

TASKSTEP

RAPPORT D'OPTIMISATION DE L'APPLICATION –
SAE4

Table des matières

I.	Introduction	3
II.	Rappel de la synthèse de l'audit	4
III.	Planning détaillé.....	6
IV.	Synthèse des résultats après optimisation	9
V.	Perspectives	12
VI.	Annexes.....	13

I. Introduction

A. Cadre de la mission

Dans le cadre d'une SAÉ (Situation d'Apprentissage et d'Évaluation) de 2^{ème} année de BUT Informatique, nous devions effectuer l'audit d'une application afin de la juger dans les domaines de la sécurité, de la qualité logicielle, de l'accessibilité, de l'ergonomie, des performances et de la gestion de données.

Nous avons pu auditer l'application Taskstep, une application gratuite et open source de gestion de tâches en ligne permettant de catégoriser ces dernières en sections, projets, contextes, ...

Ce document fait suite à cet audit et présentera les nombreuses optimisations ayant pu être réalisées par le groupe de 1PeuDinspi, allant d'une refonte architecturale à la mise à jour des technologies pour rendre l'application beaucoup plus ergonomique mais aussi ouverte à de futures éventuelles extensions.

B. Présentation de l'application auditée

L'application **TaskStep** est un gestionnaire de listes de tâches développé pour un usage local. Elle a été initialement conçue par [Rob Lowcock](#) en **2006**, puis refactorisée par [Ethan Romba](#) en **2008**. Enfin elle a été mise à jour par [Thomas Hooge](#) en **2020**. Elle fonctionne dans un environnement de développement basé sur **PHP (version 7.0 ou supérieure)** et utilise une base de données **MariaDB (ou MySQL)**. L'application est conçue pour être installée localement sur un serveur tel qu'**Apache**, accessible via une adresse comme `http://localhost` ou `127.0.0.1`. Elle est pensée pour un usage **multi-utilisateur**. TaskStep permet de gérer des tâches en les classant par échéance : immédiates, à faire cette semaine, ce mois-ci, cette année ou à long terme. Elle offre aussi des filtres par **contextes** et **projets**, un affichage automatique des tâches du jour, une mise en évidence des tâches en cours ou en retard, ainsi que la possibilité d'imprimer des listes au format **3x5**. L'application est **open-source** et disponible en plusieurs langues, dont l'anglais, le russe, l'allemand et partiellement l'espagnol.

C. Contraintes

Ce projet d'optimisation a pu être fait par 5 personnes, en 36 heures par personne (donc un temps cumulé de 180 heures au total). Nous avons utilisé XAMPP pour héberger le projet et non une plateforme d'hébergement de grande envergure (Docker, etc.). Type d'application imposé en application web, le langage PHP est donc également à conserver pour plus de sécurité côté serveur.

D. Ressources

Lien vers l'application originale : <https://github.com/eromba/taskstep>

Lien vers le GitHub de l'application optimisée par 1PeuDinspi : https://github.com/dept-info-iut-dijon/S4_C1_1PeuDinspi

II. Rappel de la synthèse de l'audit

L'audit de Taskstep auquel ce document fait suite a pu relever de nombreuses lacunes sur toutes les facettes évoquées plus tôt, qui seront également détaillées ci-dessous. Ces rapports ont pu être réalisés en 16 heures par personne, soit 80 heures de travail au total, et portaient sur un aspect différent de l'application chacun.

A. Audit d'ergonomie

Afin d'analyser l'ergonomie du site web, nous nous sommes basés sur les critères de Bastien et Scapin, et avons réalisés divers tests visuels et expérimentaux afin de déterminer quelles lacunes sont à corriger.

Nous en avons déduit que le site souffrait de nombreux défauts d'expérience utilisateurs tels que la responsivité du site, sa lisibilité (format 4:3 et site en zoom par défaut illisible) ou encore l'entrée de données gérée d'une façon non ergonomique. Très peu de points positifs existaient réellement au site web, si ce n'est l'homogénéité du site web.

B. Audit de sécurité

Pour ce qui est de la sécurité, le respect du Règlement Général sur la Protection des Données sera assuré ainsi que les bonnes pratiques de sécurité. Nous avons donc effectué différents tests tels que de brute force (à l'aide d'Hydra), d'intrusion sur le site, d'injection SQL, de XLSS ou encore de modifications de cookies.

