Vijfhart,

dat klopt voor jou!

HTML, CSS & JS with PWA

Introduction round

Role / background within KPN and before 💼

Hobbies / interests / family / pets 🐟 🐈 👶 🏂

What do you hope to learn? 👩‍🏫

Before we start…

1. Please keep your camera on 😁
2. Questions? Just ask! Feel free to interrupt. 🗣
3. Two 10 minutes breaks. ☕

Overview

1. Overview web (development)
2. HTML
3. CSS
4. JavaScript
5. Progressive Web Apps

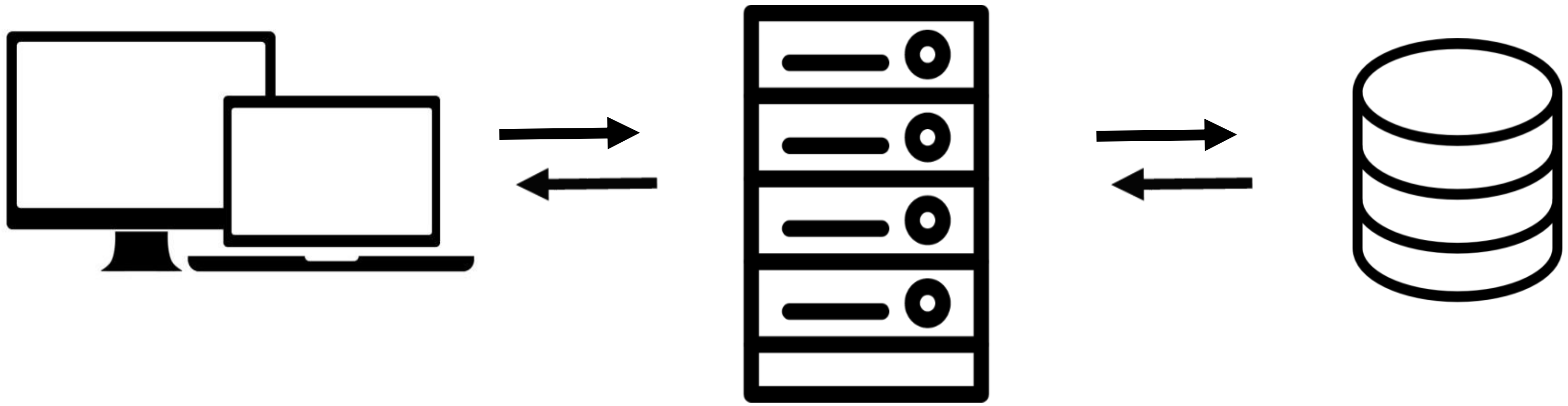
Tooling

1. Preferably an IDE (Visual Studio Code, NotePad++, Atom, Sublime etc)

2. Alternatively: online environment such as Jsfiddle or CodePen

3. Browser (Chrome, Firefox or Edge)

Overview web (development)

