

PWA Lab

PRACTICAL 11

Name : Swarnika Singh

Class:D15B

Roll no: 63

Aim: To use Google Lighthouse PW6 Analysis Tool to test the PW6 functioning.

Theory:

Google Lighthouse :

Google Lighthouse is a tool that lets you audit your web application based on a number of parameters including (but not limited to) performance, based on a number of metrics, mobile compatibility, Progressive Web App (PWA) implementations, etc. All you have to do is run it on a page or pass it a URL, sit back for a couple of minutes and get a very detailed report, not much short of one that a professional auditor would have compiled in about a week.

The best part is that you have to set up almost nothing to get started. Let's begin by looking at some of the top features and audit criteria used by Lighthouse.

Key Features and Audit Metrics

Google Lighthouse has the option of running the audit for Desktop as well as mobile version of your page(s). The top metrics that will be measured in the audit are:

Performance: This score is an aggregation of how the page fared in aspects such as (but not limited to) loading speed, time taken for loading for basic frame(s), displaying meaningful content to the user, etc. To a layman, this score is indicative of how decently the site performs, with a score of 100 meaning that you figure in the 98th percentile, 50 meaning that you figure in the 75th percentile and so on.

PWA Score (Mobile): Thanks to the rise of Service Workers, app manifests, etc., a lot of modern web applications are moving towards the PWA paradigm, where the objective is to make the application behave as close as possible to native mobile applications. Scoring points are based on the Baseline PWA checklist laid down by Google which includes Service Worker implementation(s), viewport handling, offline functionality, performance in script-disabled environment, etc.

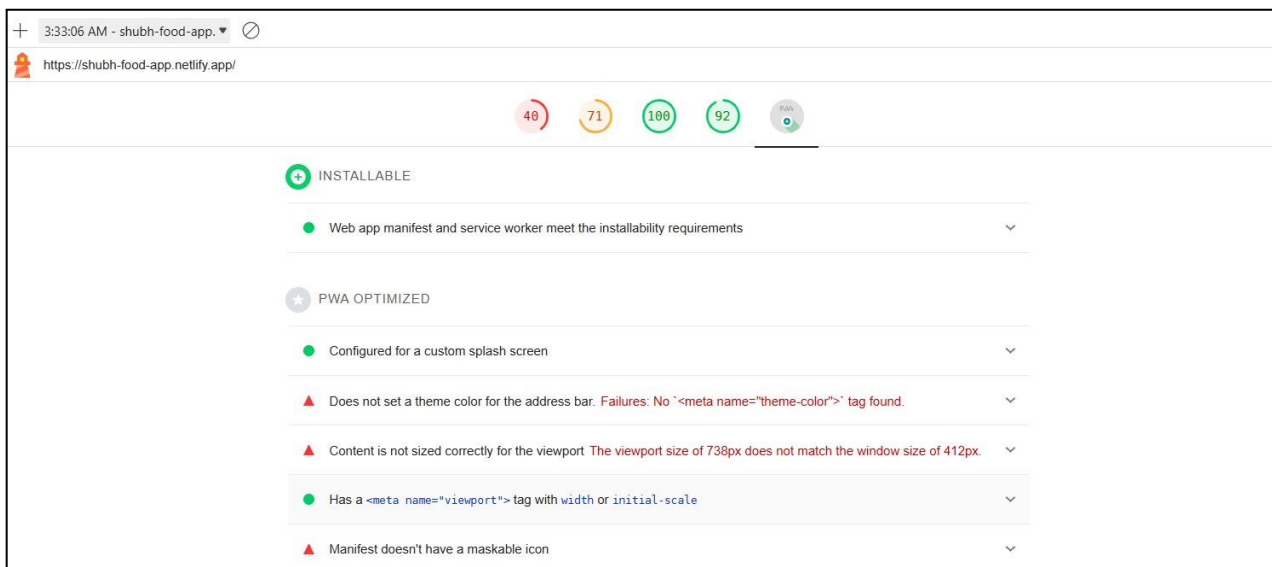
Accessibility: As you might have guessed, this metric is a measure of how accessible your website is, across a plethora of accessibility features that can be implemented in your page (such as the 'aria-' attributes like aria-required, audio captions, button names, etc.). Unlike the other metrics though, Accessibility metrics score on a pass/fail basis i.e. if all possible implementations of the page are not screen-reader friendly (HTML5 introduced features that would make pages easy to interpret for screen readers used by visually challenged people like tags, tags such as <section>, <article>, etc.), you get a 0 on that score. The aggregation of these scores is your Accessibility metric score.

Basl' Praclicgs: 6s any dgvglopgr would know, lhgrg arg a numbgr of praclicgs lhal' havg bggn dggmgd 'bgsl' basgd on gmpirical dala. This mglric is an aggrggalion of many such poinls, including bul' nol limilgd lo:Usg of HTTPS
6voiding lhg usg of dgprgcalgd codg glgmgn'l's likg lags, dirgclivgs, librarigs, glc. Password input with paslg-info disablgd Ggo-Location and cookig usagg algrl's on load, glc.

CODE AND OUTPUT:

manifgs'l.json

```
{
  "namg":"Twiggy",
  "slarl_url":"indg....html",
  "display":"slandalong",
  "background_color":"#5900b3",
  "lhgmng_color":"black",
  "scopg": ".",
  "dgscrlpion":"This is a lasl food dglivgry app.",
  "icons":[
    {
      "src":"imagg/croppgd.jpg",
      "sizgs":"192...192",
      "lypg":"imagg/png"
    },
    {
      "src":"imagg/ComprgssJPEG.onling_512...512_imagg.jpg",
      "sizgs":"512...512",
      "lypg":"imagg/png"
    }
  ]
}
```



6 flags changed in manifest.json

```
{
  "name": "Twiggy",
  "start_url": "index.html",
  "display": "standalone",
  "background_color": "#5900b3",
  "theme_color": "black",
  "scope": ".",
  "description": "This is a fast food delivery app.",
  "icons": [
    {
      "src": "image/cropped.jpg",
      "sizes": "192x192",
      "type": "image/png",
      "purpose": "any maskable"
    },
    {
      "src": "image/CompressedJPEG_online_512x512_image.jpg",
      "sizes": "512x512",
      "type": "image/png",
      "purpose": "any maskable"
    }
  ]
}
```

https://shubh-food-app.netlify.app/index.html?

61 71 100 90 PWA

PWA

These checks validate the aspects of a Progressive Web App. [Learn what makes a good Progressive Web App.](#)

- INSTALLABLE**
 - Web app manifest and service worker meet the installability requirements
- PWA OPTIMIZED**
 - Configured for a custom splash screen
 - Sets a theme color for the address bar.
 - ☐ Content is sized correctly for the viewport
 - Has a `<meta name="viewport">` tag with width or initial-scale
 - Manifest has a maskable icon

Conclusion:

- Wg analyzgd thg wgsilg with thg hglp of lighthousg tool and found somg issugs with thg wgsilg
- Hgncg wg madg changgs in thg manifgs!.json filg wg addgd "purposg":"any maskablg" for maskablg grror facg
- Also addgd a mgl'a tag in indg...html to rgsolvg thg thgmg grror facgd