Ainsi, nous en avons déduit la présence de très nombreuses lacunes qui rendent le site très fragile en termes de sécurité tels que la gestion des données par protocoles HTTPS, la sécurisation de requêtes SQL dans le programme du site ou encore des limites imposées sur le mot de passe.

C. Audit d'accessibilité

L'audit d'accessibilité a été réalisé dans le respect des normes WCAG et des lois européennes sur l'accessibilité. Des tests de contrastes WCAG, d'autres visuels ou encore des analyses du code source du projet.

Ceci a permis de déduire que le site souffrait d'un manque de balises sémantiques organisant la structure du site web, de contrastes et de couleurs mal gérées ou encore de très mauvaise gestion des liens du site web.

D. Audit de performances & Impact environnemental

L'audit de performances et d'impact environnemental a été réalisé dans le respect des principes d'éco-conception numérique et en s'appuyant sur des outils reconnus tels que Lighthouse et GreenIT Analysis. Les tests ont été menés sur plusieurs scénarios représentatifs des usages utilisateurs afin d'évaluer à la fois la rapidité d'exécution de l'application et son empreinte environnementale.

Cette analyse a permis de mettre en évidence des performances satisfaisantes en environnement léger, mais aussi des dégradations notables en contexte de forte charge. L'étude environnementale, quant à elle, a révélé plusieurs bonnes pratiques déjà respectées, tout en pointant des vulnérabilités structurelles liées au chargement des ressources, à l'absence de mise en cache, ou encore à l'architecture non adaptée à la montée en charge.

E. Audit de Gestion de données

Les normes du RGPD et les standards de gestion de données sont aux bases de l'analyse de la gestion de données du projet. Pour ce faire, une réflexion approfondie sur la gestion des données de l'application a pu être réalisée, relevant plusieurs défauts de l'application.

L'audit de gestion de données a permis de remarquer une non-conformité de cette base de données. En effet, plusieurs attributs peuvent avoir la même valeur ce qui ne correspond pas à l'utilisation de cette base de données car chaque titre de tâche doit être unique, pour cela il faut spécifier l'attribut comme unique. De plus, aucune association via des clés étrangères n'est présente, ce lien est fait avec un attribut de l'autre table et non une clé étrangère permettant une association conforme.

La table settings stocke des informations inutiles, l'application devra être multiutilisateur, pour cela l'ajout d'attribut login est nécessaire. Pour une meilleure lisibilité, la table sera renommée user.

F. Audit de qualité logicielle

La qualité du code a été jugée selon les principes SOLID, les bonnes pratiques de programmation et le PSR (PHP Standards Recommandations). Pour ce faire, des tests sous machine virtuelle vierge étaient effectuées afin de tester l'installation de l'application et le bon fonctionnement du manuel d'installation. Des analyses de qualité de code à l'aide de PHPStan et de Webimpress Coding Standards ont pu être réalisés. Et diverses lectures du code source permettant de déduire les points problématiques du projet (documentation interne, logique, répétitions, etc.)

Un sévère manque d'architecture se faisait présent durant toute l'analyse, impliquant donc une refonte architecturale du projet, ce qui nous permettra par la même occasion de respecter les principes SOLID qui se font inexistant sur tout le projet. L'usage de bases de données MySQL dans le projet se fait non seulement de façon non sécurisée mais aussi ne prenant pas en compte les éventuelles exceptions en plus de ne pas être testé.

G. Synthèse

L'application complète souffrait d'un immense manque d'architecture rendant son code illisible, en plus de ne pas être documenté. La principale décision qui en découlait était de redémarrer l'application de 0 et de modifier les différentes fonctionnalités de sorte qu'elles soient plus ergonomiques, accessibles et surtout sécurisées dans le cadre d'une application multi-utilisateurs.

Nous avons donc l'occasion de respecter tous les critères de jugement des audits précédemment évoqués à travers la refonte de ce projet, dont nous décrirons le déroulement lors de la partie suivante

III. Planning détaillé

Tableau de présentation de l'équipe de développement. Description de la méthode de travail suivie (cascade, agile ou autre...)