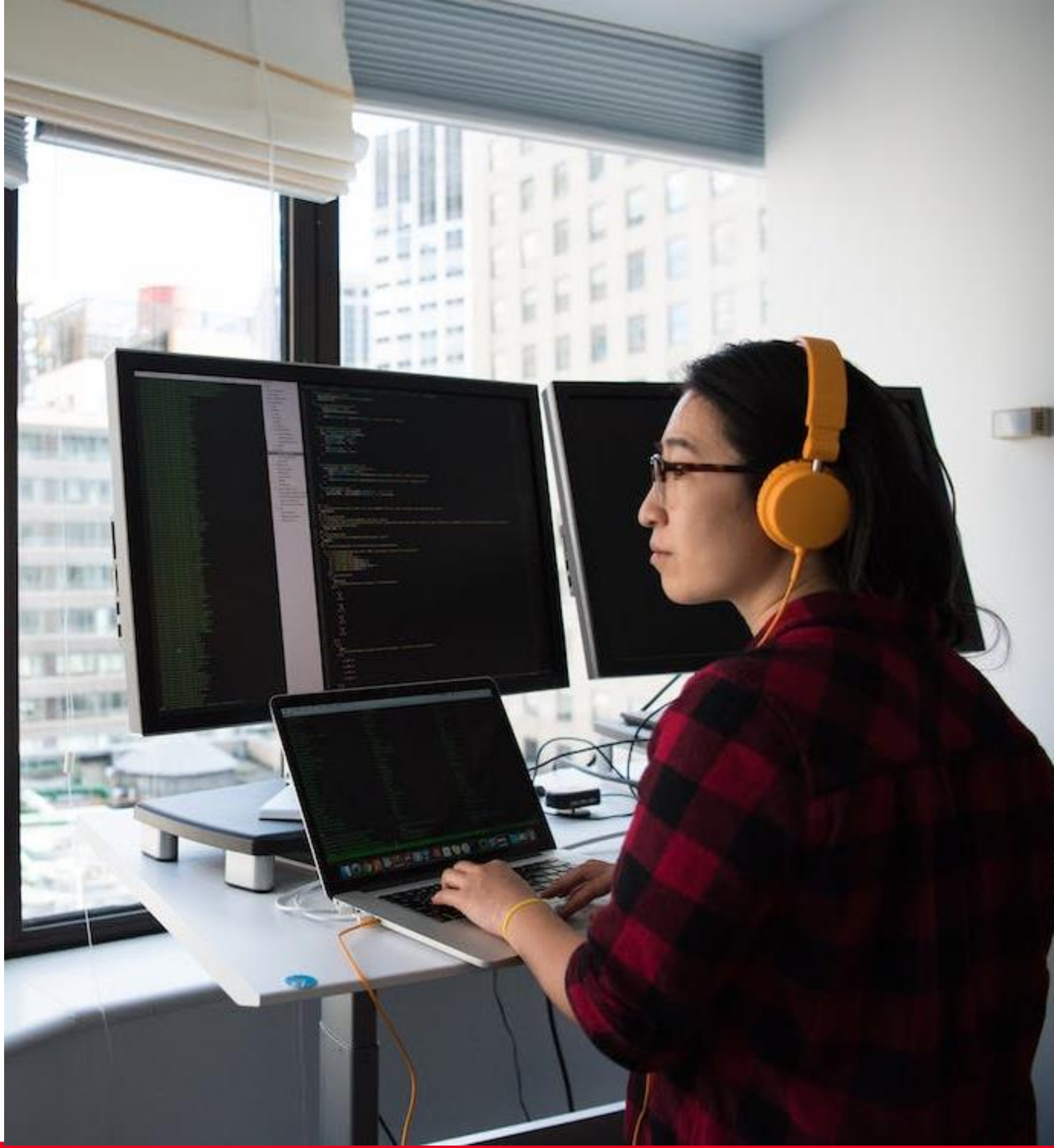


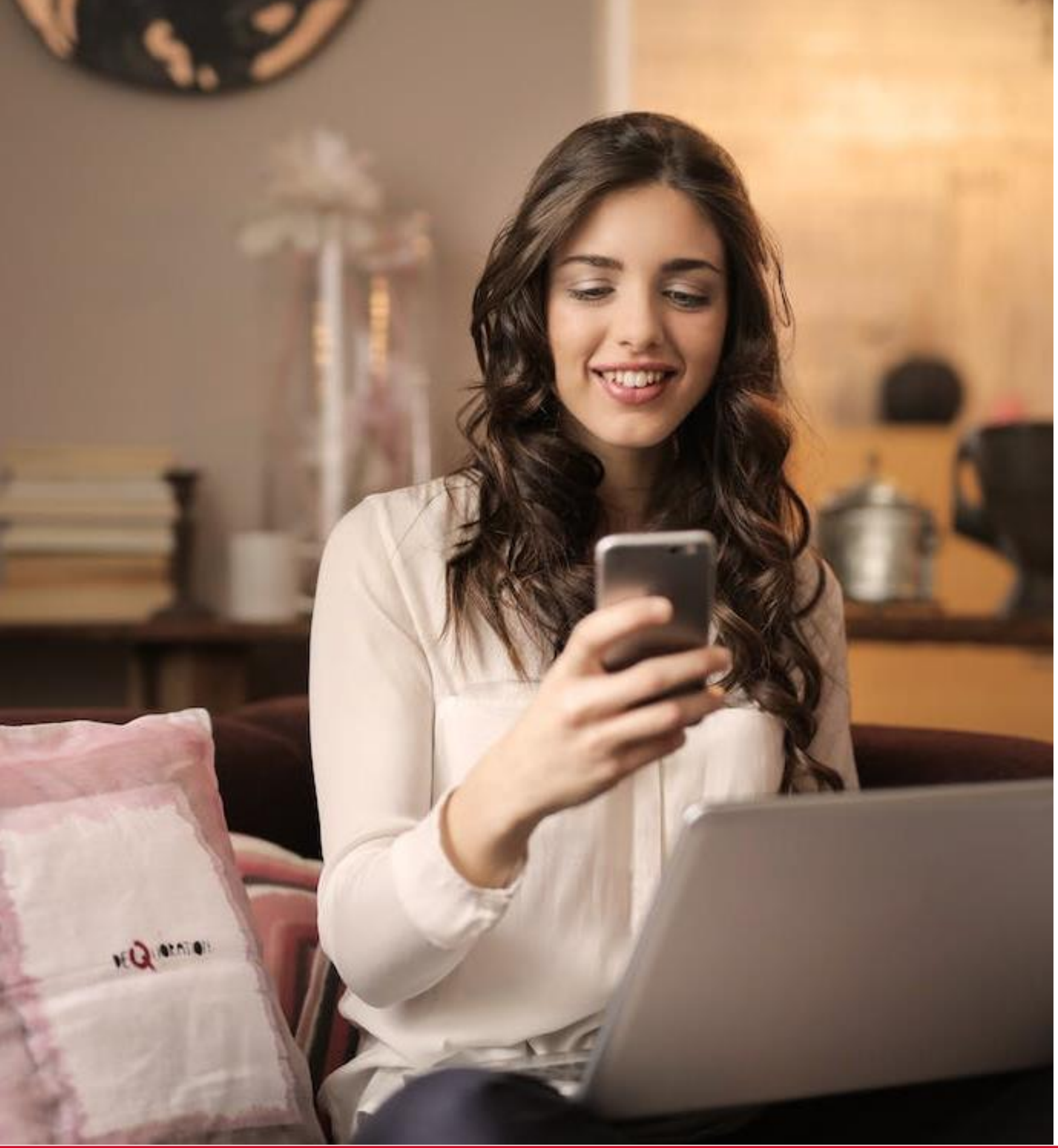
HTML, CSS and JS

HTML (Hypertext Markup Language): The standard markup language used to structure content on the web.

CSS (Cascading Style Sheets): The style sheet language used to define the visual presentation of a web page.

JS (JavaScript): The programming language that allows for dynamic and interactive elements on a web page.





Progressive Web App

Progressive web apps, also known as PWAs, are apps that are built with web technologies but appear to be a platform specific app.

They are transforming the web:

- It's becoming a mobile-first platform.
- Applications run faster.
- It's possible to work offline.
- Applications can be added to the mobile home screen.

HTML

What is HTML?

- Hyper Text Markup Language
- HTML determines what is on the web page (another word for website), for example text, buttons, forms and images.
- Using special words between < and >, the internet browser can display the content of the page.

Why do we need HTML?

HTML is the content of web pages

This includes:

- Text
- Headers
- Forms
- Buttons
- Images
- Video
- And a lot more!

Basic HTML document

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Basic HTML Document</title>
</head>
<body>
  <h1>Welcome to My Basic HTML Page</h1>
</body>
</html>
```

HTML elements

HTML element is a component on a web page

```
<startTag>Content</endTag>
```

There are many different types of HTML elements, for example: p, form, div, h1

HTML elements can contain other HTML elements

More information and more elements: https://www.w3schools.com/html/html_elements.asp

HTML attributes

- HTML attributes provide additional information about an element and help define its properties or behavior.
- They are always specified in the start tag (or the opening tag) of an HTML element.
- They are usually presented in name/value pairs like name="value".

```
<a
href="https://www.example.com">Visit
Example</a>
```

```
<img src="path_to_image.jpg"
alt="Description of Image">
```

Attributes we'll need for CSS and JavaScript

- Class: HTML elements can have a class attribute. This can be used to group certain elements and give it a certain layout or behavior.

```
<p class="special">Some text</p>
```

- Id: HTML element can have an id. This must be a unique id for the page.

```
<p id="test">Some text</p>
```

HTML element

```
<tag attr="value">
  <inner>
    Some text
  </inner>
  <inner>
    Some more text
  </inner>
</tag>
```

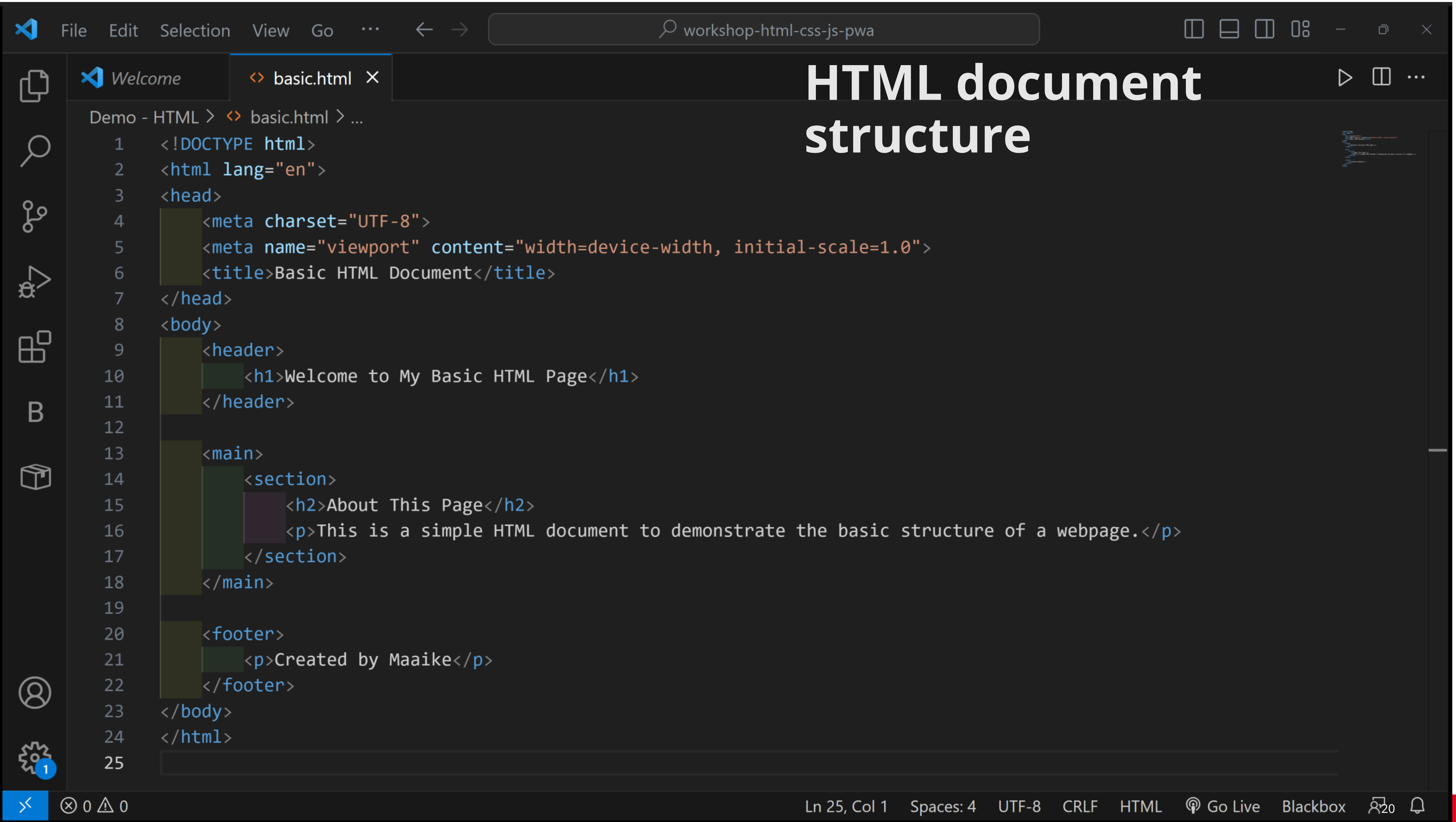
HTML element

