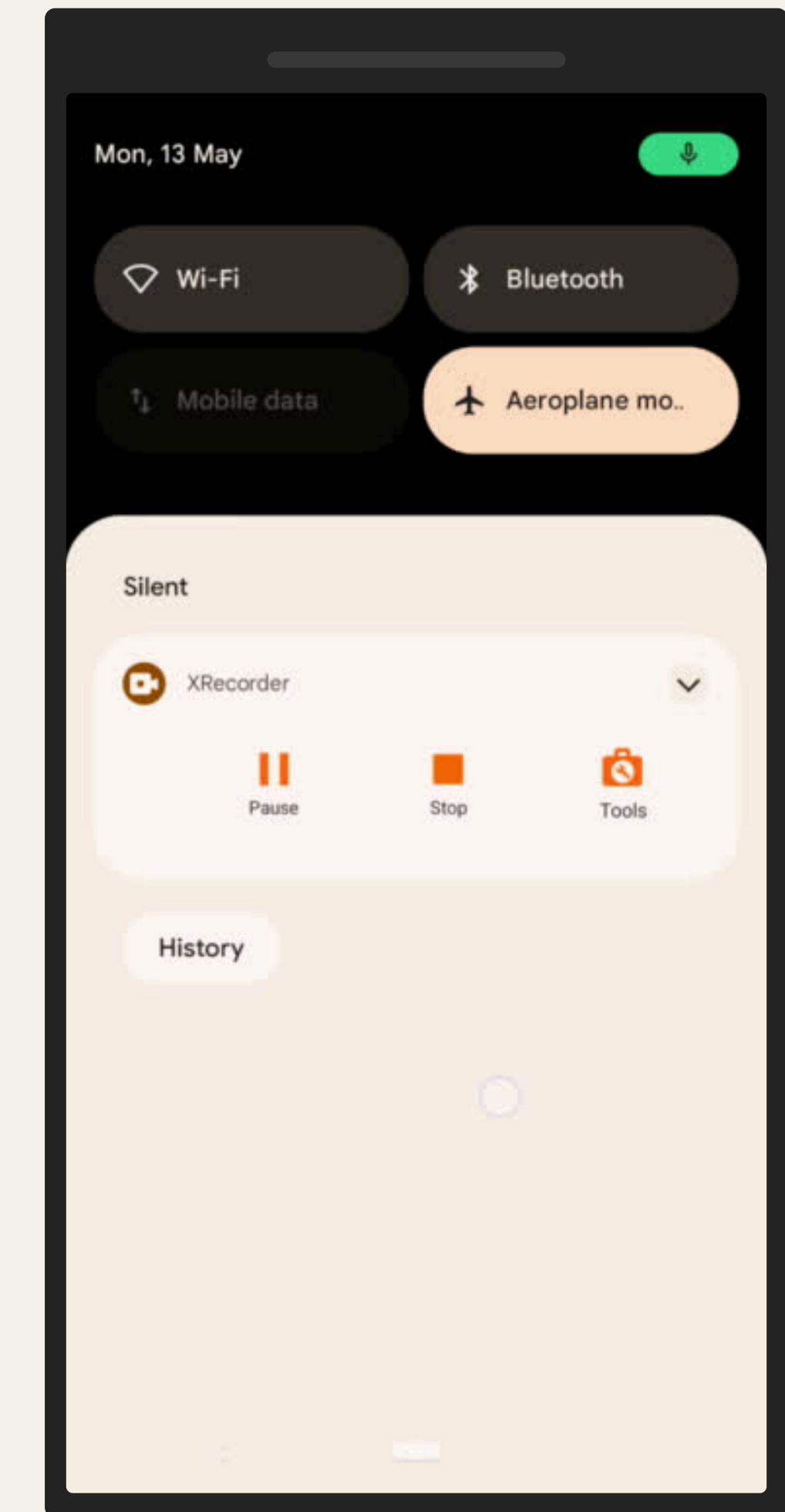
```
// Progressive enhancement
const isSupported = "connection" in navigator;

let connection = navigator.connection;
// *.type (e.g. wifi, ...)
// *.effectiveType (e.g. 2g, 3g, ...)
// *.downlink
// *.downlinkMax
// *.rtt
// *.saveData

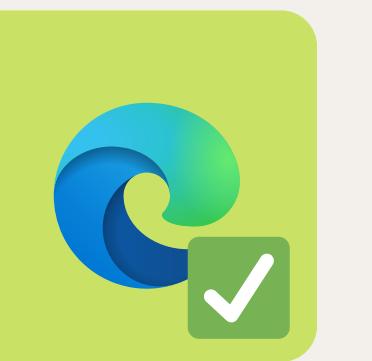
const callback = () => {
  connection = navigator.connection;
}

navigator.connection.addEventListener(
  "change",
  callback
);
```

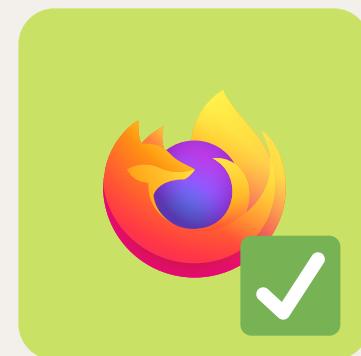
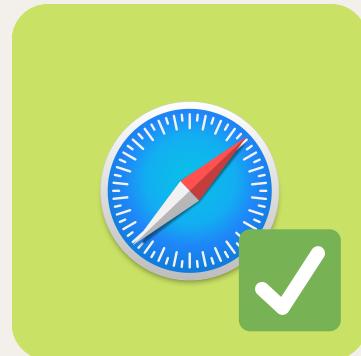
NETWORK INFORMATION API



PAGE VISIBILITY API



PAGE VISIBILITY API

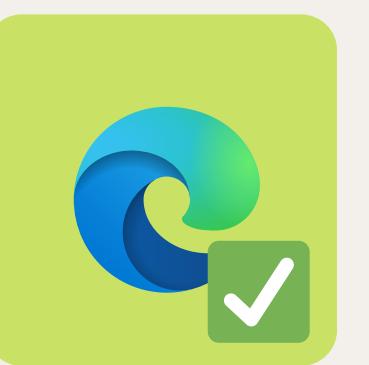


```
let isHidden = document.hidden;

const callback = () => {
  isHidden = document.hidden;
}

document.addEventListener(
  "visibilitychange",
  callback
);
```

PAGE VISIBILITY API



The screenshot shows a dark-themed web browser window. The address bar at the top displays the URL `0:23 — stopme.io` and the subdomain `app.stopme.io`. Below the address bar, there's a navigation bar with links for `/HOME`, `/NEW_24`, `/PROGRAM`, `/SPONSORS`, and `/`. The main content area features large, bold text: **THE CONFERENCE
FOR TECHNOLOGY
PROFESSIONALS**. Below this, a paragraph reads: **/NEW is a celebration of all things technology, uniting the most innovative and passionate techies that are building a brighter future for us all.** At the bottom, another paragraph states: **/NEW_24 will be held 15-16 May 2024 in Newcastle**.