A. Équipe de projet

Nom Prénom	Rôle	Champs d'intervention
DAL MOLIN Tristan	Chef de projet	Rapport d'audit, Qualité logicielle (Format de code source, principes de qualité), Accessibilité Mise en place architecturale, système de routing
ALAOUI Moulay-Wassim	Membre	Rapport d'audit, Qualité logicielle (Documentation, principes de qualité), performances et environnement, accessibilité Mise en place architecturale, connexion et inscription, messages de succès
DIOURI Wassim	Membre	Rapport d'audit, Qualité logicielle (Principes de qualité), Ergonomie Mise en place architecturale, vues et factorisation de CSS
DE MARCO Matteo	Membre	Rapport d'audit, Qualité logicielle (Tests de l'application), Gestion des données Mise en place architecturale, mise en place base de données
DUTRION Jules	Membre	Rapport d'audit, Qualité logicielle (Pile technique), sécurité, ergonomie Mise en place architecturale, sécurité du projet

Tableau 1: Équipe de projet

B. Plan d'action prévisionnel du projet

Le plan d'action suivant était décidé dans l'audit global de l'application :

Projet	Action	Priorité	Responsable de l'action	Charge (H/J)	Planification
Projet 1: Refonte architecturale	Action 1.1 : Mise en place architecture (création des packages, classes de base, etc.)	5	Tristan DAL MOLIN	4 Heures	12/05
	Action 1.2: Création des classes abstraites (vue, DAO, etc.)	5	Moulay-Wassim ALAOUI	2 Heure	12/05
Projet 2 : Création de la couche Métier	Action 2.1 : Création des classes de métier (Project, Section, Context, User et Task)	5	Matteo DE MARCO	2 Heure	12/05
Projet 3 : Création de la couche DAO	Action 3.1 : Création de ProjectDAO	3	Wassim DIOURI	1 Heure	12/05
	Action 3.2 : Création de SectionDAO	3	Moulay-Wassim ALAOUI	1 Heure	12/05
	Action 3.3 : Création de ContextDAO	3	Matteo DE MARCO	2 Heure	12/05
	Action 3.4 : Création de UserDao	3	Jules DUTRION	2 Heure	12/05
	Action 3.5 : Création de TaskDAO	3	Tristan DAL MOLIN	2 Heure	12/05
Projet 4 : Création de la couche Service	Action 4.1 : Création de ProjectService	3	Wassim DIOURI	2 Heure	13/05
	Action 4.2 : Création de SectionService	3	Moulay-Wassim ALAOUI	2 Heure	13/05
	Action 4.3 : Création de ContextService	3	Matteo DE MARCO	2 Heure	13/05
	Action 4.4 : Création de UserService	3	Jules DUTRION	4 Heures	13/05
	Action 4.5 : Création de TaskService	3	Tristan DAL MOLIN	3 Heures	13/05
Projet 5 : Création de la couche Controller	Action 5.1 : Création de ControllerHome	4	Jules DUTRION	2 Heure	13/05
	Action 5.2 : Création de ControllerSettings	4	Tristan DAL MOLIN, Matteo DE MARCO	6 Heures	13/05
	Action 5.3 : Création de ControllerTask	4	Tristan DAL MOLIN	4 Heures	13/05
	Action 5.4 : Création de ControllerContext	4	Matteo DE MARCO	2 Heures	13/05
	Action 5.5 : Création de ControllerProject	4	Wassim DIOURI	2 Heures	13/05

