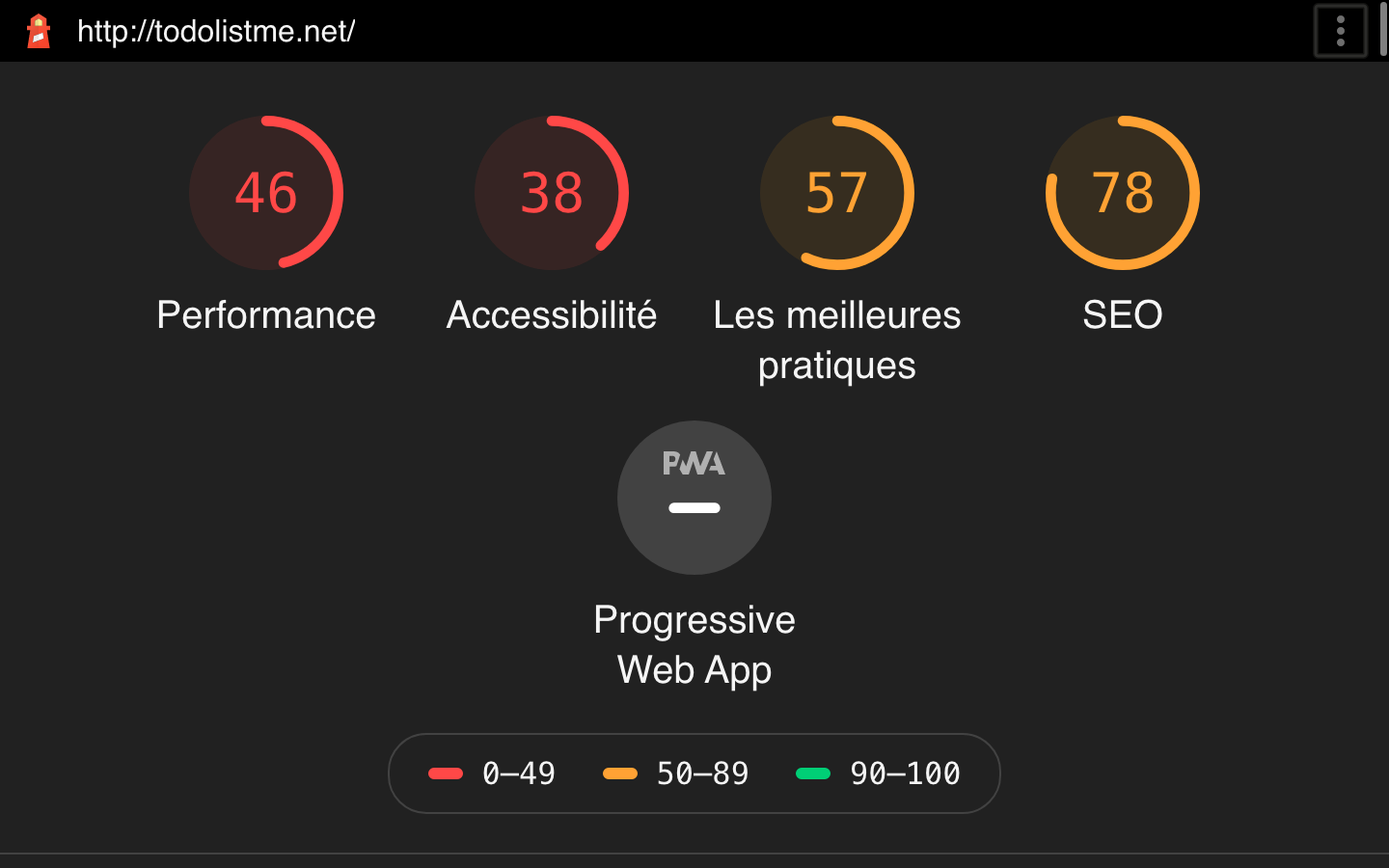
**Audit de performance du site web**

Un audit de performance permet d’évaluer la qualité d’un site web, vérifier qu’il répond aux bonnes pratiques et qu’il soit adapté aux internautes.

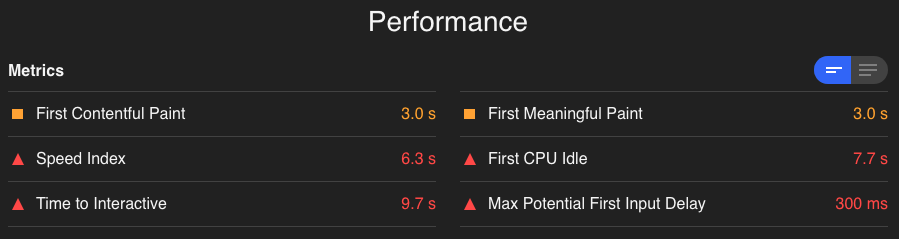
Il est important d’identifier les éléments à modifier ou à ajouter afin d’améliorer la performance. L’audit de performance est le moyen d’y parvenir.



