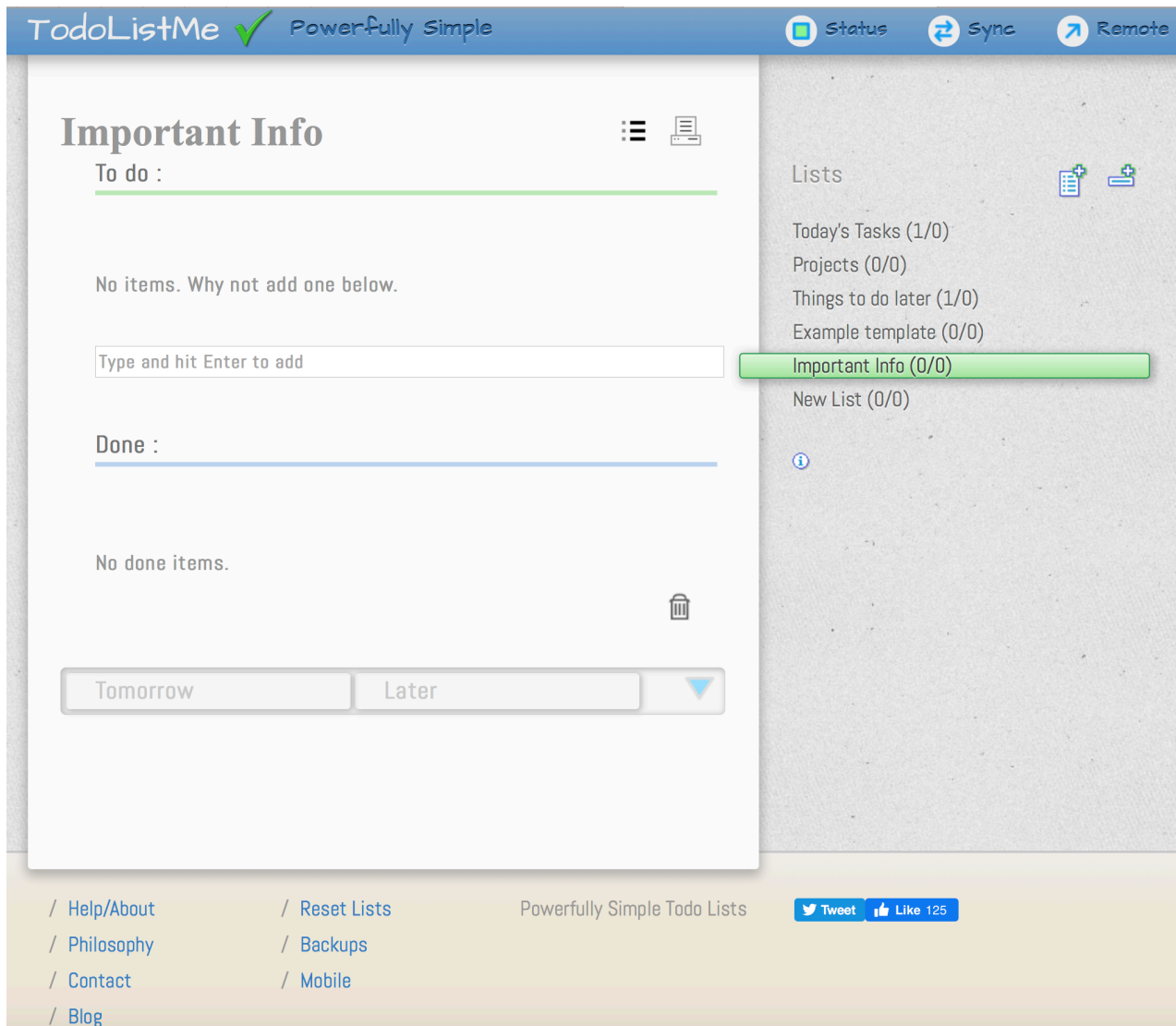


APPLICATION TODO LIST



Audit de Performance

Préparé pour : Projet 8 - OpenClassRooms

Préparé par : Audrey FRANÇOIS, Formation Développeuse d'Application Front-End

23 janvier 2020

SOMMAIRE

Contexte

Outils utilisés

Audit

- Généralités
- Performances
- Accessibilité et SEO
- Bonnes pratiques

Analyse SWOT

Conclusion

CONTEXTE

L'application analysée ici est TodoListMe (<http://todolistme.net>).