	Action 5.6 : Création de ControllerSection	4	Moulay-Wassim ALAOUI	2 Heures	13/05
Projet 6 : Création de la couche Router	Action 6.1 : Création du système de Routing	5	Tristan DAL MOLIN, Jules DUTRION	6 Heures	14/05
	Action 6.2 : Création des routes de l'accueil	4	Jules DUTRION	4 Heures	14/05
	Action 6.3 : Création des routes des paramètres	4	Tristan DAL MOLIN, Matteo DE MARCO	4 Heures	14/05
	Action 6.4 : Création des routes des tâches	4	Tristan DAL MOLIN	4 Heures	14/05
	Action 6.5 : Création des routes de Projets	4	Wassim DIOURI	4 Heures	14/05
	Action 6.6 : Création des routes de Contextes	4	Matteo DE MARCO	4 Heures	14/05
	Action 6.7 : Création des routes de Sections	4	Moulay-Wassim ALAOUI	4 Heures	14/05
Projet 7 : Création de la couche Vue	Action 7.1 : Création des Header/Footer du projet	4	Wassim DIOURI	6 Heures	14/05
	Action 7.2 : Création de la page d'accueil	3	Tristan DAL MOLIN	4 Heures	14/05
	Action 7.3 : Création des pages de listes	3	Matteo DE MARCO	4 Heures	14/05
	Action 7.4 : Création de la page d'édition de tâches	2	Tristan DAL MOLIN	4 Heures	14/05
	Action 7.5 : Création de la page d'édition de projet	2	Moulay-Wassim ALAOUI	2 Heure	14/05
	Action 7.6 : Création de la page d'édition de contexte	2	Moulay-Wassim ALAOUI	2 Heure	14/05
	Action 7.7 : Création de la page d'édition de section	2	Moulay-Wassim ALAOUI	2 Heure	14/05
	Action 7.8 : Création de la page de paramètres	3	Tristan DAL MOLIN	2 Heures	14/05
	Action 7.9 : Création de la page de connexion	5	Jules DUTRION	2 Heure	14/05
	Action 7.10 : Création de la page d'aide	1	Moulay-Wassim ALAOUI	4 Heures	14/05
Projet 8 : Tests unitaires	Action 8.1 : Tests de la table Project	2	Wassim DIOURI	4 Heures	13/05
	Action 8.2 : Tests de la table User	2	Jules DUTRION	4 Heures	13/05
	Action 8.3 : Tests de la table Context	2	Matteo DE MARCO	4 Heures	14/05
	Action 8.4 : Tests de la table Section	2	Moulay-Wassim ALAOUI	4 Heures	14/05
	Action 8.5 : Tests de la table Task	2	Tristan DAL MOLIN	4 Heures	14/05
Projet 9 : Ergonomie / Accessibilité	Action 9.1 : Messages d'erreurs gérés en javascript	3	Matteo DE MARCO	2 Heure	15/05

	Action 9.2 : Responsivité du site	3	Tristan DAL MOLIN, Jules DUTRION	8 Heures	15/05
	Action 9.3 : Noms de liens et actions	2	Moulay-Wassim ALAOUI	2 Heures	15/05
	Action 9.4 : Alt d'images	2	Matteo DE MARCO	1 Heure	15/05
	Action 9.5 : Révision des visuels	2	Wassim DIOURI	8 Heures	16/05
	Action 9.6 : Gestion de la navigation	4	Tristan DAL MOLIN	2 Heures	16/05
	Action 9.7 : Arborescence des titres	3	Wassim DIOURI	1 Heures	16/05
	Action 9.8 Balises sémantiques	3	Moulay-Wassim ALAOUI	1 Heures	16/05
	Action 9.9 : Contraste	2	Jules DUTRION	4 Heures	16/05

Nous avons donc pu respecter le plan d'action prévisionnel précédent durant tout le projet et avons également eu beaucoup d'avance sur ce qui devait être fait au cours de la semaine. C'est une avance qui a été perdue sur la rédaction des audits et la préparation d'une soutenance orale du projet, nous permettant de le finir dans les temps attendus.

IV. Synthèse des résultats après optimisation

A. Introduction

À la suite de l'audit de l'application Taskstep, nous avons pu déduire qu'une refonte totale de l'architecture du projet était nécessaire. Ainsi, pour optimiser cette application, nous sommes repartis de 0 et avons réimplémenté les différentes fonctionnalités de l'application de sorte qu'elle soit beaucoup plus accessible et ergonomique, mais aussi qu'elle puisse aisément être ouverte à de futures extensions.

B. Audit d'ergonomie

L'audit d'ergonomie nous avait permis, à l'aide des critères de Bastien et Scapin, de déterminer que le site souffrait de nombreux défauts d'expérience tels que la responsivité du site, sa lisibilité (format 4:3 et site en zoom par défaut illisible) ou encore l'entrée de données gérée d'une façon non ergonomique.

Ainsi, nous avons pu mettre à jour l'affichage du site pour le rendre responsive et en format 16:9, adapté aux écrans de nos jours. La navigation a été rendue beaucoup plus claire à l'aide de titres de page et de boutons plus clairs sur là où ils redirigent. Les différents messages d'erreurs étaient également plus précis après correction, nous indiquant beaucoup plus clairement ce qu'il faut réparer en cas d'erreur.

