

Projecte UF2 - Optimització de continguts Multimèdia



INDEX

1.Pàgina Plats	2
1.1 Performance and images:	2
2.Pàgina About	4
2.1 Performance and images:	4
2.2 SVG:	5
3.Pàgina Principal/Home	7
3.1 Clip-Path:	7
3.2 Css Animation:	8
5.Footer	9
6.Imagemin i Parcel	10

REPOSITORI:

<https://github.com/Cirvianum-DAW/mp9-pj-uf2-optimitzacio-imatges-AaronLunaa.git>

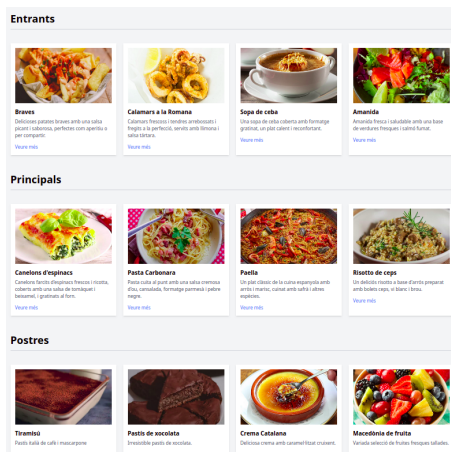
VIDEO IMAGEMIN:

<https://youtu.be/fcfkJNHZoY>

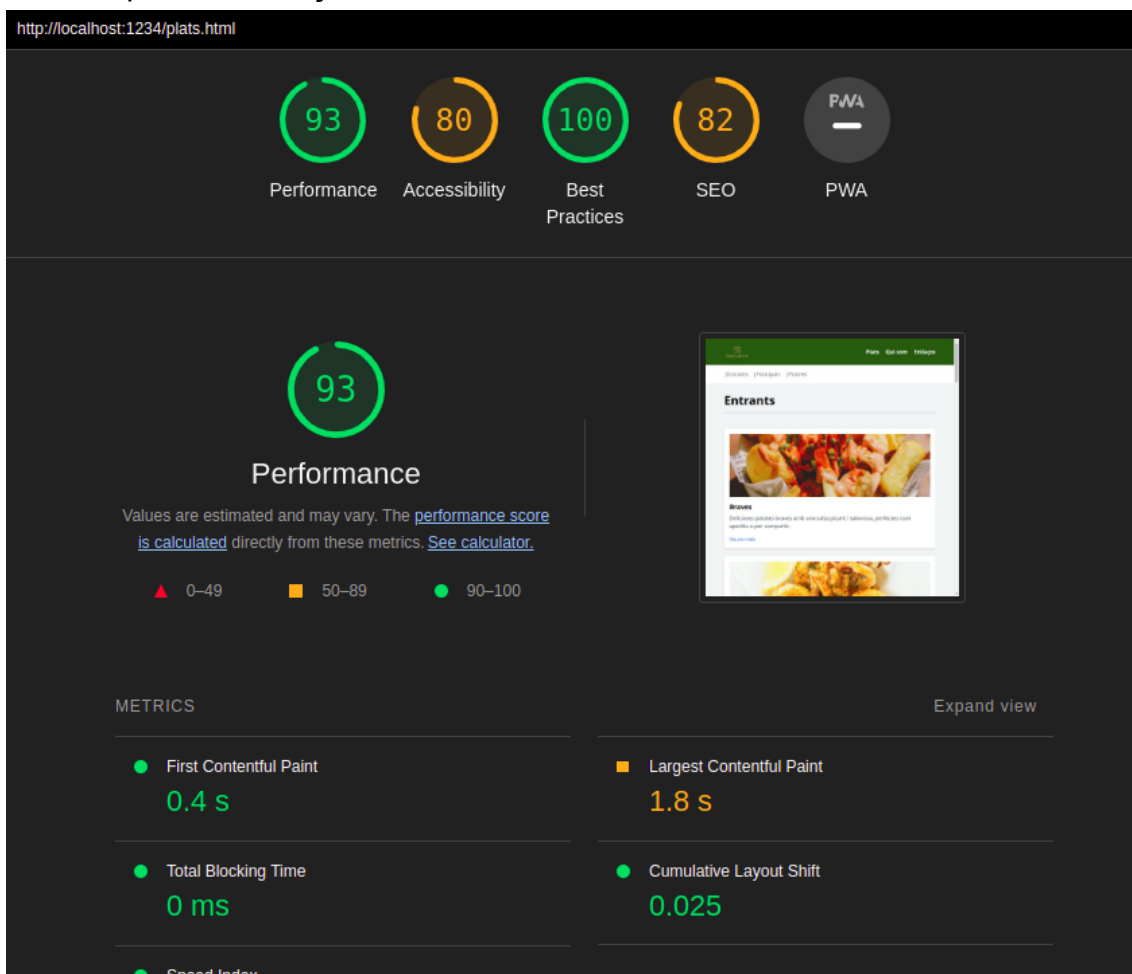
1.Pàgina Plats

1.1 Performance and images:

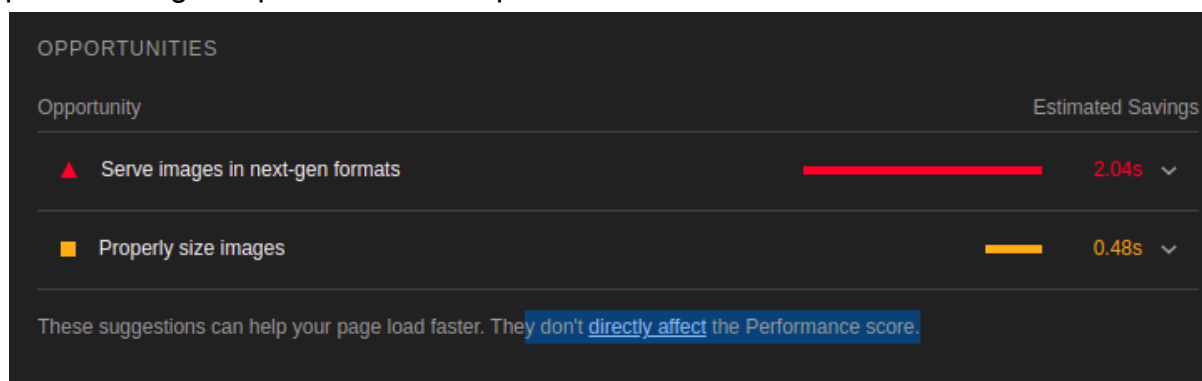
És la pàgina de tota la web que més pes té i més impacte té en el rendiment ja que té en total 12 imatges, en format png totes.



Primer de tot faig ús del lighthouse per comprovar la performance de la web i em surt una puntuació força elevada:



Tot i que apareix una bona puntuació em segeix recomanant que utilitzi webp i avif per les imatges i que millori el seu pes.