L'objectif de cette dernière rejoint la notre : permettre aux utilisateurs d'organiser au mieux leurs tâches quotidiennes.

Elle propose de :

- Créer des listes de tâches à effectuer, les enregistrer, les déclarer complètes, les éditer et les effacer.
- Créer des catégories de liste de tâches à effectuer.
- Donner une temporalité aux missions listées.

La comparaison avec notre application en l'état s'arrête à la 1ère ligne des propositions énumérées ci-dessus. En effet, les fonctionnalités de notre application consistent en une liste unique de tâches à réaliser. Aussi, l'audit portant sur les performances de l'application concurrente, nous permettra de soulever les écueils de cette dernière à éviter, de formuler des préconisations applicables à notre application en vu d'un éventuel « *scaling* » de Todos.

OUTILS UTILISES

Nous nous sommes basés sur les outils suivants :

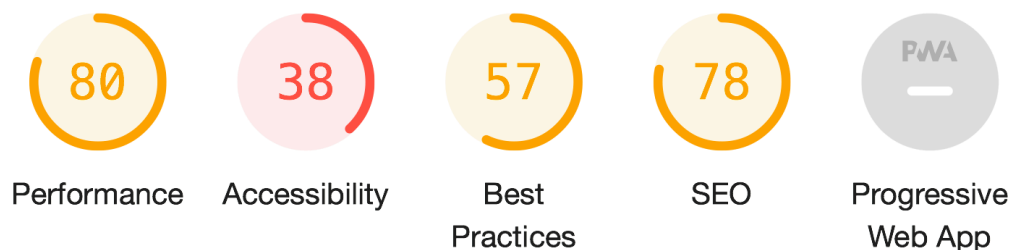
- DevTools de Chrome
- GTMetrix (<https://gtmetrix.com>)
- Dareboost (<https://www.dareboost.com/fr/dashboard>)

Les tests ont été menés sur différents device, différents CPU et différentes connexions.

AUDIT



<http://todolistme.net/>



APPLICATION TODO LIST

Généralités

Le 1er constat, **le temps de chargement** :

- Concurrent : 5,9 secondes. (5 secondes sont préconisées par le Rail Model performance)
- Notre application : environ 0.6 secondes (Acceptable).

La page concurrente n'est pas interactive avant plusieurs secondes. L'ordre d'affichage des éléments est peu pertinent.

Time to Interactive

5.9 s

Max Potential First Input Delay


















110 ms

View Trace

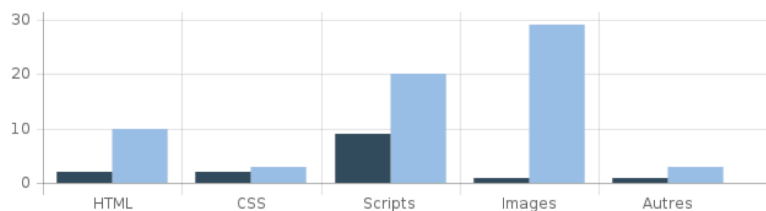
Values are estimated and may vary. The performance score is based only on these metrics.



Le nombre de requêtes : Concurrent : **92** (important) / Notre application : **14**.

500 ms		1000 ms	
			
Name		Status	
 texture.png	200		
 todolistme.net	200		
 3fh1mzuNMTm.js?_nc_x=Ij3Wp8lg...	200		
 top_not_saved.png	200		
 jquery-ui.js	200		
 top_sync.png	200		
 lists.js	200		
 lib.js	200		
 javascript_e.js	200		
 purge.png	200		
 favicon.ico	200		
 arrow_down.png	200		
 print.png	200		
 sort_order.png	200		
 ads?client=ca-pub-997142094504...	200		
92 requests 1.1 MB transferred 3.0 MB resource			

Nombre de requêtes par type de ressource



Poids par type de ressource


















Légende : ■ Notre application / ■ Application concurrente

D'après les diagrammes comparatifs, les scripts et les images de l'application concurrente sont au cœur des problématiques de chargement.