C. Audit de sécurité

L'audit de sécurité nous permettait de déduire que le site ne respectait pas le Règlement Général sur la Protection de Données, laissant le site ouvert à d'éventuelles injections SQL, des intrusions (mode sans mot de passe) ou encore des brut force.

Finalement, nous avons donc pu optimiser la quasi-totalité des failles de sécurité du site web y compris toutes celles évoquées plus tôt. Beaucoup peuvent nécessiter de bien meilleures fonctionnalités (limite par session pouvant être évitée par suppression du cookie de session par exemple) mais pour un site web ne servant qu'à lister des tâches de façon locale ou interne à une entreprise, nous estimons avoir apporté des ajustements suffisants à la sécurité du site web.

D. Audit d'accessibilité

L'audit d'accessibilité mettait en valeur différents défauts du site web tels que le contraste, la mauvaise utilisation de balises sémantiques ou encore des gestions de liens très défectueux.

Les recommandations prioritaires qu'étaient l'utilisation des bonnes balises sémantiques, l'amélioration du contraste entre texte et arrière-plan, l agrandissement de la taille du texte qui était illisible en zoom 100%, ... Quelques points secondaires restent à traiter, mais l'application répond désormais largement aux standards d'accessibilité numérique.

E. Audit de performances & Impact Environnemental

Cet audit s'est appuyé sur des outils tels que GreenIT Analysis (pour analyser l'éco-conception du site) et Lighthouse (pour les performances), afin d'analyser 2 fonctionnalités majeures : l'ajout d'une tâche et la visualisation des tâches, testées en environnement léger (peu de données insérées) et lourd (plus de 3000 tâches).

L'ajout majeur ayant permis d'optimiser les performances du site aura été d'ajouter la pagination à la liste des tâches, se faisant directement en requête SQL avec « offset » et « limit », de sorte qu'on ne retourne que 10 tâches par requête au lieu de 3000 en cas d'environnement lourd par exemple. Ce simple changement a permis à l'application de passer d'un score Lighthouse de 59/100 à 100/100, et d'un score EcolIndex E à A.

Le passage à une architecture MVC fait légèrement obstacle aux performances, faisant circuler des données à travers différentes couches (Controller -> Service -> DAO). Mais la perte de performances que cela cause est minime face aux nombreuses optimisations qu'une telle architecture permet de façon aussi modulaire.

F. Audit de Gestion de données

La base de données de l'application souffrait principalement d'un manque de clarification sur certains attributs, de manque de contraintes sur d'autres (titre de tâche devant être non nul par exemple) et de clefs étrangères sur tout le modèle conceptuel de données.

Nous avons donc pu apporter des clefs étrangères, les attributs et tables inutiles ont pu être retirée du projet et différents index sur des clefs étrangères ont pu être créés pour améliorer les performances. À travers ce nouveau MCD, nous avons donc su améliorer l'usage de notre base de données.

G. Audit de qualité logicielle

Notre analyse de qualité logicielle du projet était jugée selon les principes SOLID, les bonnes pratiques de programmation et le PSR (PHP Standards Recommandations). Nous avons pu remarquer le manque d'architecture réelle du projet étant donné qu'il est fait sans Programmation Orientée Objet, impliquant donc aussi un manque de tests unitaires, de documentation interne ou externe du projet mais aussi beaucoup de redondances de code qui ne respectaient aucunement les principes SOLID.

Nous avons donc pris la décision de refaire intégralement l'architecture de l'application étant donné que Taskstep n'avait originellement aucune structure, ni même de programmation orientée objet. Ceci se faisant dans le respect des principes SOLID, en documentant chaque classe, interface et fonction publique, et en respectant chacun des critères décrits durant tout cet audit d'optimisation.

Globalement, ces changements permettent de rendre l'application beaucoup plus modulaire et évolutive qu'elle ne l'était auparavant, le respect des bonnes pratiques de programmation en étant en grande partie responsable. D'autres optimisations secondaires n'ont pas pu être finalisées dans les temps (comme la gestion de cas particuliers ou l'implémentation de tests unitaires hors-couche DAO) et pourraient être réalisés lors d'une poursuite du projet.

H. Synthèse du projet

Suite à notre intervention, Taskstep a pu connaître une transformation totale, que ce soit sur le plan architectural, fonctionnel ou ergonomique. La refonte de cette architecture suivant une Programmation Orientée Objet, respectant les principes SOLID et une architecture MVC, rend cette application beaucoup plus modulaire et ouverte aux évolutions. Donc si nos optimisations pourraient devenir obsolètes un jour ou l'autre, il sera désormais beaucoup plus aisément de la remettre à niveau ou d'en modifier ses visuels.