```
<tag attr="value">inner HTML</tag>
```

Nesting HTML elements

- Nested elements are HTML elements inside other elements.
- Outer element is the "parent"; inner element is the "child".
- Enables complex structures, like lists within lists.
- Ensure proper opening and closing to avoid display errors.

HTML document structure


File Edit Selection View Go
workshop-html-css-js-pwa
Welcome
basic.html
Demo - HTML > basic.html > ...
Ln 25, Col 1 Spaces: 4 UTF-8 CRLF HTML Go Live Blackbox

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Basic HTML Document</title>
</head>
<body>
    <header>
        <h1>Welcome to My Basic HTML Page</h1>
    </header>

    <main>
        <section>
            <h2>About This Page</h2>
            <p>This is a simple HTML document to demonstrate the basic structure of a webpage.</p>
        </section>
    </main>

    <footer>
        <p>Created by Maaike</p>
    </footer>
</body>
</html>
```

Exercise

CSS

What is CSS?

Cascading Style Sheets

Used for creating the layout of the web page

With CSS we define rules for certain HTML elements

We can do a lot of things with CSS! Change the font, color, position, size, shadows, borders, shape… We can even create complete animations!

Adding CSS to your page

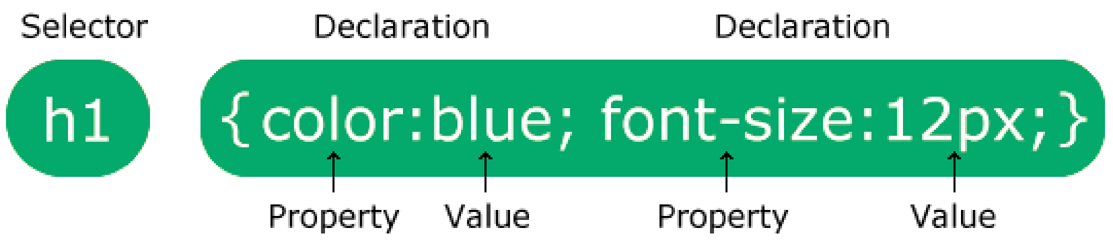
Different ways to add CSS to your page:

- Style attribute on an element
- Style HTML element
- Link a separate page (preferred way)

To link a separate page, in the head tag add:

```
<link rel="stylesheet" href="name-
css-file.css">
```

CSS Syntax



https://www.w3schools.com/css/css_syntax.asp

CSS Selectors

We'll keep it simple here:

- By tag name:

```
p {
  color: red;
}
```

- By class:

```
.special {
  color: green;
}
```

- By id:

```
#element2 {
  color: yellow;
}
```

CSS properties

Used for adjusting a certain part of the layout

More info:

https://www.w3schools.com/css/css_colors.asp

Specificity

When CSS declarations are conflicting, the one with the most points win:

- Id: 100
- Class: 10
- Tagname: 1

The most specific declaration determines the layout that shows. That's called specificity.

CSS Demo

Exercise

Create an HTML page

JavaScript

What is JavaScript?

Scripting language

That can be used server side (node.js) and client side

Client side: we’ll use it to make our pages interactive

We could do a 7 week course on JavaScript...

But here's the 20 minutes version:

- Connecting a JS file
- Writing a function
- Triggering a function onclick
- Selecting elements by id
- Getting the value from input boxes and dropdowns
- Changing the innerHTML
- Changing the style with JavaScript
- Adding and removing classes with JavaScript

Connecting a JS file

- Link JavaScript file to HTML using the <script> tag.
- Place before the closing </body> tag for performance.
- `<script src="script.js"></script>`

Writing a function

- Functions encapsulate reusable code.
- Define using function keyword.

```
function greet() {
    alert("Hello, World!");
}
```

- You can call this function with:

```
greet()
```

Triggering a function onclick

Use the onclick attribute in HTML.

Calls a JS function when element is clicked.

```
<button onclick="greet()">Click
Me</button>
```

Note: we won’t deal with proper event handlers in this brief overview

Selecting elements by id

- Use document.getElementById().
- Requires unique id attribute in HTML.

HTML:

```html
<div id="myDiv">Hello</div>
```

JavaScript:

```javascript
let element = document.getElementById("myDiv");
```

Getting the value from input boxes and dropdowns

Use the value property for inputs.

For dropdowns, access selected option's value.

HTML:
```
<input id="username" type="text">

<select id="options">

    <option value="1">Option 1</option>

</select>
```

JS:
```
let inputValue =
document.getElementById("username").value;

let dropdownValue =
document.getElementById("options").value;
```

Changing the innerHTML

Use innerHTML property.

Modify content inside an element.