APPLICATION TODO LIST

Performance

Name	Status	Type	I...	Size	Time ▼
 todolistme.net	200	document	O..	4.6 KB	694 ms
 texture.png	200	png	(...	129 KB	501 ms
 ads?client=ca-pub-9971420945046...	200	document	s...	25.3 KB	451 ms
 jquery-ui.js	200	script	(...	122 KB	432 ms
 settings	200	fetch	w..	336 B	386 ms
 3fh1mzuNMTm.js?_nc_x=lj3Wp8lg5Kz	200	xhr	l...	127 KB	353 ms
 javascript_e.js	200	script	(...	10.1 KB	353 ms
 info.png	200	png	(...	529 B	321 ms
 top_new_window.png	200	png	(...	887 B	317 ms
 lists.js	200	script	(...	9.9 KB	316 ms
 like.php?app_id=&channel=https%3...	200	document	a...	16.9 KB	315 ms
 arrow_down.png	200	png	(...	945 B	314 ms
 top_not_saved.png	200	png	(...	814 B	309 ms
 lib.js	200	script	(...	1.7 KB	306 ms
 favicon.ico	200	x-icon	O..	17.5 KB	275 ms

Nom / fichier / document / type	PROBLEMATIQUES de l'application concurrente	PRECONISATIONS pour notre application
Fichier HTML	<ul style="list-style-type: none">- Structure peu organisée.- Des balises <scripts> sont disséminées un peu partout.	<ul style="list-style-type: none">- Veiller à ce que notre code HTML respecte les normes W3C.- Insérer les balises < script > à la fin du document.
Image PNG + « Texture.png »	<ul style="list-style-type: none">- Poids et temps de chargement,- Image : background.- Plusieurs requêtes consacrées au chargement d'images de type icônes au format PNG.	<ul style="list-style-type: none">- Background : utiliser un aplat de couleur via des propriétés CSS- Réduire le poids et la taille des images et utiliser des images au format de type webP

APPLICATION TODO LIST

Nom / fichier / document / type	PROBLEMATIQUES de l'application concurrente	PRECONISATIONS pour notre application
Publicités / services tiers / iframes (Google/DoubleClick Ads, jQuery CDN, Other Google APIs/SDKs, Google Analytics, Facebook, Twitter)	<ul style="list-style-type: none"> - Au chargement 66 requêtes appelées concernent les services tiers avant que les fonctionnalités principales ne soient chargées. 	<ul style="list-style-type: none"> - Ajouter un attribut « async » ou « defer » dans les balises < script > concernées - Ajouter des indicateurs de ressources preconnect ou dns-prefetch. - Iframes : boutons réseaux sociaux, préférer un lien cliquable et un script asynchrone.
Jquery-ui.js Fichiers JS	<ul style="list-style-type: none"> - 86,6% du code jQuery non utilisé - Certains scripts pas minifiés. 	<ul style="list-style-type: none"> - Analyser si l'utilisation de jQuery-ui.js est justifiée. - Minifier les fichiers .js
Ressources statiques	<ul style="list-style-type: none"> - 35 pas mises en cache. 	<ul style="list-style-type: none"> - Mettre en cache certaines ressources (optimisation pour les visites ultérieurs de l'utilisateur)
Appel API Google FastButton	<ul style="list-style-type: none"> - Erreur 404. 	<ul style="list-style-type: none"> - Réparer le lien brisé, - Ou supprimer les références à la ressource inexistante.
Polices Web / fonts.googleapis.com/css?family=Abel Architects+Daughter	<ul style="list-style-type: none"> - Attention aux dépendances critique d'une page web à un contenu externe. S'assurer que le texte reste lisible pendant le chargement de la police 	<ul style="list-style-type: none"> - Éliminez ces dépendances autant que possible - Ou s'assurer que l'appel est asynchrone et qu'il y a des alternatives en cas d'échec - Utilisation de font-display
Fichiers CSS	<ul style="list-style-type: none"> - pas minifiés - mot clé « !important » utilisé 17 fois - 1 sélecteur CSS trop complexe - 1 propriété CSS surchargée 	<ul style="list-style-type: none"> - Minifier fichier .css - Revoir l'organisation des fichiers .css - Définir une fois la même propriété dans une règle. Modification que nous avons apportée dans notre application.