La refonte totale des visuels a également permis de rendre son utilisation beaucoup plus intuitive et agréable. Les pages étant désormais responsive, donc adaptée à tout type d'écran, Taskstep est donc également disponible sur téléphone. L'accessibilité du site ayant été nettement améliorée, nous le rendons également accessibles à d'éventuelles personnes en situation de handicap (visuel ou autres).

Finalement, cette application a su être améliorée tant sur ses performances, que sur son accessibilité sur tout type de plateforme et sa sécurité. Nous avons su lui attribuer de nombreux points forts supplémentaires tels que sa modularité à l'aide du respect des principes SOLID, sa meilleure expérience ergonomique selon les critères de Bastien et Scapin ou encore son optimisation de performances. Des points faibles demeurent et l'application reste à sujet de nombreuses améliorations, chose qui est donc désormais possible sans reprendre le projet de 0, car nous avons non seulement réalisé une optimisation totale de Taskstep mais également offert des perspectives beaucoup plus concrètes d'évolution pour cette application, maintenant bâtie sur de solides fondations.

V. Perspectives

Malgré les nombreuses optimisations réalisées durant ce projet, nous aurions beaucoup de perspectives pour de futures modifications du projet :

- Les sections du site web dépendent beaucoup de différentes dates, rendant leur mise en place très éprouvante sur le long terme. Il pourrait être pertinent de faire dépendre les routes de section de la date des tâches déjà enregistrée en base de données
- De même, les autres sections du site web pourraient être arrangés différemment, peut-être rendus modulaires de la même façon que les projets et les contextes
- Plusieurs nouveaux dossiers CSS pourraient donner d'autres styles au site web, et le système de gestion de ce style pourrait également être changé de sorte qu'il modifie plus d'éléments pour améliorer l'accessibilité du site
- La sécurité du site pourrait être améliorée en enregistrant l'adresse IP d'un utilisateur afin de gérer son nombre de connexion, ou encore en exigeant une adresse mail pour s'inscrire sur le site.

VI. Annexes

Annexe 1: Qualité Logicielle.....	14
Annexe 2: Sécurité	44
Annexe 3: Gestion des Données	58
Annexe 4: Performances et Impact Environnemental.....	69
Annexe 5: Accessibilité.....	91
Annexe 6: Ergonomie.....	112

TASKSTEP

C1 – 1PEUDINSPI

- Tristan DAL MOLIN
- Moulay-Wassim ALAOUI
- Jules DUTRION
- Matteo DE MARCO
- Wassim DIOURI