BATTERY STATUS API



BATTERY STATUS API

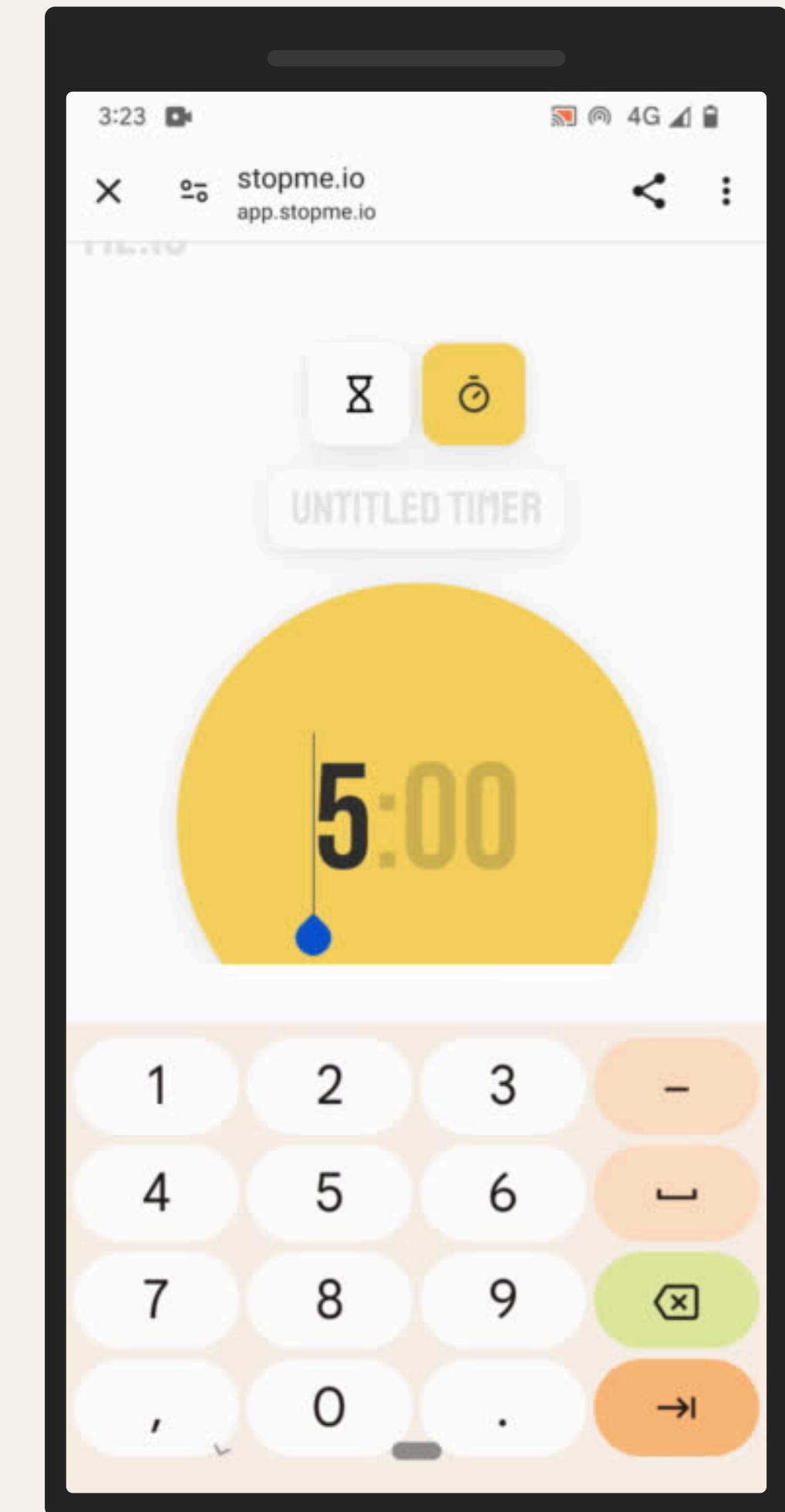


```
const info = await navigator.getBattery();
// *.level
// *.charging
// *.chargingTime
// *.dischargingTime

info.addEventListener("levelchange", ...);
info.addEventListener("chargingchange", ...);
// ...
```

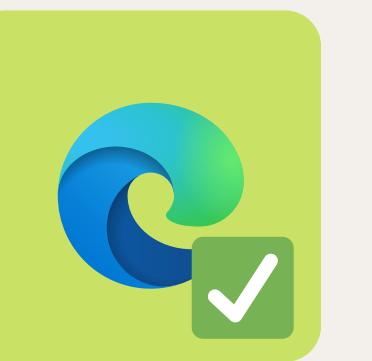
https://developer.mozilla.org/en-US/docs/Web/API/Battery_Status_API

BATTERY STATUS API



**ENHANCE YOUR
COMPONENTS**

SCREEN WAKE LOCK API



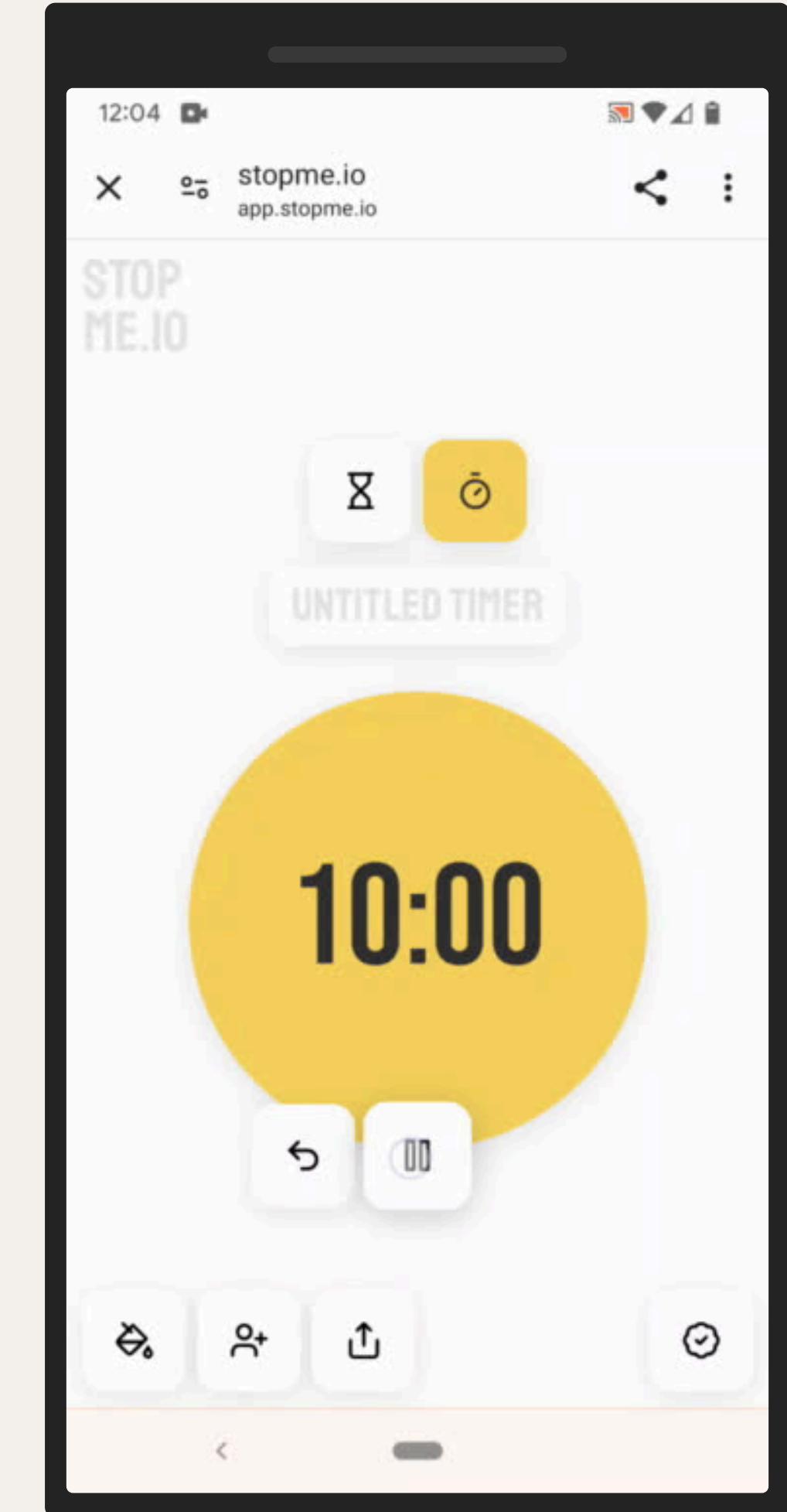
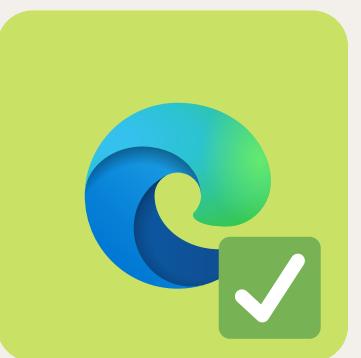
SCREEN WAKE LOCK API



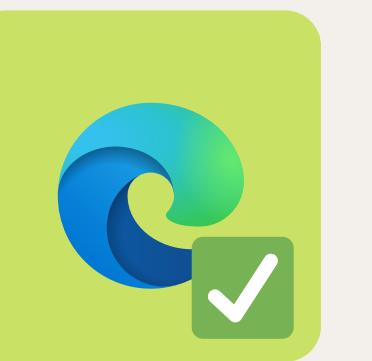
```
const wakeLock = await navigator
  .wakeLock
  .request("screen")
  .catch((e) => {
    // Lock request failed, often because
    // of low battery etc
  });

await wakeLock.release();
```