L’analyse indique les résultats sur une échelle de 100 points :

* **Performance 46**
* **Accessibilité 38**
* **Les meilleures pratiques 57**
* **SEO 78**

**La performance**

****

**L’analyse des résultats donne 46/100. C’est en dessous de la moyenne. L’application est trop lente**

**Premier contenu affiché :**  
Le navigateur met 3.0 **secondes** pour afficher le rendu du premier élément de contenu du DOM.  
  
**Premier élément significatif affiché :**  
  
.Selon l'audit, il faut **3**.**0 secondes** avant que le contenu principal s’affiche.

**Indice de vitesse :**

Montre à quelle vitesse le contenu de la page est visiblement rempli.

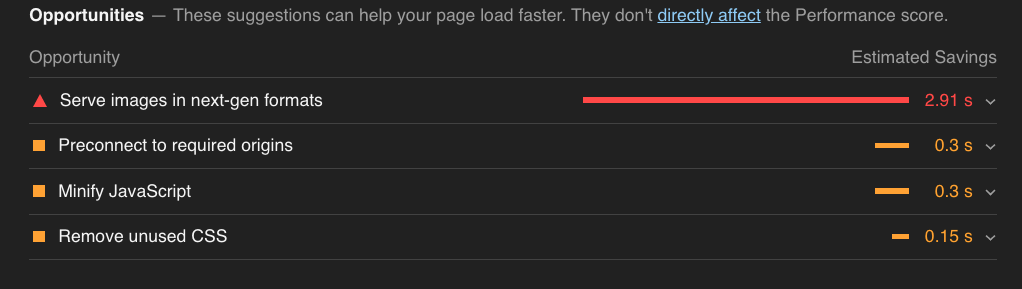
Cela prend environ **6.3** **secondes**.

**Temps avant interactivité :**  
Le navigateur met **9.7 secondes** pour que la page soit pleinement interactive.

**Délai d’entrée maximal :**  
L’application met **300 millisecondes** pour répondre à une entrée de l’utilisateur.

Au-delà de 50 millisecondes, l’utilisateur peut percevoir des lags (interactions en retard), ici il sera donc évident que ce sera le cas.

L’outil **DevTools audit** donne aussi des axes et des recommandations pour améliorer les performances du site :

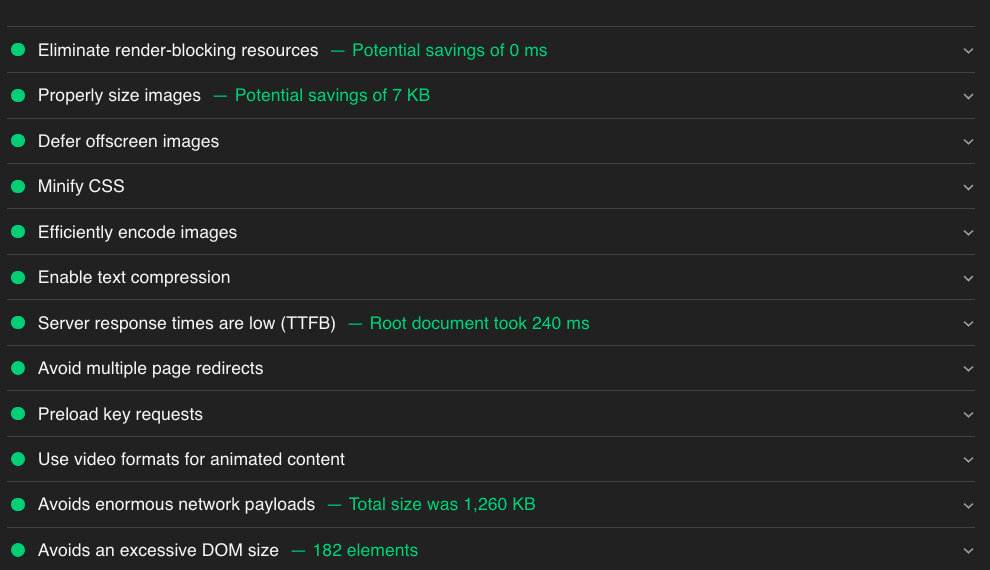


• **2.91** secondes seraient gagnées si les formats d'image étaient plus adaptés. Format d'image : **JPEG 2000, JPEG XR et WebP** offrent une meilleure compression que PNG ou JPEG.

• **0.3** secondes seraient gagnées si les fichiers JavaScript étaient **minifiés**. En effet, la réduction des fichiers JavaScript peut réduire la taille des charges utiles et le temps d'analyse des scripts.