BUT2 Informatique



VII. Ancienne conception du projet

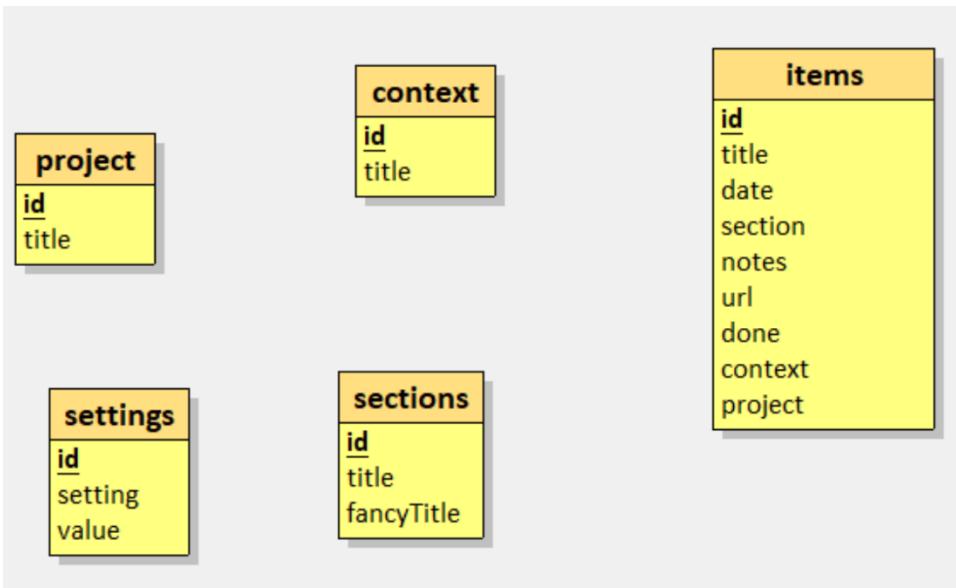


Figure 1 : Modèle de Conception de Données du projet

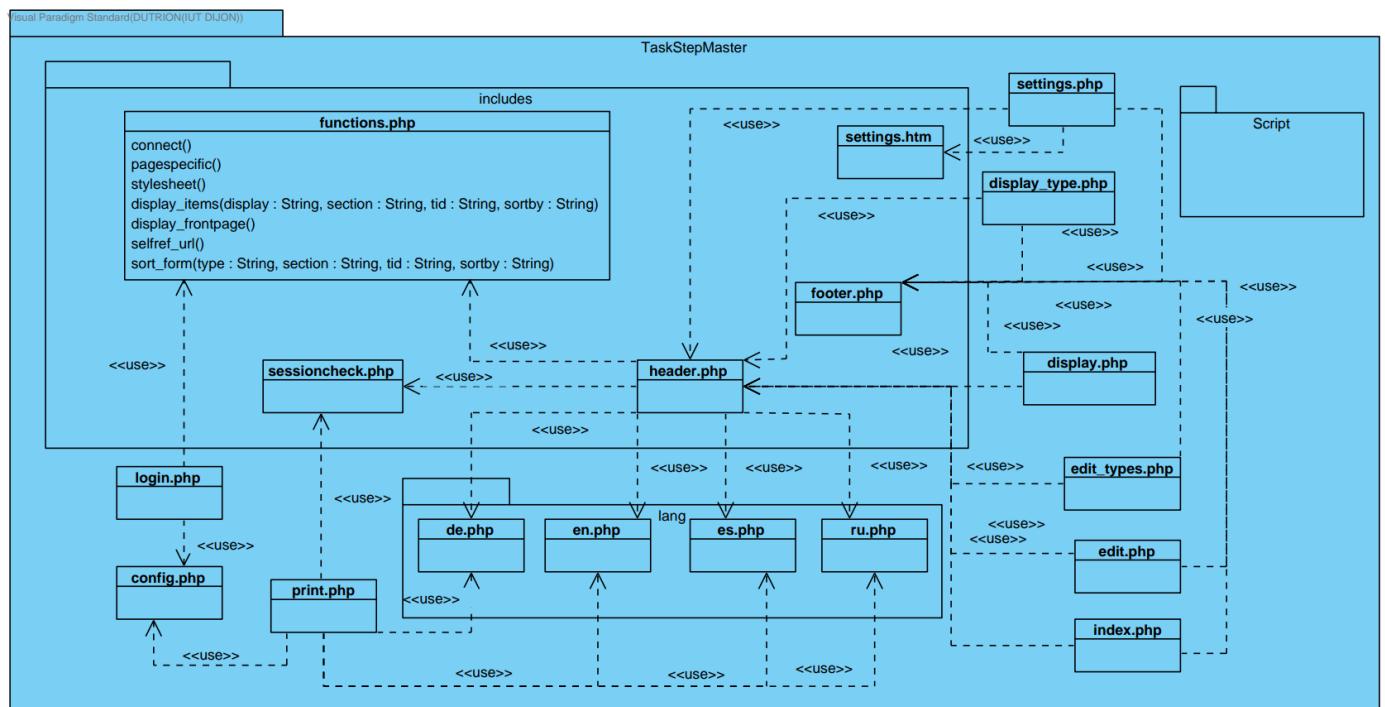


Figure 2 : Conception de l'application (UML)

VIII. Nouvelle conception du projet

Afin de remanier l'application de façon SOLID et optimisée, nous avons refait la structure du projet de 0 par Programmation Orientée Objets en PHP, utilisant notamment une structure MVC :