HTML:
```
<p id="demo">Old Text</p>
```

JavaScript:
```
document.getElementById("demo").innerHTML =
"New Text";
```

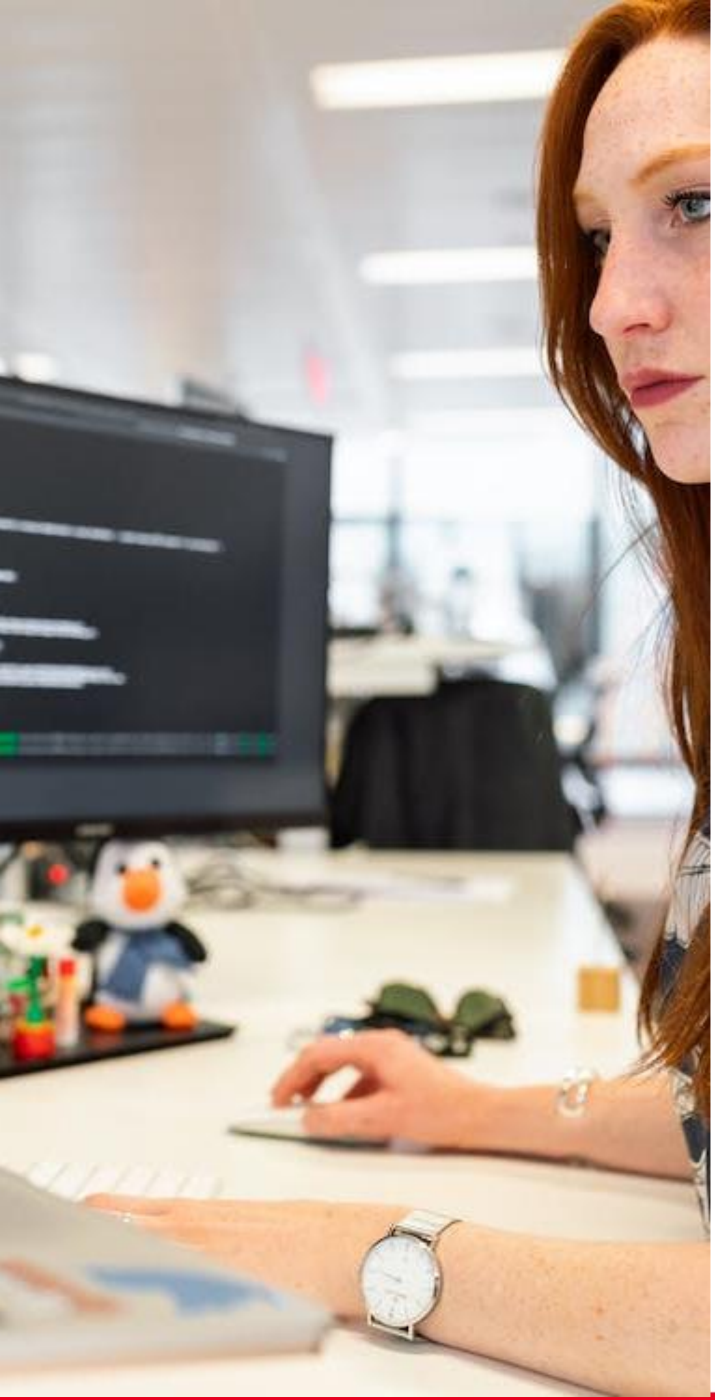
Changing the style with JavaScript

Access styles using style property.

Modify CSS properties in camelCase.

HTML:
```
<div id="box">Content</div>
```

Javascript:
```
document.getElementById("box").style.backgroundColor =
"blue";
```



Adding and removing classes with JavaScript

Use classList property.

Add or remove CSS classes.

HTML:
```
<div id="box" class="oldClass">Content</div>
```

JS:
```
document.getElementById("box").classList.add("newClass");
document.getElementById("box").classList.remove("oldClass");
```

CSS:
```
.newClass {
    color: red;
}
```

JavaScript Demo

Exercise

Progressive Web Apps

Progressive Web Apps

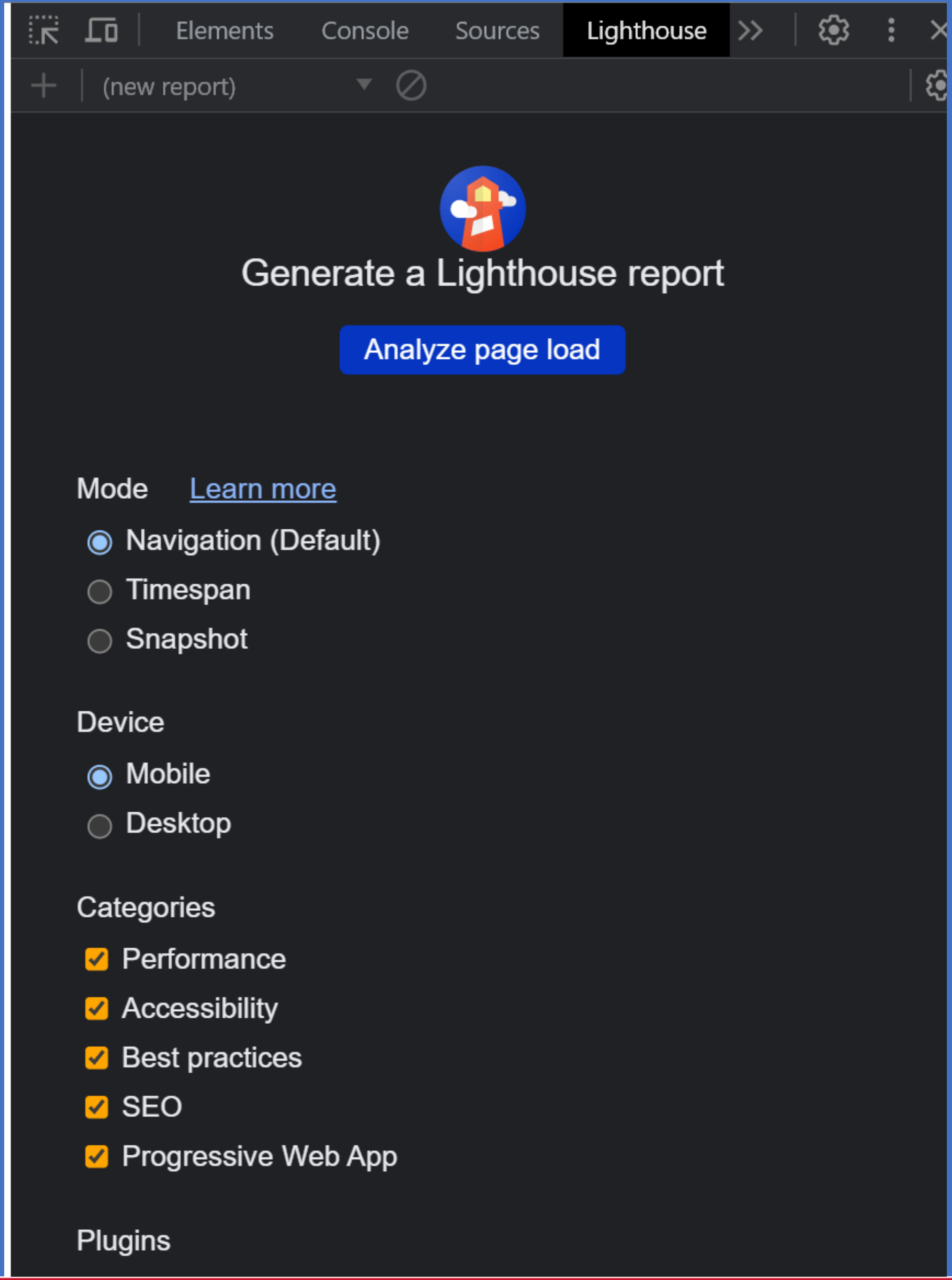
- Websites that deliver a native mobile experience
- Features such as push notifications, camera, geolocation and more is possible
- PWA’s can be installed on devices like normal native apps
- Without connection it can interact with the device (such as camera)
- And without a connection it can receive push notifications

Best part about PWAs?