• Supprimer le CSS inutilisé pour réduire les octets inutiles pourrait faire gagner 0.15 secondes.

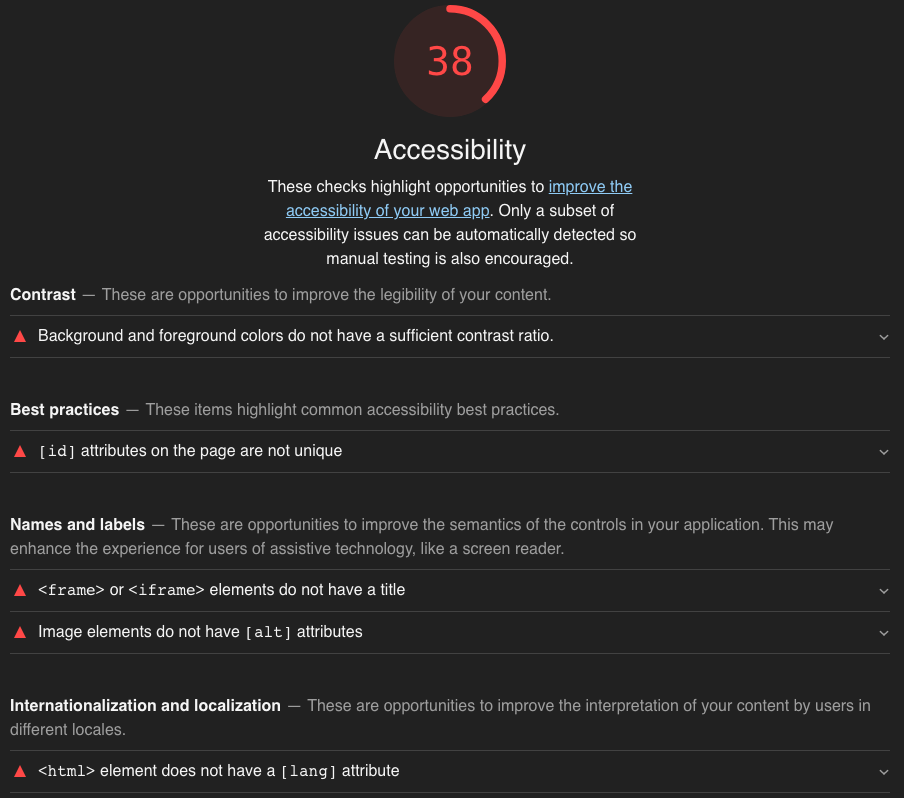
Il nous indique également les points positifs de l’application :



* Les images sont dimensionnées correctement.
* Le chargement des images non visibles est asynchrone.
* Le CSS est minifié. Le texte est compressé.
* Le serveur renvoie une réponse dans un temps correcte.
* Peu de redirections entraînant des retards avant le chargement de la page

**L’accessibilité**

L’outil DevTool indique également des axes d’amélioration concernant l’accessibilité :



**Contraste :** Les couleurs d'arrière-plan et de premier plan n'ont pas un rapport de contraste suffisant.

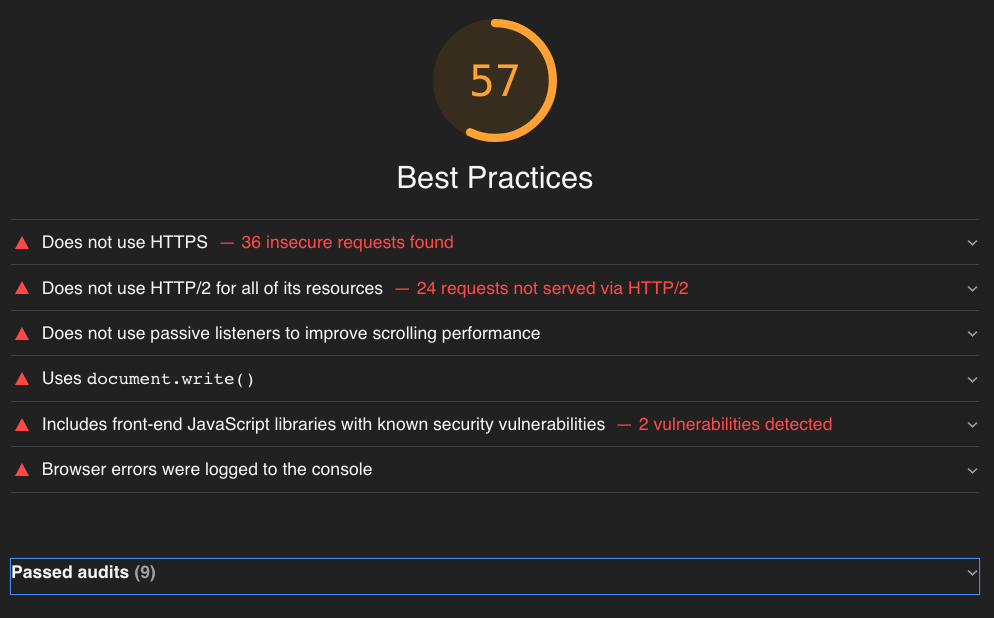
Un texte à **faible contraste** est difficile, voire impossible à lire pour de nombreux utilisateurs.