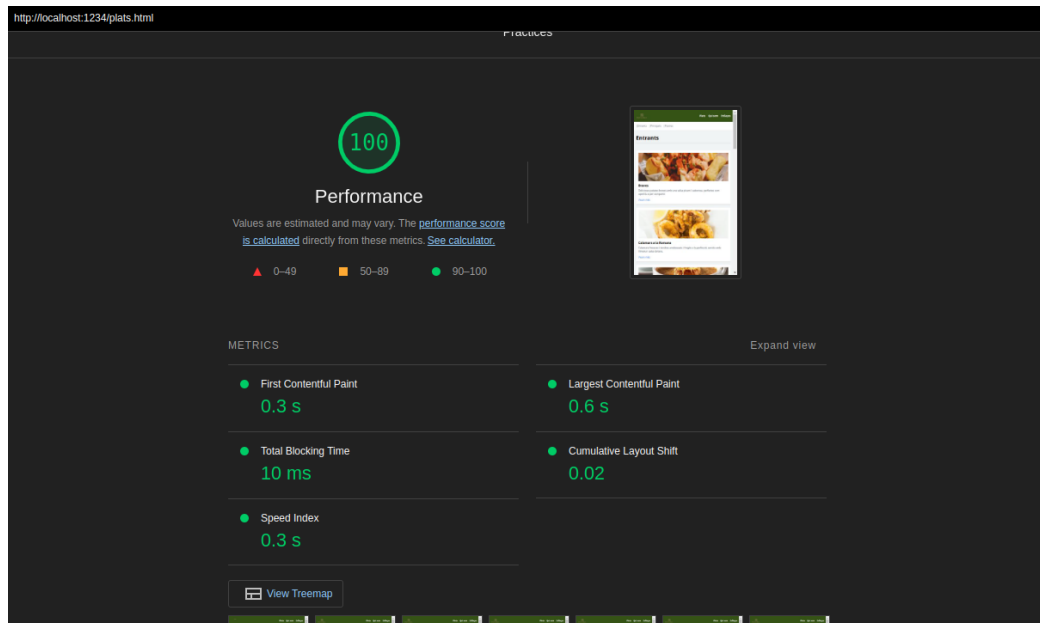


Passo totes les imatges dels plats png a webp utilitzant gimp (amb una compressió de 50 on baixa una mica la qualitat però no excessivament, excepte algunes imatges que si perdien significativament no hi he aplicat gaire compressió), aquí la taula:

Imatge	Original size(PNG)	WebP size (Saving %)
<i>Entrants</i>		
amanida.png	336,9 kB	22,6 kB (93,3%)
braves.png	327,8 kB	15,9 kB (95,1%)
calamars.png	226,1 kB	12,6 kB (94,4%)
sopa.png	220,2 kB	13,2 kB (94,0%)
<i>Principals</i>		
canelons.png	213,3 kB	14,9 kB (93,0%)
carbonara.png	319,4 kB	22,7 kB (92.9%)
paella.png	248,3 kB	54,3 kB (78.13%)
risotto.png	263,7 kB	16,3 kB (93,8%)
<i>Postres</i>		
cremacatalana.png	186,4 kB	25,4 kB (86,42%)
macedonia.png	163,6 kB	17,1 kB (89,55%)
pastis.png	195,5 kB	19,6 kB (89,95%)
tiramisu.png	232,3 kB	13,1 kB (94,36%)

Donat tots els percentatges dels savings, la mitjana és de 93,6% en totes les imatges de la web de plats.

Ara si faig un altre diagnosi amb el lighthouse, apareix una puntuació de 100 en performance:

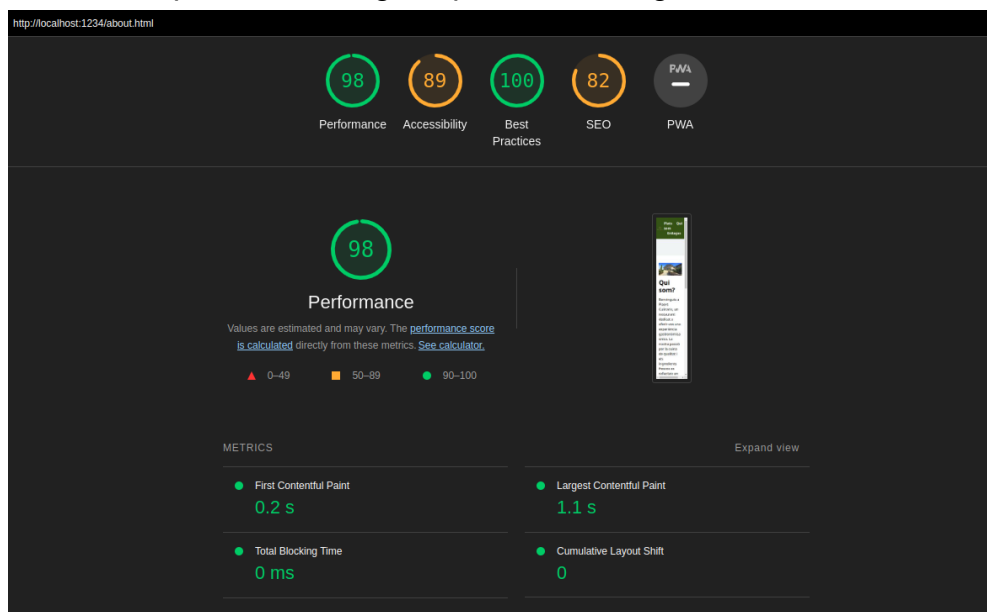


I ja no apareix l'apartat de oportunitats per millorar el rendiment de la web.