SCREEN WAKE LOCK API



VIBRATION API



VIBRATION API



```
navigator.vibrate(duration);

// Patterns
// e.g. vibrate for 100ms with 50ms pauses
navigator.vibrate([100, 50, 100, 50, 100]);

// Stop long vibration or pattern
navigator.vibrate(0);
```

https://developer.mozilla.org/en-US/docs/Web/API/Vibration_API

VIBRATION API



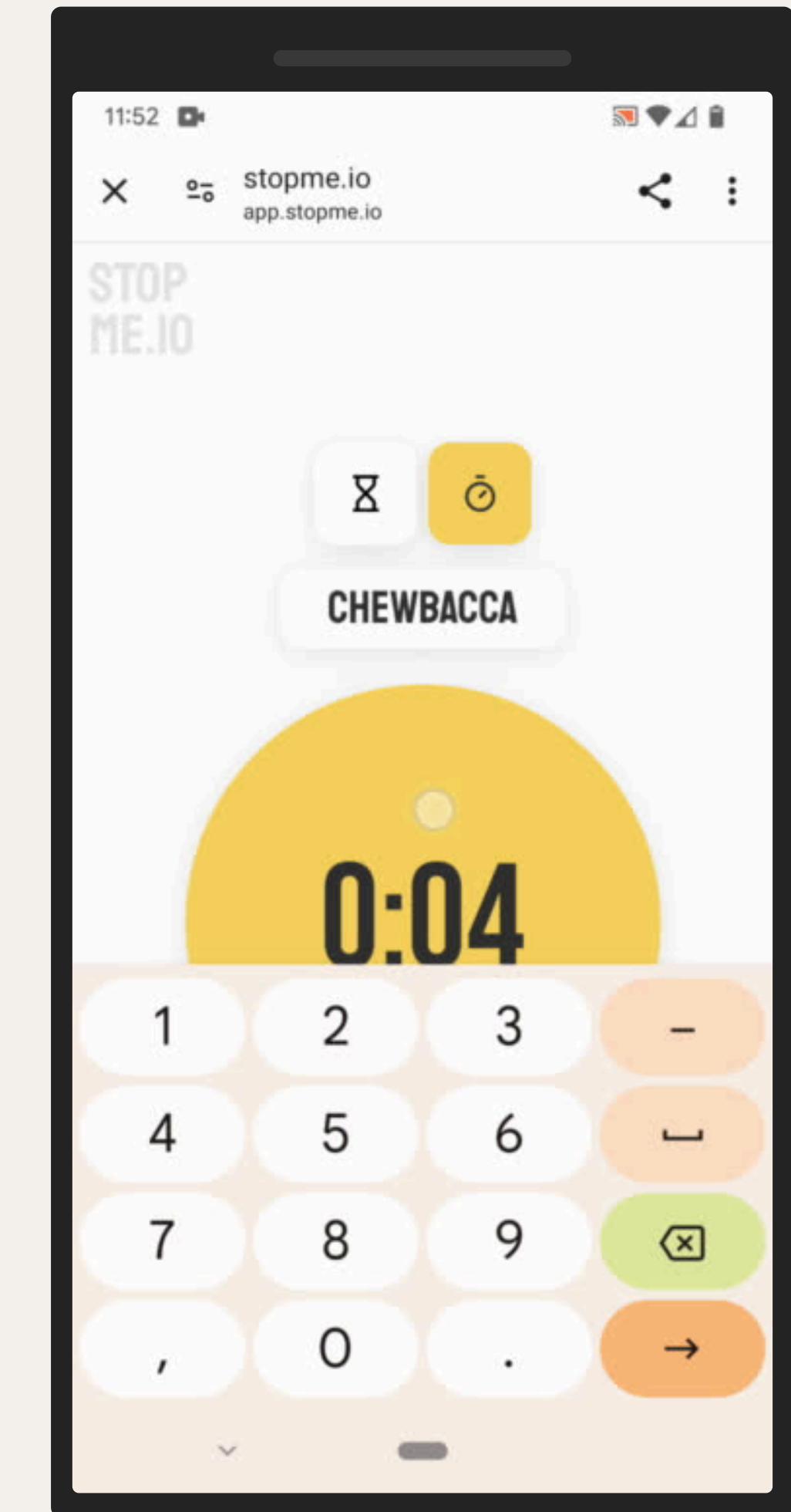
```
// SOS in Morse
navigator.vibrate([
  100, 30, 100, 30, 100, 30, 200, 30, 200,
  30, 200, 30, 100, 30, 100, 30, 100
]);

// Super Mario
navigator.vibrate([
  125, 75, 125, 275, 200, 275, 125, 75,
  125, 275, 200, 600, 200, 600
]);

// Star Wars
navigator.vibrate([
  500, 110, 500, 110, 450, 110, 200, 110,
  170, 40, 450, 110, 200, 110, 170, 40, 500
]);
```

*It vibrates,
I swear!*

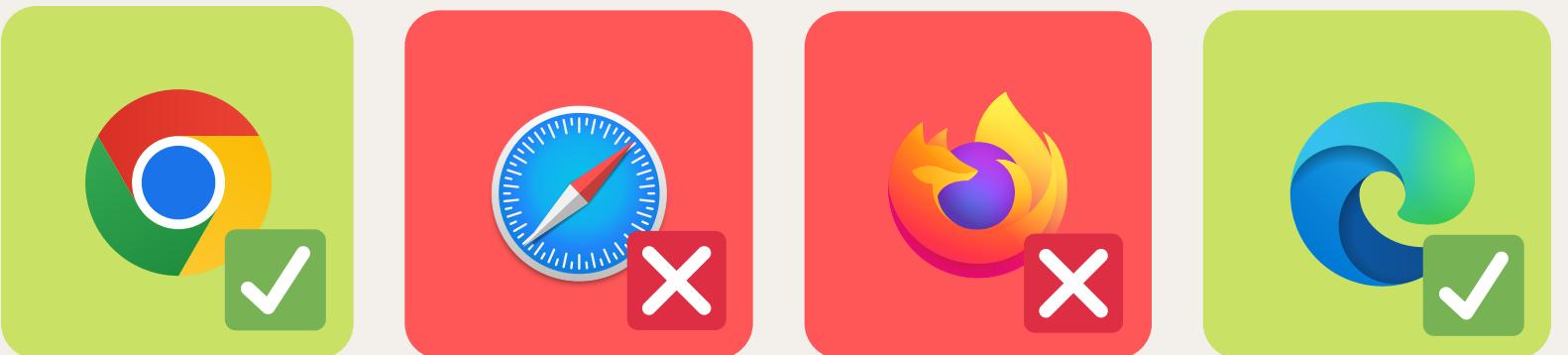
VIBRATION API



EYEDROPPER API



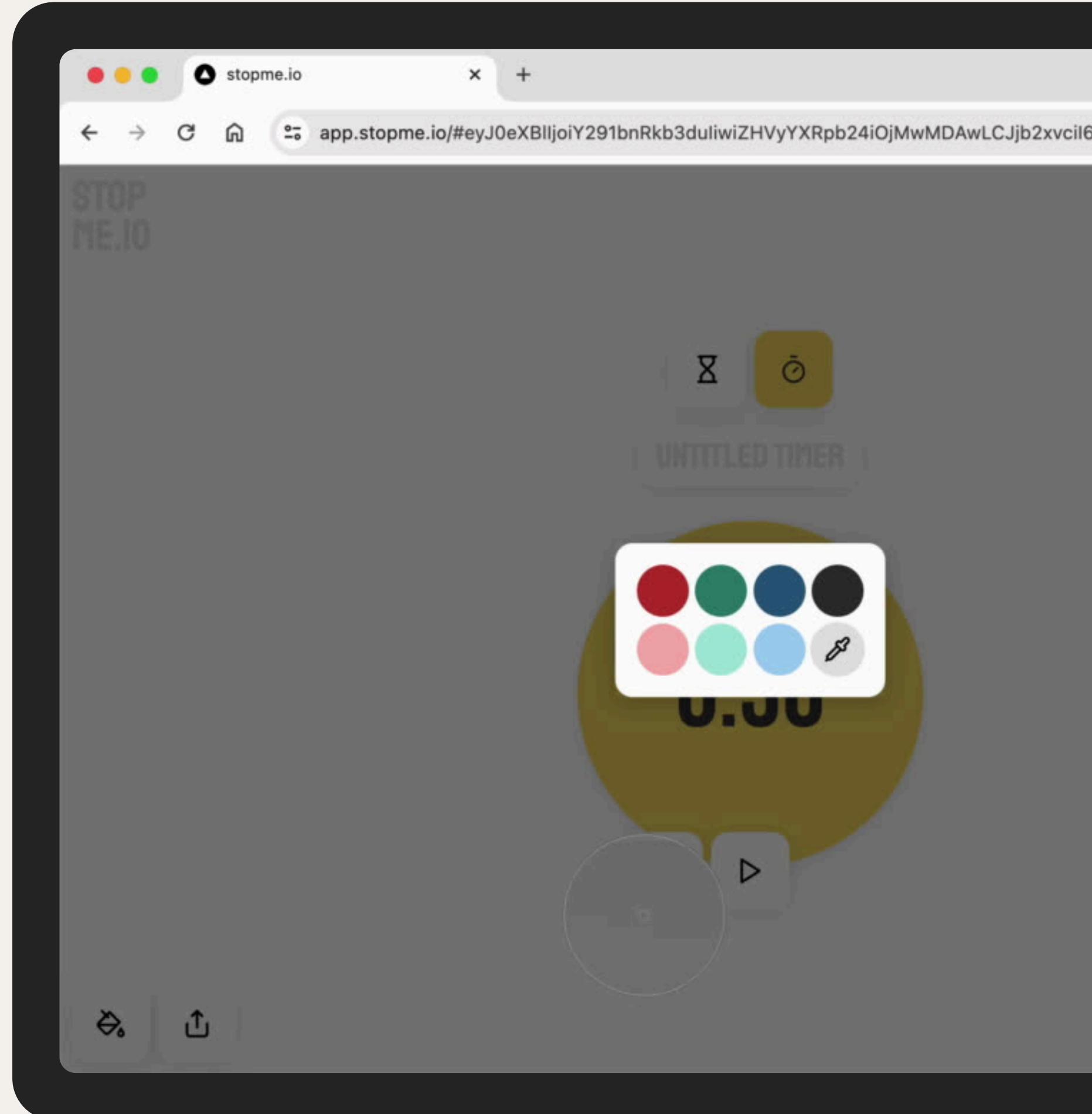
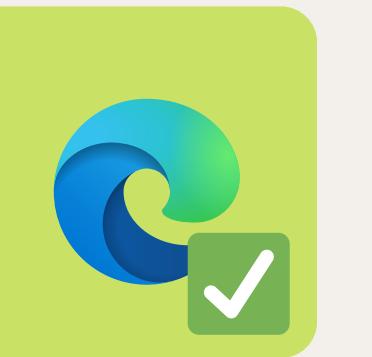
EYEDROPPER API



```
const eyeDropper = new EyeDropper();

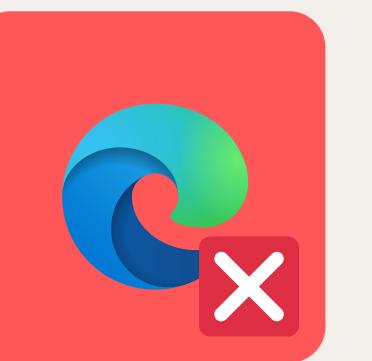
document.getElementById("btn")
  .addEventListener("click", () => {
    // Needs to be triggered by user action
    eyeDropper.open()
      .then((result) => {
        // Returns the selected color
        // *.sRGBHex
      })
      .catch((e) => {
        // Catches any errors, including
        // when the user cancels selection
      });
  });
});
```