**Meilleures pratiques :** La valeur d'un attribut **id** doit être unique pour éviter que d'autres instances ne soient ignorées par les technologies d'assistance.

**Names and labels :** Les balises d'image n'ont pas d'attributs [alt] nécessaire au non voyant.

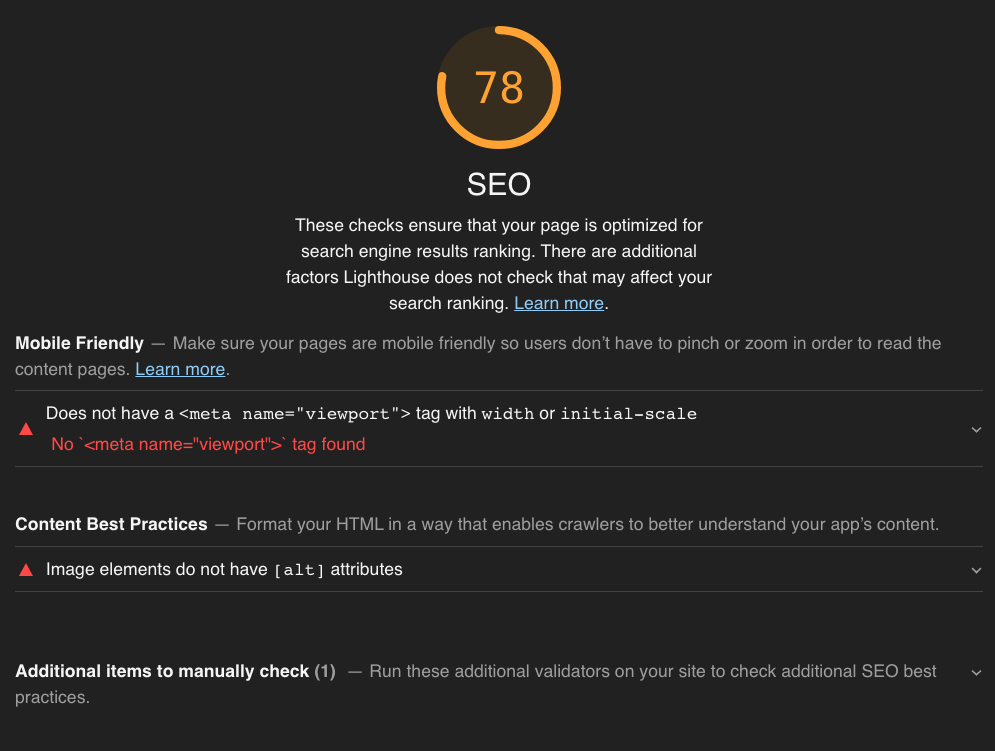
**Internationalisation et localisation :** la langue du site web n’est pas stipuler.

**Les meilleures pratiques**



* HTTPS : Tous les liens doivent être protégés par HTTPS, même ceux qui ne gèrent pas les données sensibles. HTTPS empêche les intrus de falsifier ou d'écouter passivement les communications entre votre application et vos utilisateurs, et est une condition préalable pour HTTP / 2 et de nombreuses nouvelles API de plate-forme Web.
* Une erreur dans la console.

**SEO Optimisation pour les moteurs de recherche**



* L’application ne dispose pas de balise <meta name="viewport"> pour optimiser votre application pour les écrans mobiles.

**Conclusion**

Les outils devTools nous montre que les performances du site **http://todolistme.net/** sont très en deca de la moyenne. L’application est lente est pas du tout optimisée sur différents points.

En vue de ces faiblesses. Nous pourrons améliorer notre application todolist en en vue d'un éventuel "scaling" :

**1. Améliorer le UX design et ajouter des fonctionnalités comme la catégorie de liste et la temporalité**

**2. En compressant les images pour réduire leur poids,**

**3. En minifiant les fichiers JavaScript grâce à Uglify, Minify ou Packer par exemple,**

**4. En nettoyant le CSS inutile présent dans le code.**

**5. Nous pourrions améliorer les contrastes et permettre un tri taches.**

**6. Travailler sur le référencement naturel**