It’s not that hard to achieve!

We need to add the following:

- manifest.json
- A service worker (js file)
- An icon



Generate a Lighthouse report

Analyze page load

Mode Learn more

[✓ Navigation (Default)]
Timespan
Snapshot

Device

[✓ Mobile]
Desktop

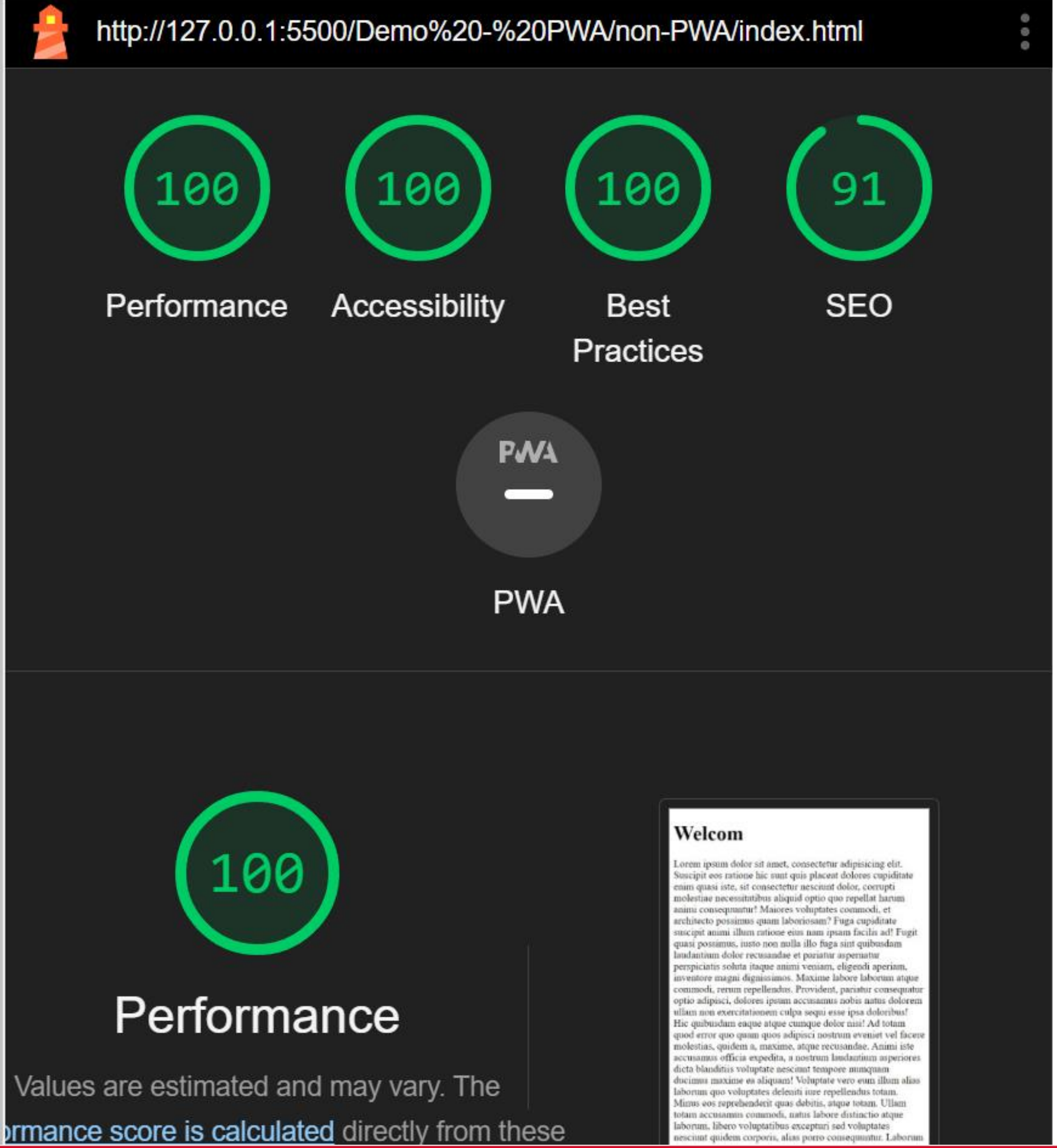
Categories

- ✓ Performance
- ✓ Accessibility
- ✓ Best practices
- ✓ SEO
- ✓ Progressive Web App

Plugins

We can inspect web pages

In the lighthouse tab, we can get information on whether or app is installable as a PWA


http://127.0.0.1:5500/Demo%20-%20PWA/non-PWA/index.html

100 Performance
100 Accessibility
100 Best Practices
91 SEO
PWA

100
Performance

Values are estimated and may vary. The
rmance score is calculated directly from these

Welcom

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Suscipit eos ratione hic sunt quis placeat dolores cupiditate enim quasi iste, sit consectetur nesciunt dolor, corrupti molestiae necessitatibus aliquid optio quo repellat harum animi consequuntur! Maiores voluptates commodi, et architecto possimus quam laboriosam? Fuga cupiditate suscipit animi illum ratione eius nam ipsam facilis ad! Fugit quasi possimus, iusto non nulla illo fuga sint quibusdam laudantium dolor recusandae et pariatur aspernatur perspiciatis soluta itaque animi veniam, eligendi aperiam, inventore magni dignissimos. Maxime labore laborum atque commodi, rerum repellendus. Provident, pariatur consequatur optio adipisci, dolores ipsam accusamus nobis natus dolorem ullam non exercitationem culpa sequi esse ipsa doloribus! Hic quibusdam eaque atque cumque dolor nisi! Ad totam quod error quo quam quos adipisci nostrum eveniet vel facere molestias, quidem a, maxime, atque recusandae. Animi iste accusamus officia expedita, a nostrum laudantium asperiores dicta blanditiis voluptate nesciunt tempore numquam ducimus maxime ea aliquam! Voluptate vero eum illum alias laborum quo voluptates deleniti iure repellendus totam. Minus eos reprehenderit quas debitis, atque totam. Ullam totam accusamus commodi, natus labore distinctio atque laborum, libero voluptatibus excepturi sed voluptates nesciunt quidem corporis, alias porro consequuntur. Laborum

Progressive Web App

Currently, our app is not a PWA.

However, it is performing well and it is accessible. So we can skip the steps were we need to make it fast and accessible.

But it's not a PWA…

Let's change that!

Register a serviceworker

We add a script that checks whether the browser supports serviceWorker, and if that’s the case, we add it.