EYEDROPPER API



**ALMOST AS GOOD
AS NATIVE**

CONTACT PICKER API



CONTACT PICKER API

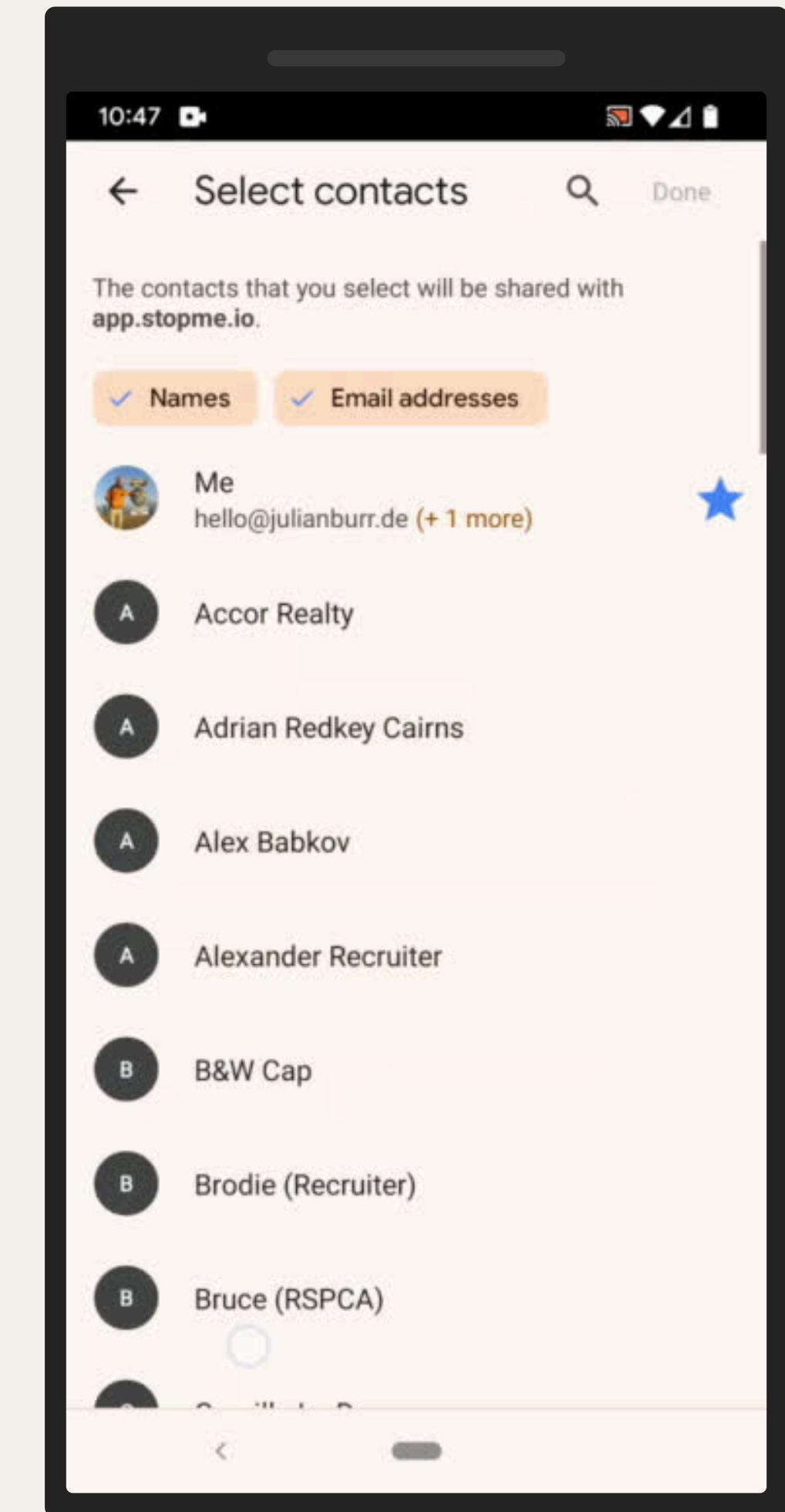
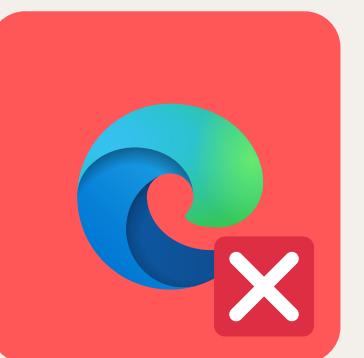


```
const props = ["name", "email", "tel", "icon"];
const opts = { multiple: true };

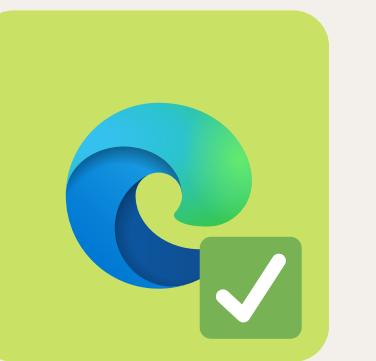
const contacts = await navigator.contacts
  .select(props, opts);
  .catch((e) => {
    // Handle any errors
 });
```