2. Pàgina About

2.1 Performance and images:

Faig un diagnosi de lighthouse i m'apareix una puntuació de 98 i em recomana utilitzar webp o avif i corregir el pes de la imatge:

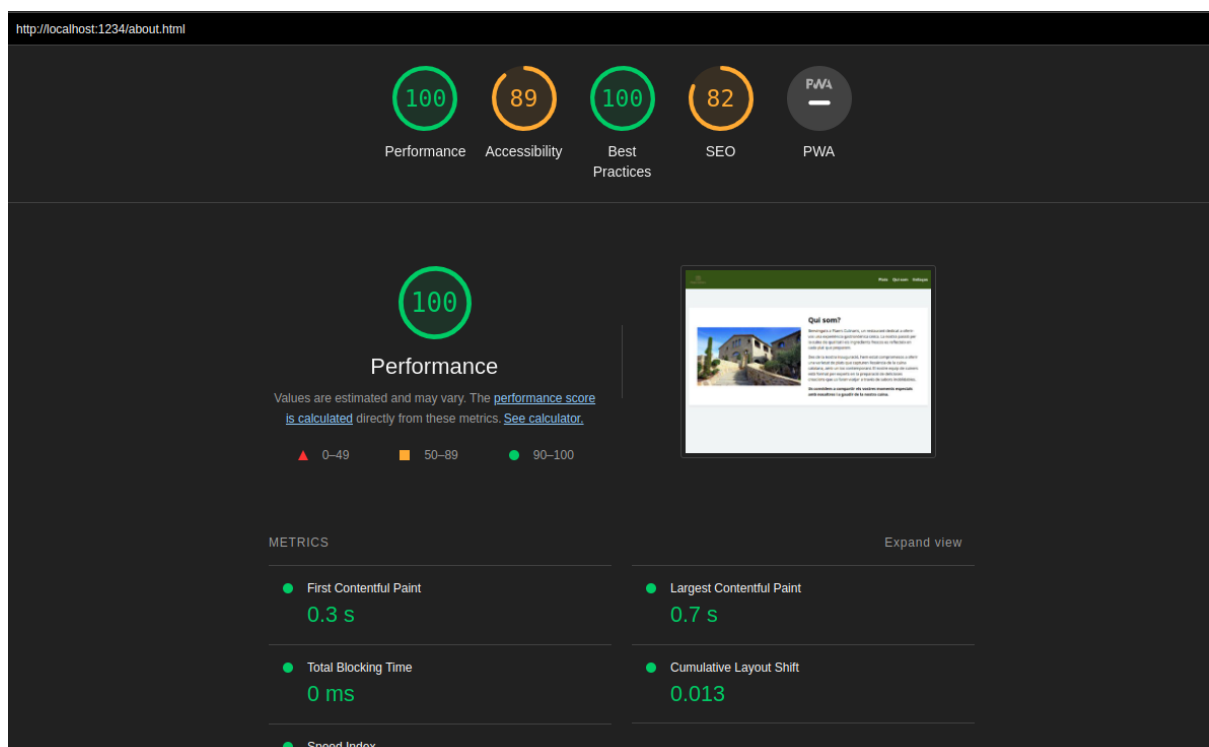


OPPORTUNITIES	
Opportunity	Estimated Savings
<div>■</div> Properly size images	<div></div> 0.68s
<div>■</div> Serve images in next-gen formats	<div></div> 0.32s

These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

Faig el canvi de la imatge jpg a webp i mostro els resultats:

Imatge	Original size(PNG)	WebP size (Saving %)
masia.jpg	884,3 kB	217,7 kB (75,34%)



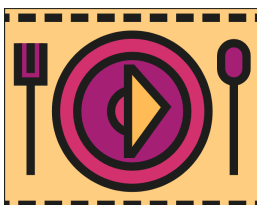
2.2 SVG:

Encara que provar de passar el meu logo (que vaig crear personalment amb un generador de logos online) a svg, amb una web convertora com:

<https://png2svg.com/>

Però he après que no serveix passar de png a svg ja que augmenta el pes de la imatge i la imatge queda deformada.