1. Add to HTML:

```
<script src="registerSW.js"></script>
```

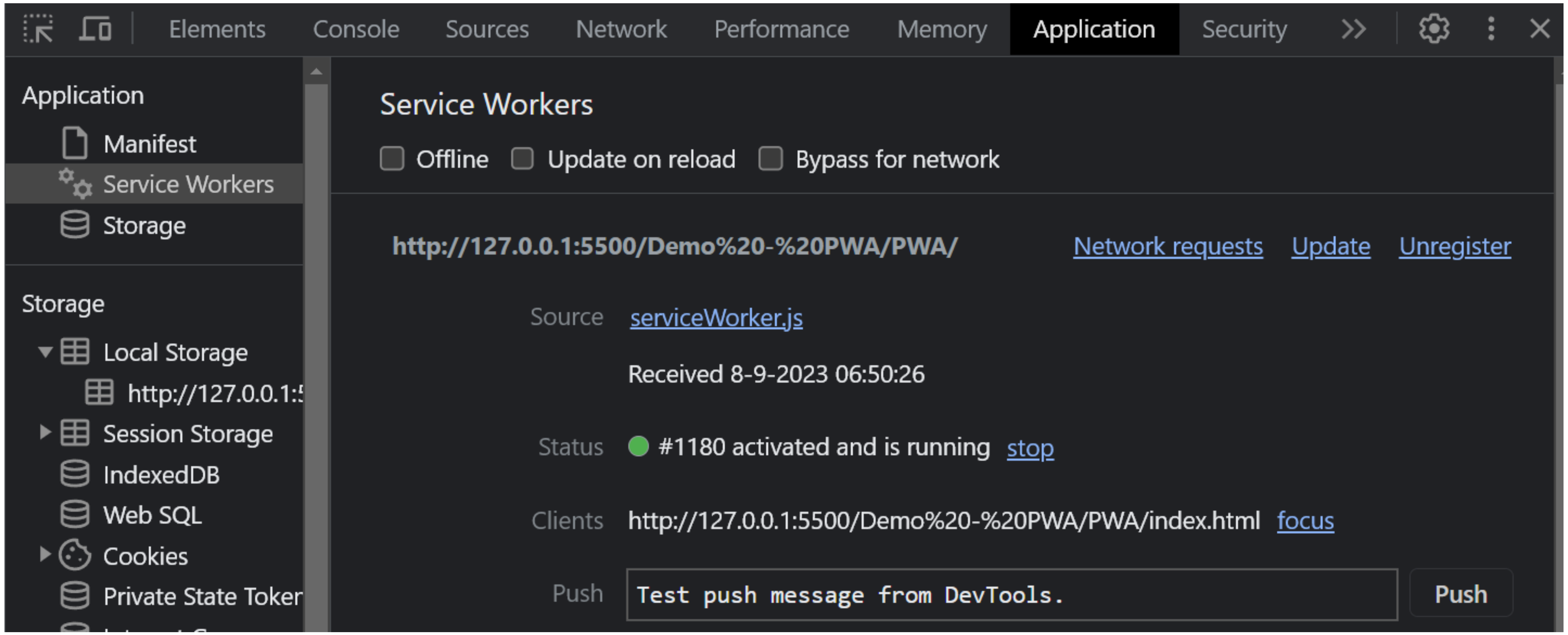
2. Here’s the registerSW.js:

```
if("serviceWorker" in navigator) {
    navigator.serviceWorker.register("serviceWorker.js");
}
```

3. For now, add an empty serviceWorker.js

Verify in the browser

We can see that the serviceWorker is registered!



Service Workers

☐ Offline ☐ Update on reload ☐ Bypass for network

http://127.0.0.1:5500/Demo%20-%20PWA/PWA/ Network requests Update Unregister

Source serviceWorker.js

Received 8-9-2023 06:50:26

Status ● #1180 activated and is running stop

Clients http://127.0.0.1:5500/Demo%20-%20PWA/PWA/index.html focus

Push Test push message from DevTools. Push

Progressive Web App

We will use workbox to do the heavy lifting for us.

This is a library to easily build PWAs.

We will use it to cache the URLs in our app so that we can view it online and generate the files we need.

You can install workbox like this:

```
npm install workbox-cli
```

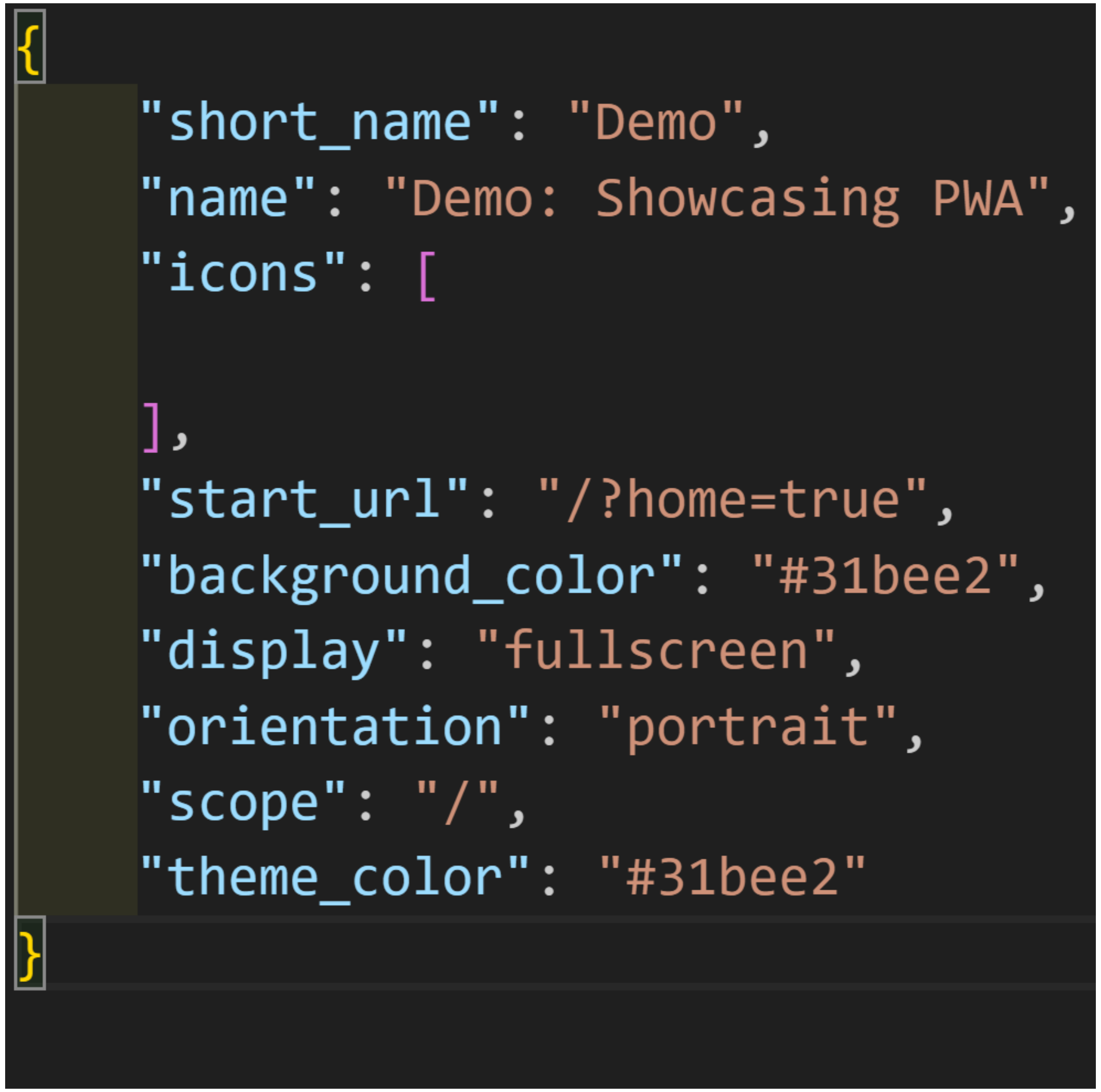
Or import it with cdn:

```
importScripts(

'https://storage.googleapis.com/workbox-cdn/releases/6.4.1/workbox-sw.js'

);
```