https://developer.mozilla.org/en-US/docs/Web/API>Contact_Picker_API

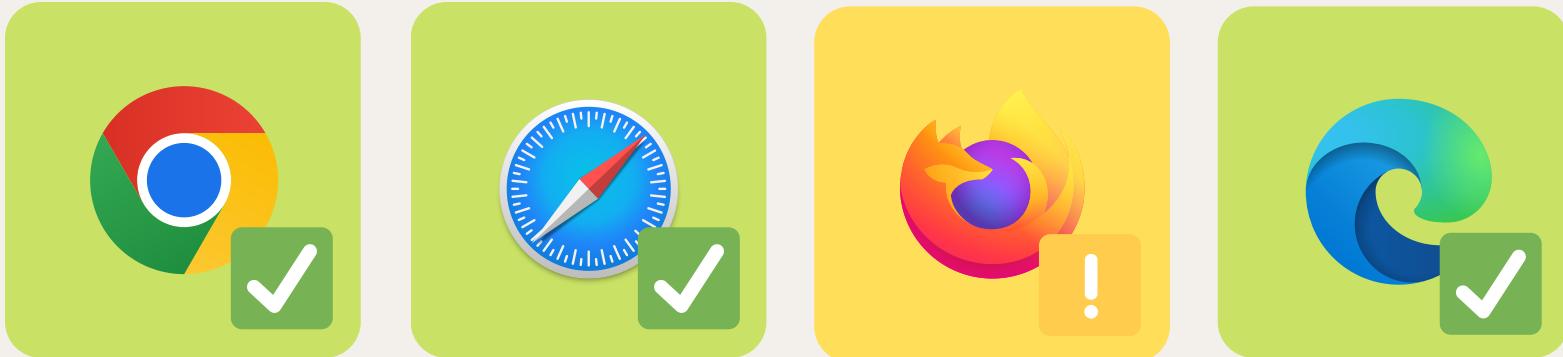
CONTACT PICKER API



WEB SHARE API

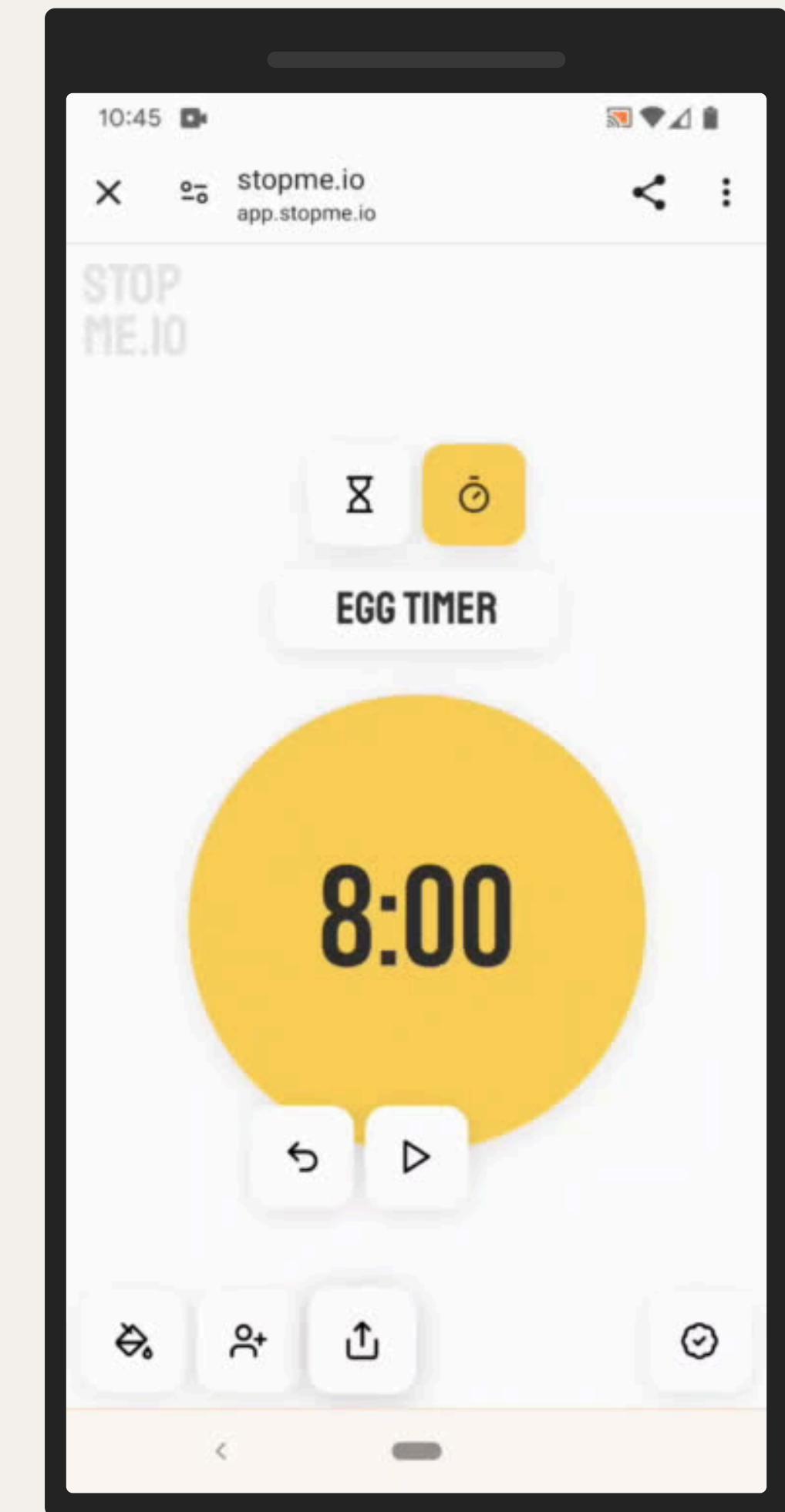
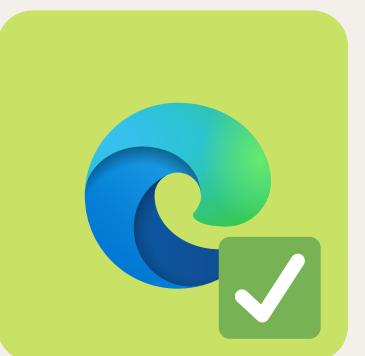
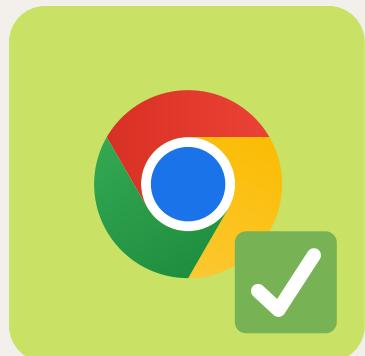


WEB SHARE API



```
const shareData = {  
  title: "../NEW",  
  text: "Celebrate all things technology",  
  url: "https://slashnew.tech"  
};  
  
document.getElementById("btn")  
  .addEventListener("click", () => {  
    navigator.share(shareData)  
      .catch((e) => {  
        // Handle any errors  
      });  
  });
```

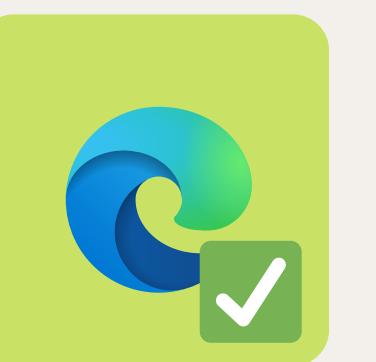
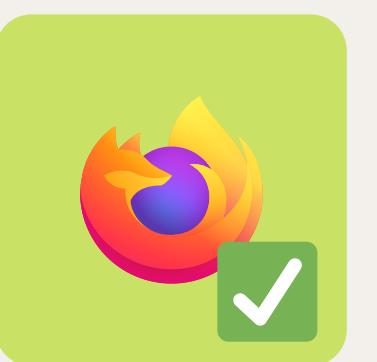
WEB SHARE API



**AUTHENTICATION
DONE BETTER**

passwordless

WEB AUTHN API



WEB AUTHN API



```
// Create credentials object on the client
// using challenge generated on the server
const registerCredential =
  await navigator.credentials.create({
    publicKey
  });

// Credentials are stored with the user
// identity are sent back and stored on the
// server

// Authenticate again using server challenge
const authCredential =
  await navigator.credentials.get({
    publicKey
  });

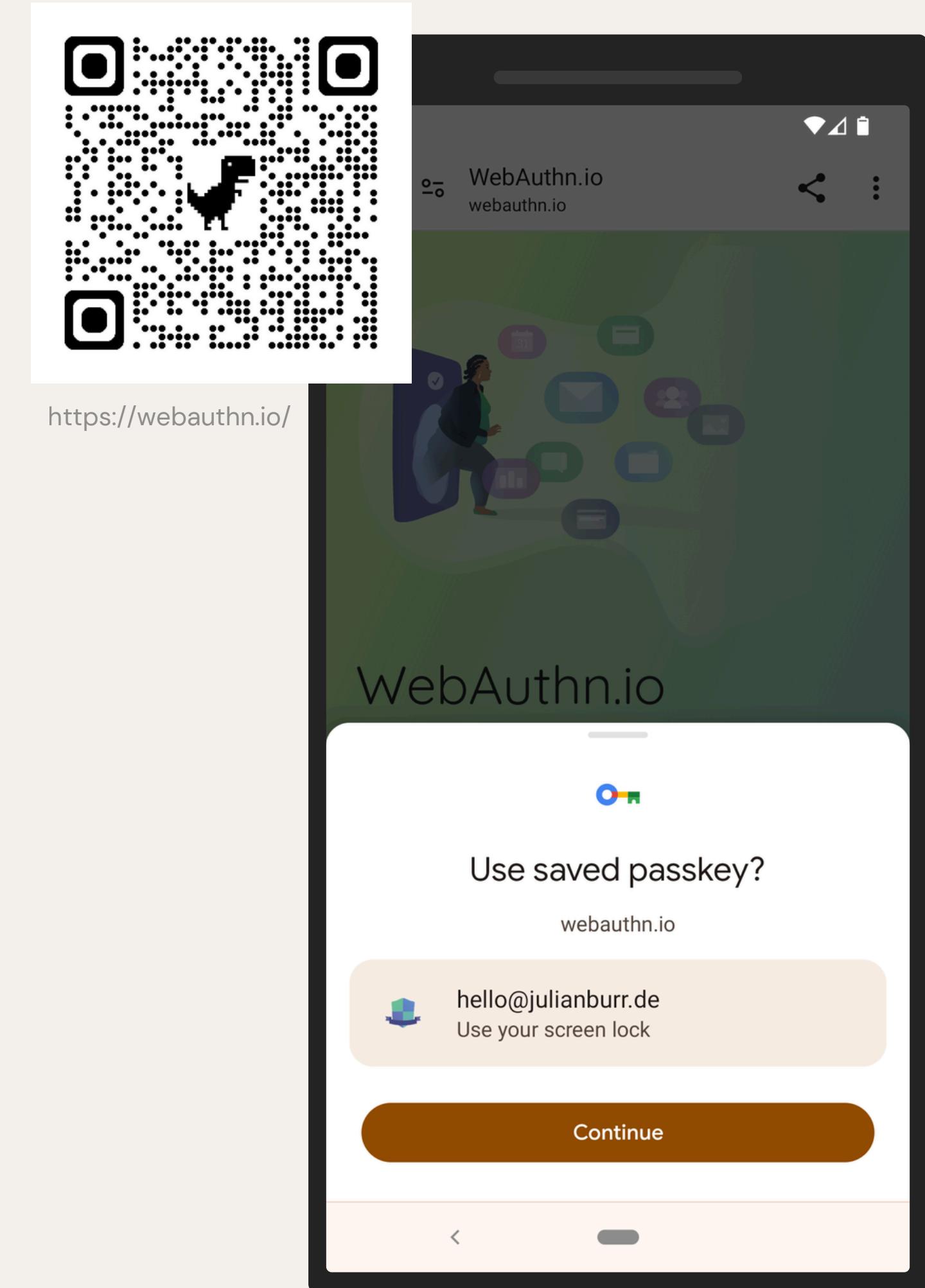
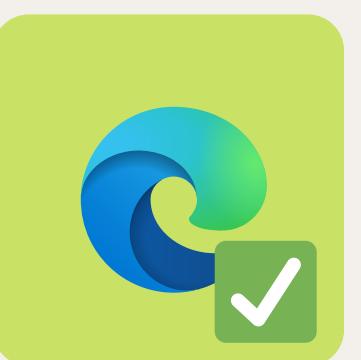
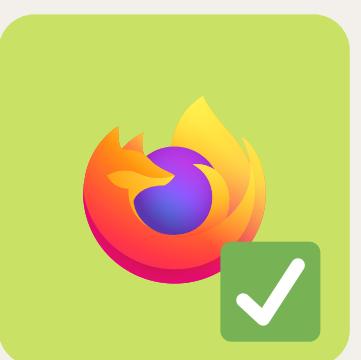
// Use stored public key to verify validity
// of auth credentials
```