Així que el que faré per complir amb l'apartat de la rúbrica de svg serà utilitzar un logo svg d'internet que sí que pugui manipular correctament.



La classe de css que fa l'animació l'he definit així:

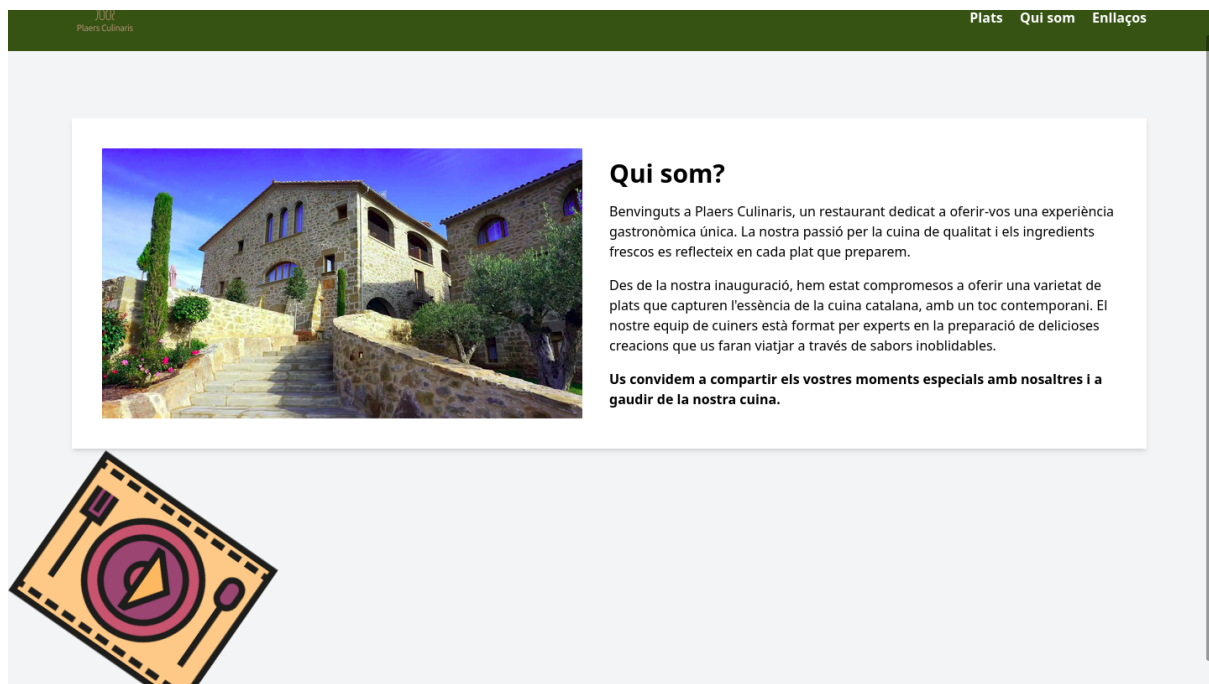
```
/* animacio que fa que l'svg doni voltes */
@keyframes rotate {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}

.animated-svg {
  animation: rotate 5s linear infinite;
  width: 250px;
  height: 250px;
}
```

Així al html:

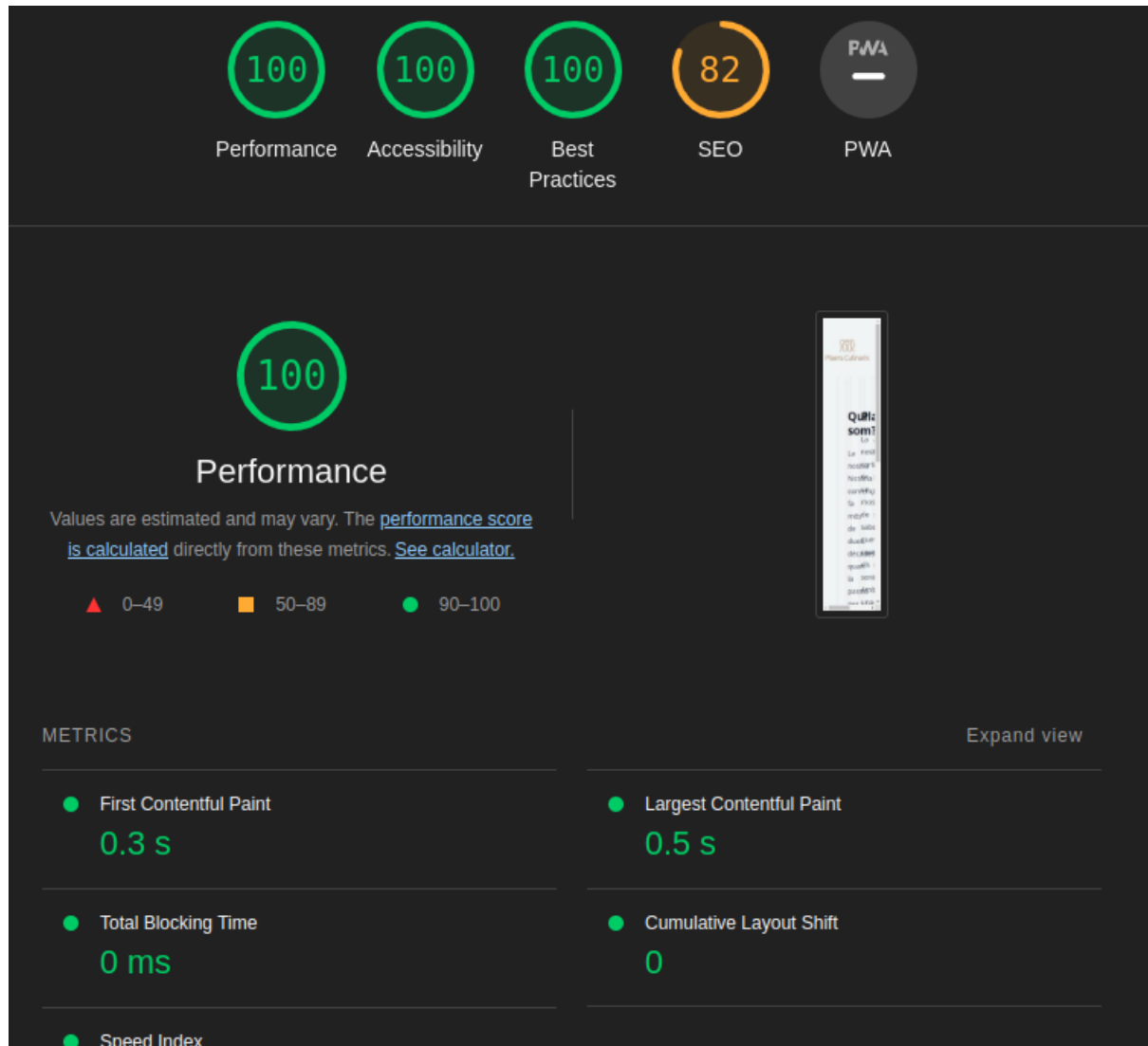
```
</div>
</div>
<div class="text-center mt-8 ml-10">
  
```

Ho he ficat en un div apart de el de la presentació perquè no trenques la estètica del about.



3.Pàgina Principal/Home

Sense fer cap canvi el lighthouse em mostra una puntuació de 100.