Manifest.json



```
{
    "short_name": "Demo",
    "name": "Demo: Showcasing PWA",
    "icons": [

    ],
    "start_url": "/?home=true",
    "background_color": "#31bee2",
    "display": "fullscreen",
    "orientation": "portrait",
    "scope": "/",
    "theme_color": "#31bee2"
}
```

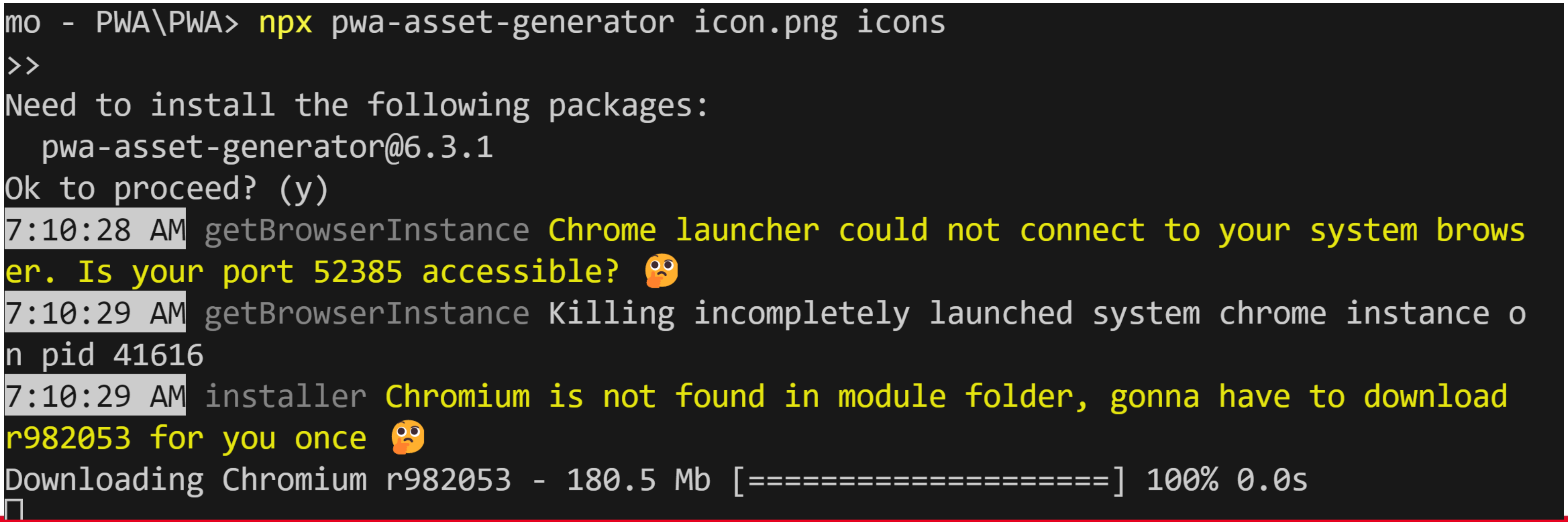
- Add the file manifest.json

- Make sure that the HTML references the manifest.json

Icon

- We need to provide many icons in the icon array
- We’ll run a tool for this:

```
npx pwa-asset-generator icon.png
icons
```



```
mo - PWA\PWA> npx pwa-asset-generator icon.png icons
>>
Need to install the following packages:
  pwa-asset-generator@6.3.1
Ok to proceed? (y)
7:10:28 AM getBrowserInstance Chrome launcher could not connect to your system brows
er. Is your port 52385 accessible? 🤔
7:10:29 AM getBrowserInstance Killing incompletely launched system chrome instance o
n pid 41616
7:10:29 AM installer Chromium is not found in module folder, gonna have to download
r982053 for you once 🤔
Downloading Chromium r982053 - 180.5 Mb [====================] 100% 0.0s
```

serviceWorker.js

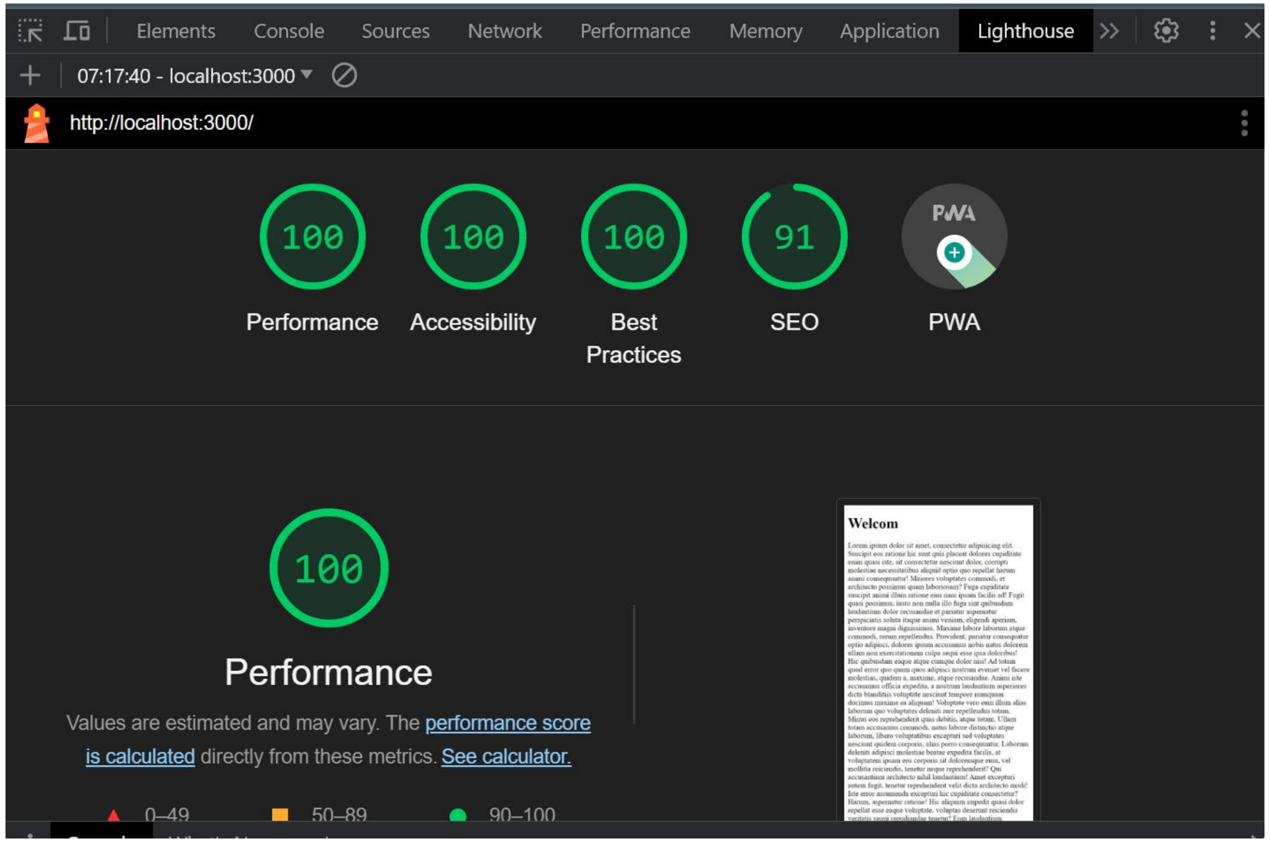
We now need to add our content to serviceWorker.js

```
importScripts(
    'https://storage.googleapis.com/workbox-cdn/releases/6.4.1/workbox-sw.js'
);

workbox.routing.registerRoute(
    ({request}) => request.destination === "image",
    new workbox.strategies.CacheFirst() // great for files that don't change often
)
```