WEB AUTHN API

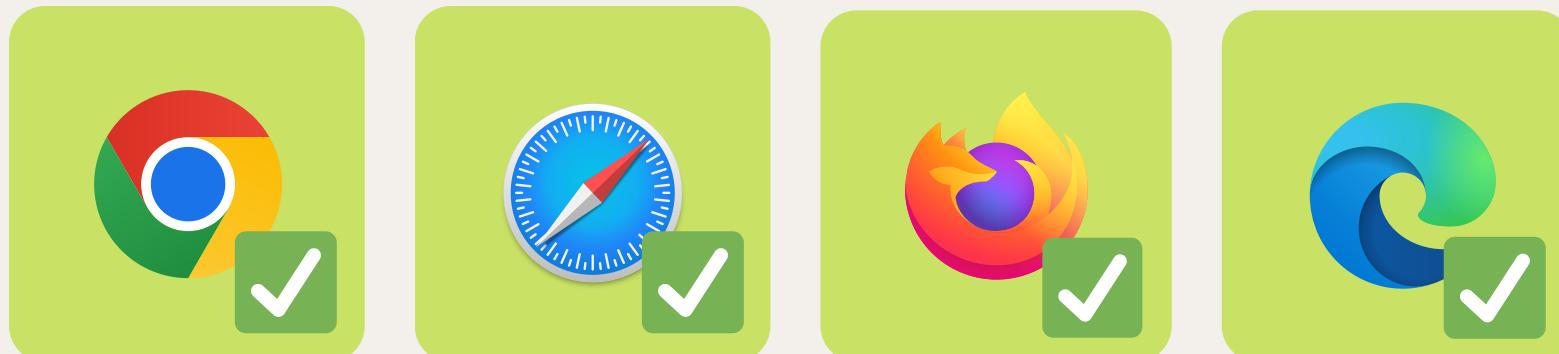


```
import * as s from "@simplewebauthn/server";\n\n// Register\nconst options =\n    await s.generateRegistrationOptions(opts);\n// *.challenge\n// Store the challenge to compare later\n\n// Verify\nconst verification =\n    await s.verifyRegistrationResponse(opts);\n// *.registrationInfo\n// Store on the server\n\n// Authenticate\nconst verification =\n    await s.verifyAuthenticationResponse(opts);\n// *.authenticationInfo
```

WEB AUTHN API



WEB AUTHN API



WebAuthn.io

webauthn.io

WebAuthn

A demo of the WebAuthn API

hello@julianburr.de

Register

Advanced Settings

Create a passkey

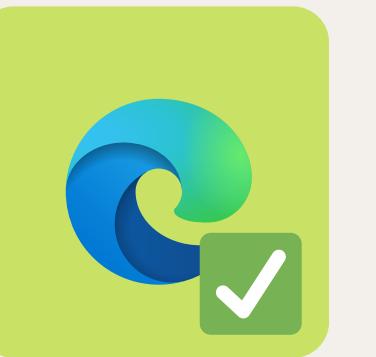
Choose how you want to create a passkey for webauthn.io

- iCloud Keychain
- Pixel 3
- Use a different phone or tablet
- Your Chrome profile
- USB security key