APPLICATION TODO LIST

Accessibilité et SEO

Le rapport de contraste entre le fond et le 1er plan n'est pas suffisant. Même constat pour notre application.

=> S'assurer que les contrastes soient suffisants.

Un ID n'est pas unique (concurrent).

=> Modifiez une valeur d'ID si elle est utilisée plusieurs fois.

=> Dans notre application, ajout d'un ID à input.new-todo lié au label correspondant.

Les éléments `<iframe>` sans titre, les images pas d'attribut « alt », certains formulaires n'ont pas de labels et la balise `<html>` n'a pas d'attribut « lang ».

=> Revoir la sémantique du code.

Vigilance quant aux bonnes pratiques d'accessibilités. Voici quelques pistes d'améliorations :

=> Faire en sorte que la navigation soit possible avec le clavier.

=> Vérifier que les éléments interactifs indiquent leur fonction et leur état et se distinguent bien des éléments non-interactifs.

=> L'ordre du DOM doit correspondre à l'ordre visuel.

Absence de `<meta name="viewport">` tag pour optimiser l'affichage sur les mobiles.

=> Rendre notre application responsive.

=> Dans notre application, il manquait une méta-description.

Bonnes Pratiques

Notre concurrent n'utilise pas le protocole HTTPS.

=> Tous les sites Web doivent être protégés par HTTPS

Il n'utilise pas HTTP / 2 pour toutes ses ressources

=> HTTP/2 charge les ressources de la page rapidement, avec moins de données sur le réseau /1

La bibliothèque jQuery@2.2.4 présente des failles de sécurité (site vulnérable).

=> Arrêtez d'utiliser la bibliothèque signalée. Privilégier un framework Javascript, tel que React.js, (approche par composants et meilleure structuration du code).

Des erreurs de navigation s'affichent sur la console.

=> Nous avons résolu le problème d'affichage du favicon dans notre Application.

APPLICATION TODO LIST

ANALYSE SWOT

	ATOUTS	HANDICAPES
Notre Application TODOS	Forces : <ul style="list-style-type: none">- Rapidité de chargement- Fonctionnelle- Simple- Code organisé MVC	Faiblesses : <ul style="list-style-type: none">- Liste unique de tâches à réaliser- Bugs à corriger dans le code- Code à optimiser- Accessibilité à optimiser- Non autofinancé
Application Concurrente ToDoListMe	Opportunités : <ul style="list-style-type: none">- Plus de fonctionnalités pratiques- Financement (publicité)	Menaces : <ul style="list-style-type: none">- Maîtrise du temps de chargement- Sécurité

CONCLUSION

ToDoListMe est plus complète que Todos en terme de fonctionnalités cependant il s'agit là d'une application plutôt lente au chargement, non optimisée en terme d'accessibilité, SEO et au design discutable. Certaines bonnes pratiques permettraient d'améliorer quelque peu ses performances.

En relevant les écueils de ToDoListMe, nous constatons que notre application, après correction de quelques bugs, de par sa performance sort son épingle du jeu. Elle est beaucoup plus rapide et facile à utiliser. Nous pourrions rendre notre outil encore plus attractif et performant en veillant aux points suivants :

- Ajouter 3 fonctionnalités :
 - Créer des catégories de listes,
 - Programmer les tâches dans un calendrier.
 - Possibilité de créer un compte utilisateur avec option Premium (éventuel financement).
 - Produire un code minifié et épuré,
 - Mettre en cache certaines ressources,
 - Vérifier qu'il n'y ait pas de message d'erreur dans la console de débogage (ajout d'un favicon).
 - Veiller à ce que les connexions aux services tiers soient pleinement justifiées et si c'est le cas faire en sorte qu'elles impactent le moins possible le chargement de notre application,
 - Veiller à ce qu'elle soit Responsive.
 - Orientée Progressive Web App.
 - Utiliser le protocole HTTPS et configurer notre serveur en HTTP/2,
 - Produire un code HTML propre, respectant les standards pour un meilleur référencement,
 - Améliorer le contraste fond/1er plan.
-