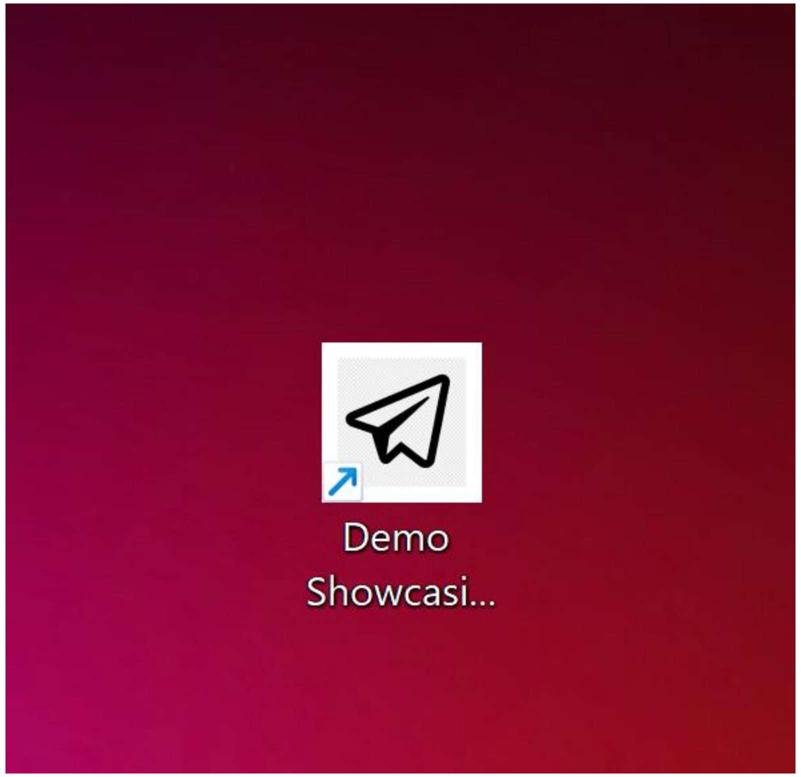
Run the app…

- And do the lighthouse analysis again.
- You’ll see our app now has the PWA badge unlocked!



Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

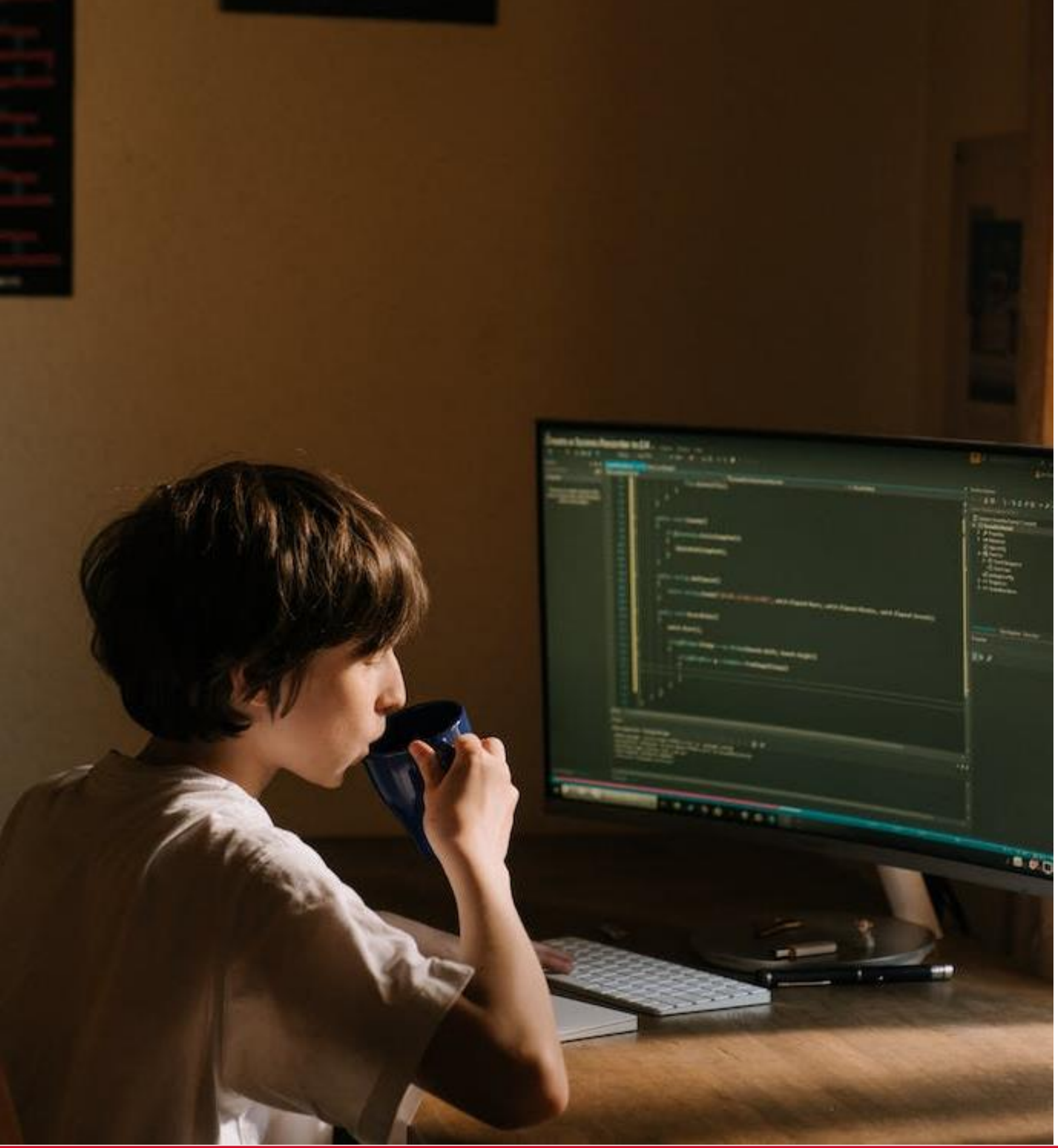
Install on desktop



- Go to the site on edge browser
- Hamburger menu
- App > install as app
- Create desktop shortcut

Progressive Web App Demo

Let’s do these steps together!



There’s a lot more possible…

- Push notifications: Engage users with real-time updates and reminders.
- Background sync: Update content seamlessly, even offline.
- Access native features: Integrate camera, GPS, and other device capabilities.
- Installable: Add to home screen, mimicking native app feel.

Exercise

Let’s put it all together!

1. Extra exercise: HTML, CSS and JS

Wrap up

1. HTML
2. CSS
3. JavaScript
4. Progressive Web Apps

THANK
YOU

Tot ziens!

Volg ons op LinkedIn

Ontvang onze nieuwsbrief

Bekijk onze last minutes

vijfhart.nl

vijfhart

IT-OPLEIDINGEN