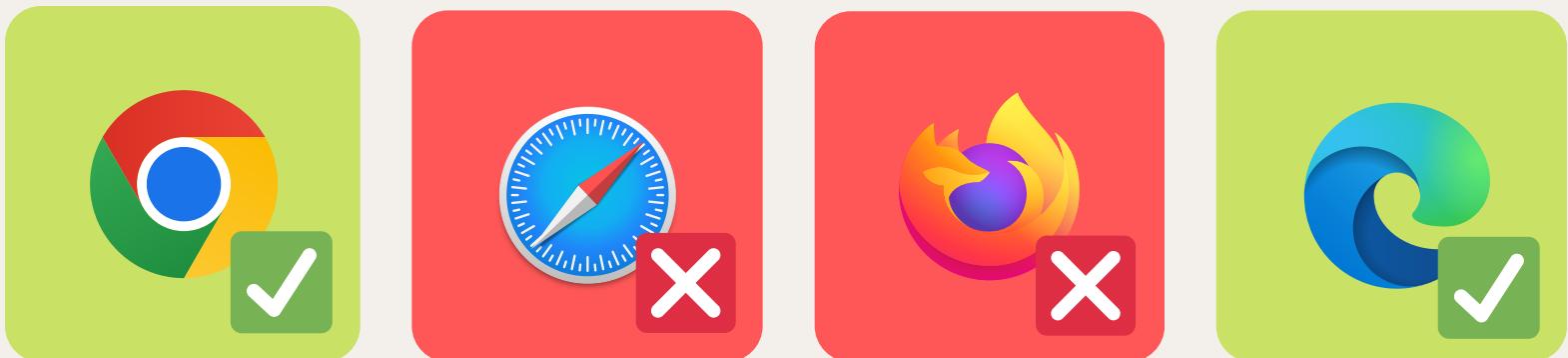
Manage devices

Cancel

WEB OTP API

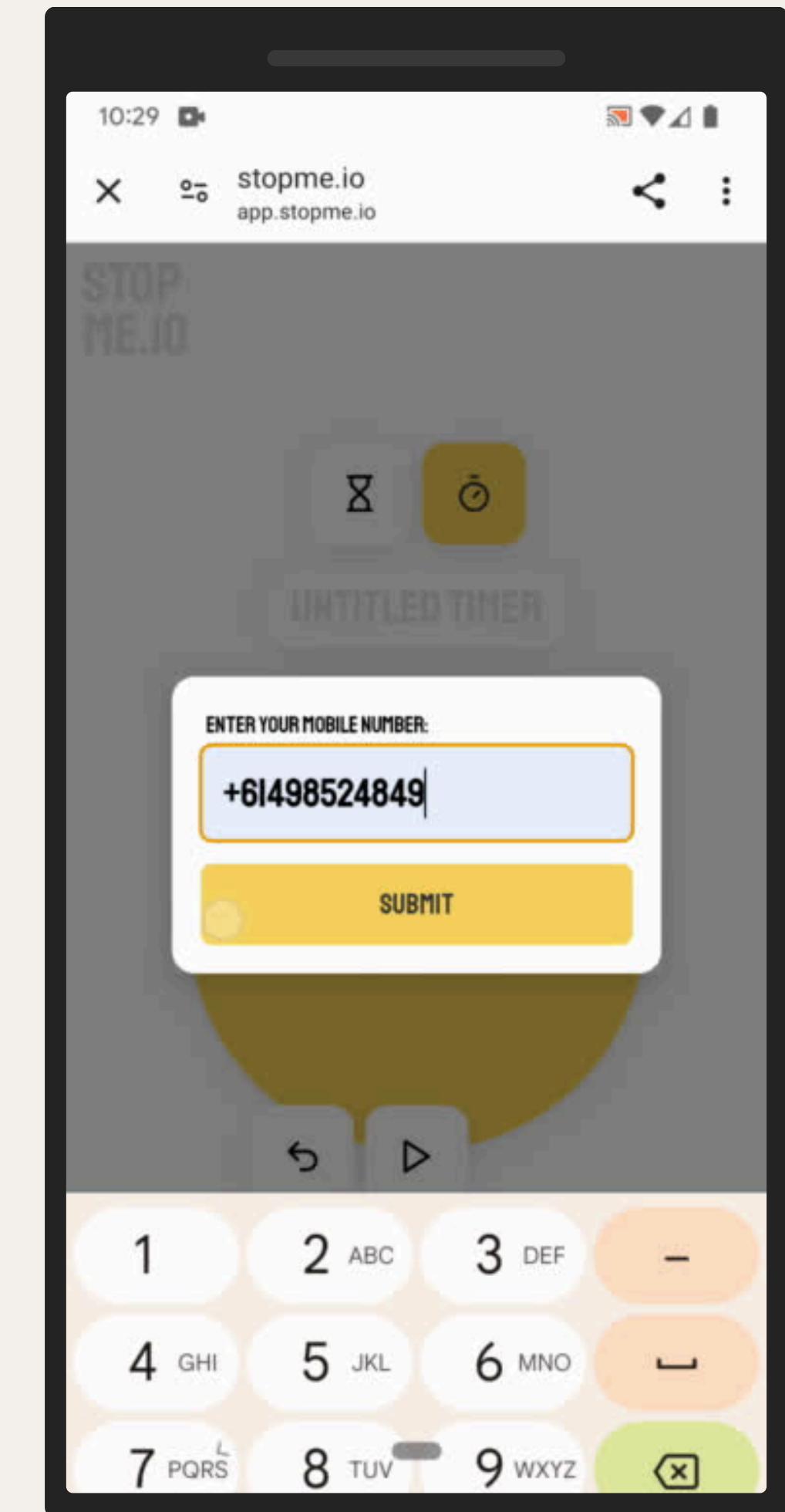


WEB OTP API

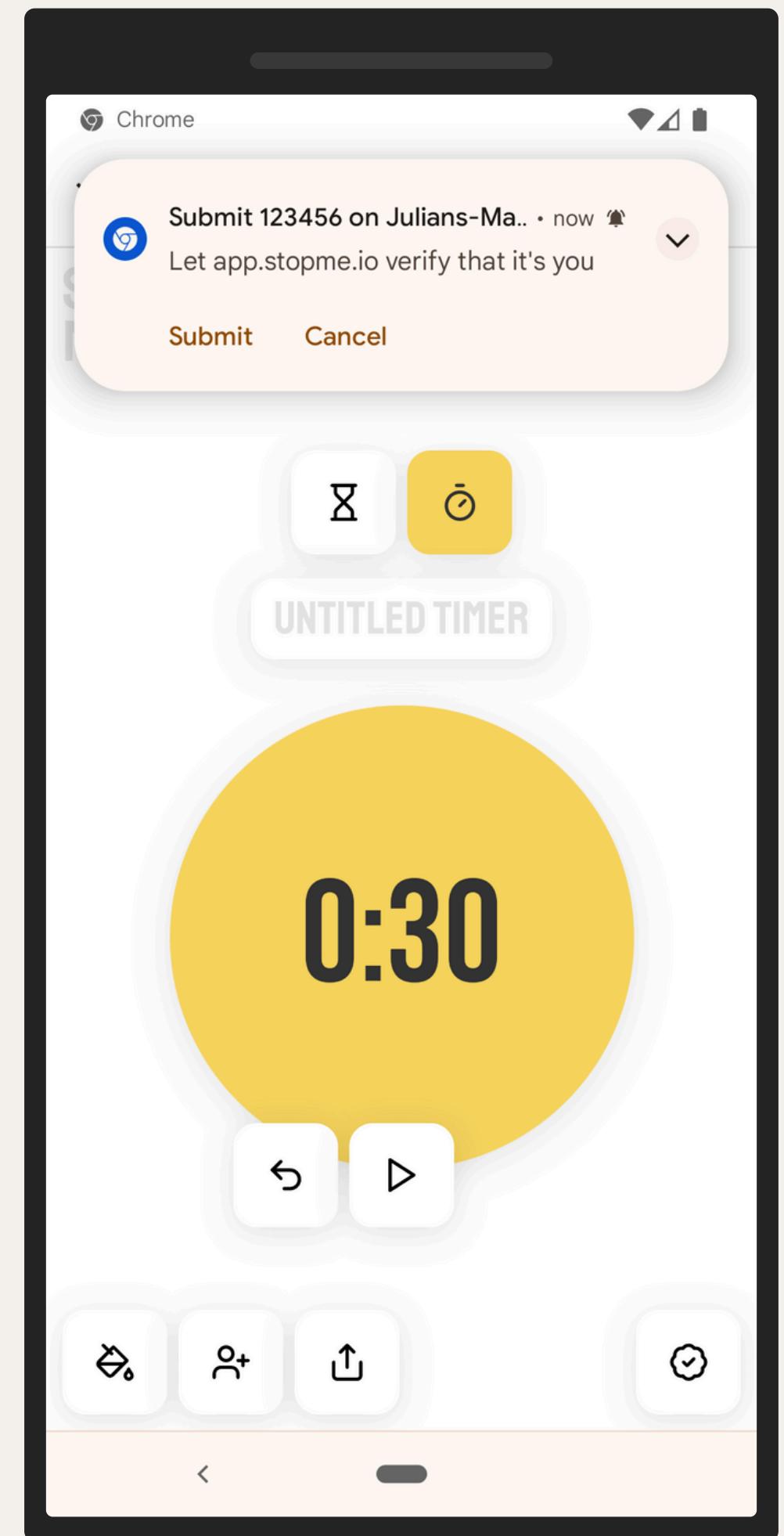
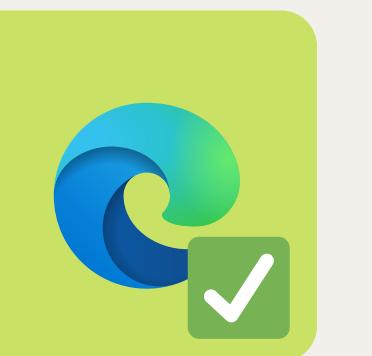


```
navigator.credentials  
  .get({  
    otp: { transport: ["sms"] },  
    signal: ac.signal,  
  })  
  .then((otp) => {  
    // *.code  
  })  
  .catch((e) => {  
    // Handle any errors  
});  
  
// Format of the SMS so it can be processed  
// Your verification code is 123456.\n\n  
// @app.stopme.io #123456
```

WEB OTP API



WEB OTP API



HONOURABLE Mentions

DEBUGGING

CONSOLE

PERFORMANCE

MEMORY

DEBUGGING

CONSOLE

PERFORMANCE

MEMORY

```
console.log(...);  
  
console.assert(...);  
console.count(...);  
console.countReset(...);  
  
console.dir(...);  
console.table(...);  
console.group(...);  
console.groupCollapsed(...);  
console.groupEnd(...);  
  
console.time(...);  
console.timeEnd(...);  
  
console.trace(...);  
  
console.clear();
```

DEBUGGING

CONSOLE

PERFORMANCE

MEMORY

```
const time = performance.now();

performance.mark("start");
performance.mark("end", { detail: { ... } });

performance.measure("login", "start", "end");

const observer =
  new PerformanceObserver((list, obj) => {
    list.getEntries().forEach((entry) => {
      // *.name
      // *.startTime
      // *.duration
      // *.detail
    });
  });
observer.observe({ type: "resource" });
```

DEBUGGING

CONSOLE

PERFORMANCE

MEMORY

```
const memory = navigator.deviceMemory;  
// Device has at least ${memory}GiB of RAM  
  
const memorySample = await performance  
    .measureUserAgentSpecificMemory();  
// *.bytes  
// *.breakdown[ ].bytes  
// *.breakdown[ ].attribution  
// *.breakdown[ ].types
```

GETTING CREATIVE

MIDI

MEDIA CAPTURE

WEB RTC

CODECS

GETTING CREATIVE

MIDI

MEDIA CAPTURE

WEB RTC

CODECS

```
navigator
  .requestMIDIAccess()
  .then((midiAccess) => {
    // *.inputs[].id
    // *.inputs[].type
    // *.inputs[].name
    // *.inputs[].manufacturer
  });

const callback = (e) => {
  // *.timeStamp
  // *.data
}

midiAccess.inputs[index].onmidimessage =
  callback;
```

GETTING CREATIVE

MIDI

MEDIA CAPTURE

WEB RTC

CODECS

```
// Capture media from input devices
const stream = await navigator.mediaDevices
    .getUserMedia({ video: true, audio: false });
```

```
const videoEl =
    document.getElementById("video");
videoEl.srcObject = stream;
videoEl.play();
```

```
// Capture video from element or canvas
const el =
    document.getElementById("record-me");
const stream = await el.captureStream();
el.play();
```