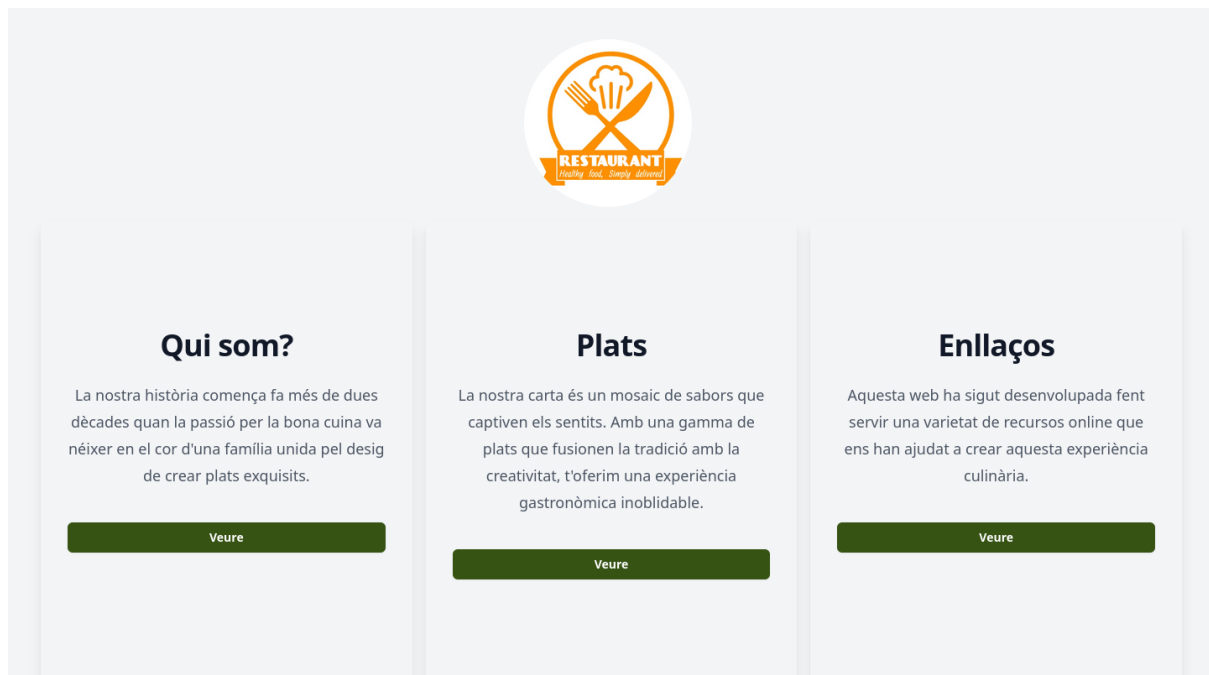
3.1 Clip-Path:

El que faré a aquesta pàgina és utilitzar clip-path a la homepage. Com el logo que tinc no té fons i és un png transparent afegiré un logo de internet per mostrar com funciona clip-path.

Aquest és l'estil de css:

```
.logo-cercle {  
  clip-path: circle(50% at 50% 50%);  
  border-radius: 50%;  
  width: 200px;  
  height: 200px;  
}
```


I així es com queda:



També he fet uns canvis per a què la pàgina sigui responsive, ja que era la única pàgina de tota la web que no ho era.

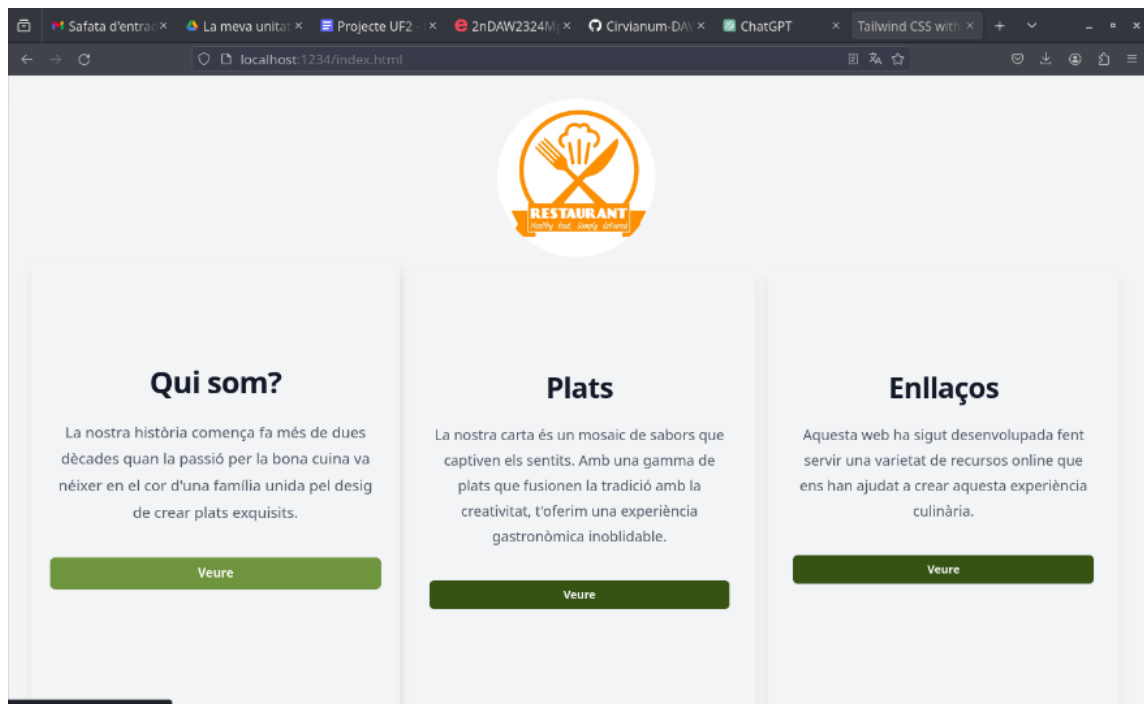
3.2 Css Animation:

Per a l'animació requerida a la homepage, el que he fet ha set afegirli animacions de hover als 3 divs principals i als botons alhora de passar el ratolí per sobre.

Així el css:

```
.hover-scale {  
  transition: transform 0.3s ease;  
}  
  
.hover-scale:hover {  
  transform: scale(1.05);  
}  
  
.hover-opacity {  
  transition: opacity 0.3s ease;  
}  
  
.hover-opacity:hover {  
  opacity: 0.8;  
}  
  
.hover-scale {  
  transition: transform 0.3s ease;  
}  
  
.hover-scale:hover {  
  transform: scale(1.05);  
}
```

Així queda la web al passar el ratolí per sobre de un div i botó:



5.Footer

El footer l'he fet senzill de aquesta manera a totes les webs:

```
<footer
| class="bg-lime-900 text-white py-4 text-center fixed bottom-0 w-full"
>
| <p>
|   &copy; 2023 Plaers Culinaris - Aaron Luna. Tots els drets reservats.
| </p>
</footer>
```

Recursos

- Contingut
 - Textos de plats i detall fet per intel·ligència artificial:
 - chat.openai.com

© 2023 Plaers Culinaris - Aaron Luna. Tots els drets reservats.

6.Imagemin i Parcel

Primer de tot instal·lo imagemin i els seus plugins al projecte:

```
npm install imagemin imagemin-mozjpeg imagemin-pngquant imagemin-gifsicle imagemin-webp --save-dev
```

Modifico el package.json perquè es pugui executar imagemin:

```
"scripts": {  
  "dev": "parcel src/index.html",  
  "build": "parcel build src/index.html && npm run imagemin",  
  "imagemin": "imagemin src/img/**/* --out-dir=dist/img/imagemin"  
},
```

Creo un js anomenat imagemin.config:

```
JS imagemin.config.js > ...  
1  module.exports = {  
2    plugins: [  
3      require("imagemin-mozjpeg")(),  
4      require("imagemin-pngquant")({ quality: [0.6, 0.8] }),  
5      require("imagemin-gifsicle")(),  
6      require("imagemin-webp")({ quality: 75 }),  
7    ],  
8  };  
9
```

El que fa és aplicar-li compressió als diferents tipus d'imatge.

I executo:

```
aaron@aaron:~/github-classroom/Cirvianum-DAW/mp9-pj-uf2-optimitzacio-imatges-AaronLunaa$ npm run imagemin  
  
> tailwind-parcel-project@1.0.0 imagemin  
> imagemin src/img/**/* --out-dir=dist/img/imagemin  
  
12 images minified
```

El que farà això és que després de construir l'aplicació amb Parcel, la tasca imagemin s'executarà per processar les imatges de la carpeta src/img i guardar-les a la carpeta de sortida dist/img/imagemin.

Aquí estan després de processar totes les imatges que tenia (totes ja eren .webp perquè les havia transformat manualment):