GETTING CREATIVE

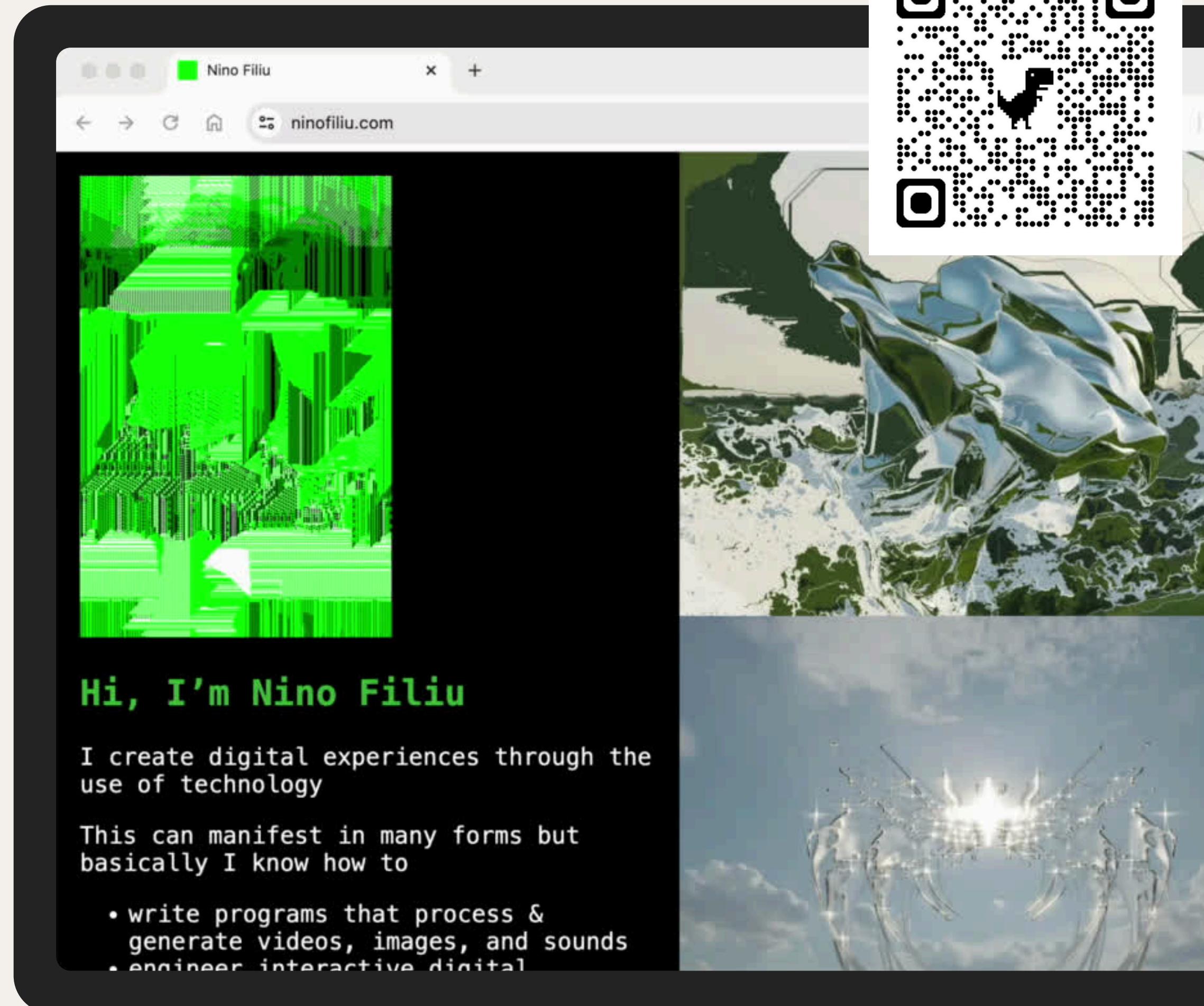
MIDI

MEDIA CAPTURE

WEB RTC

CODECS

@ninofiliu -- <https://ninofiliu.com/>



SHARING CONTENT

BLUETOOTH

USB API

SHARING CONTENT

BLUETOOTH

USB API

```
const perms = await navigator.permissions
  .query({ name: "bluetooth" });

const isAvailable = await navigator.bluetooth
  .getAvailability();

navigator.bluetooth.addEventListener(
  "availabilitychanged",
  (e) => {
    // Do something
  }
);

const devices =
  await navigator.bluetooth
    .getDevices();
```

SHARING CONTENT

BLUETOOTH

USB API

```
const device = await navigator.bluetooth
  .requestDevice({ filters: [...] });
// *.name

const server = await device.gatt.connect();
const service = await server
  .getPrimaryService("health_thermometer");

const char = await service
  .getCharacteristic("measurement_interval");
```

SHARING CONTENT

BLUETOOTH

USB API

```
const device = await navigator.usb
  .requestDevice({ filters: [...] });
// *.productName
// *.manufacturerName

const data = new Uint8Array(...);
await device.transferOut(endpoint, data);
```

THE NEXT GENERATION

NAVIGATION API

URL PATTERNS

POPOVER / DIALOG

VIEW TRANSITION

THE NEXT GENERATION

NAVIGATION API

URL PATTERNS

POPOVER / DIALOG

VIEW TRANSITION

THE NEXT GENERATION

NAVIGATION API

URL PATTERNS

POPOVER / DIALOG

VIEW TRANSITION

```
const p = new URLPattern({ pathname: "/foo" });
p.test("https://example.com/books")); // true
```

```
const p =
  new URLPattern({ pathname: "/books/:id" });
const match =
  p.exec("https://example.com/books/123");
// *.pathname.groups.id = 123
```

```
const p =
  new URLPattern(
    "/books/:id(\d+)",
    "https://example.com"
);
```

THE NEXT GENERATION

NAVIGATION API

URL PATTERNS

POPOVER

VIEW TRANSITION

```
<button popovertarget="mypopover">  
  Toggle the popover  
</button>  
  
<div id="mypopover" popover>  
  Popover content  
</div>  
  
document.addEventListener("keydown", (e) => {  
  if (e.key === "h") {  
    mypopover.togglePopover();  
  }  
});
```

THE NEXT GENERATION

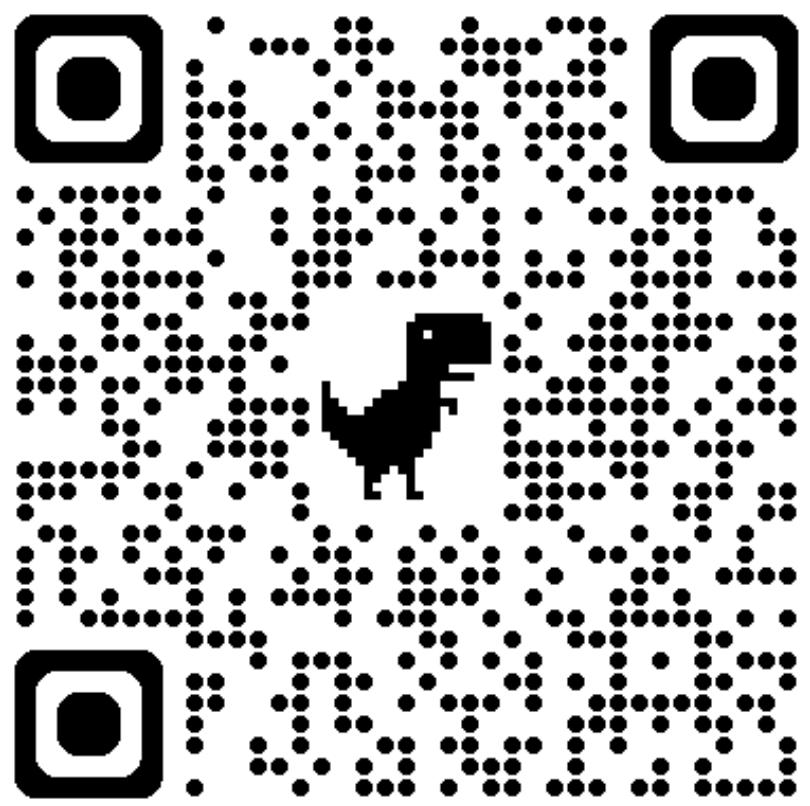
NAVIGATION API

URL PATTERNS

POPOVER

VIEW TRANSITION

```
function updateView(e) {  
  e.preventDefault();  
  const details =  
    e.target.closest("details");  
  
  if (!document.startViewTransition) {  
    // Fallback for older browsers  
    details.toggleAttribute("open")  
  }  
  
  document.startViewTransition(() =>  
    details.toggleAttribute("open")  
  );  
}
```



THANKS!

@JBURR90 / JULIAN BURR

**[HTTPS://WWW.JULIANBURR.DE/
SLASHNEW-2024-SLIDES.PDF](https://www.julianburr.de/slashnew-2024-slides.pdf)**