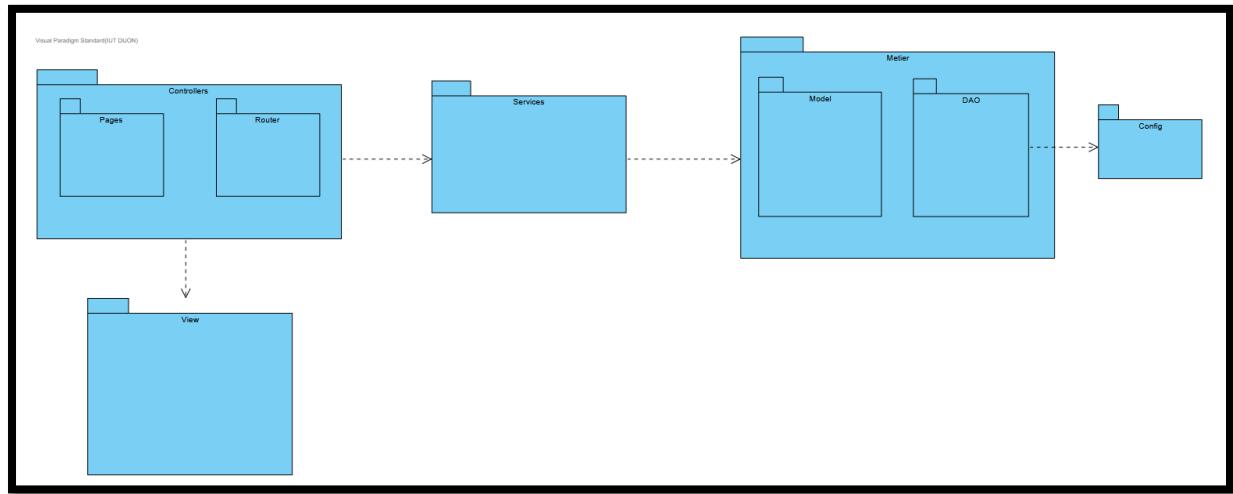


Figure 3 : Diagramme des packages

IX. Audit d'optimisation de qualité logicielle

L'audit qui suit contient les résultats de l'audit principal ainsi que ses corrections et optimisations en dernière colonne.

La notation des résultats est la suivante :

- Le point est une bonne pratique identifiée (point très bien respecté) : BP
- Le point est conforme (c'est respecté, éventuellement avec quelques limites) : CF
- Le point est non conforme (c'est insuffisamment respecté) : NC
- Le point est critique (ce n'est pas du tout respecté, c'est totalement absent...) : CR

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
A – Documentation technique				
A.1 – Documentation externe				
A.1.1 Document indiquant la plateforme d'exécution (Qu'installer sur une VM pour exécuter l'appli)	Visual Studio Code	Lecture du fichier README.htm pour analyser les étapes d'installation locale et vérifier les composants nécessaires (AMP).	BP – L'application doit être exécutée sur un serveur local type XAMPP contenant Apache, MySQL et PHP et tout est indiqué en README.	Rien à modifier
A.1.2 Document indiquant la plateforme de développement	Visual Studio Code	Lecture du README.md pour vérifier la plateforme de développement	BP – Le projet est développé en PHP. Aucun environnement spécifique n'est imposé, tout est indiqué en README	Rien à modifier
A.1.3 Document décrivant l'architecture globale de l'application	Visual Studio Code	Vérification que l'architecture du projet est présente dans sa documentation (UML, conception, ...)	N/A – Aucune architecture, non applicable	Un fichier .vpp contient la conception complète du projet (cf la nouvelle conception du projet)
A.1.4 Document décrivant les dépendances aux outils tiers de l'application	Visual Studio Code	Lecture du README.md pour vérifier les dépendances aux outils tiers de l'application (API extérieures, logiciels à installer en plus de l'application de base)	BP – Présence des dépendances du projet en README. La plateforme d'exécution nécessite PHP ≥ 7.0, MariaDB et un serveur local pour exécuter le projet	Aucune dépendance supplémentaire
A.1.5 Cahier des charges ou document décrivant les fonctionnalités de l'application	Visual Studio Code	Lecture du README.md pour vérifier si les fonctionnalités y sont décrites. (Diagrammes de séquences, descriptions des fonctionnalités)	CF – Les fonctionnalités sont listées dans le README : organisation des tâches, filtres	Un UML du projet entier a pu être réalisé, comprenant notamment un diagramme de cas d'usage de l'application

			contextuels/projets, affichage auto des tâches du jour, impression, surlignage des tâches urgentes ou en retard, multilingue (Anglais, Russe, Allemand, Espagnol).	
--	--	--	--	--

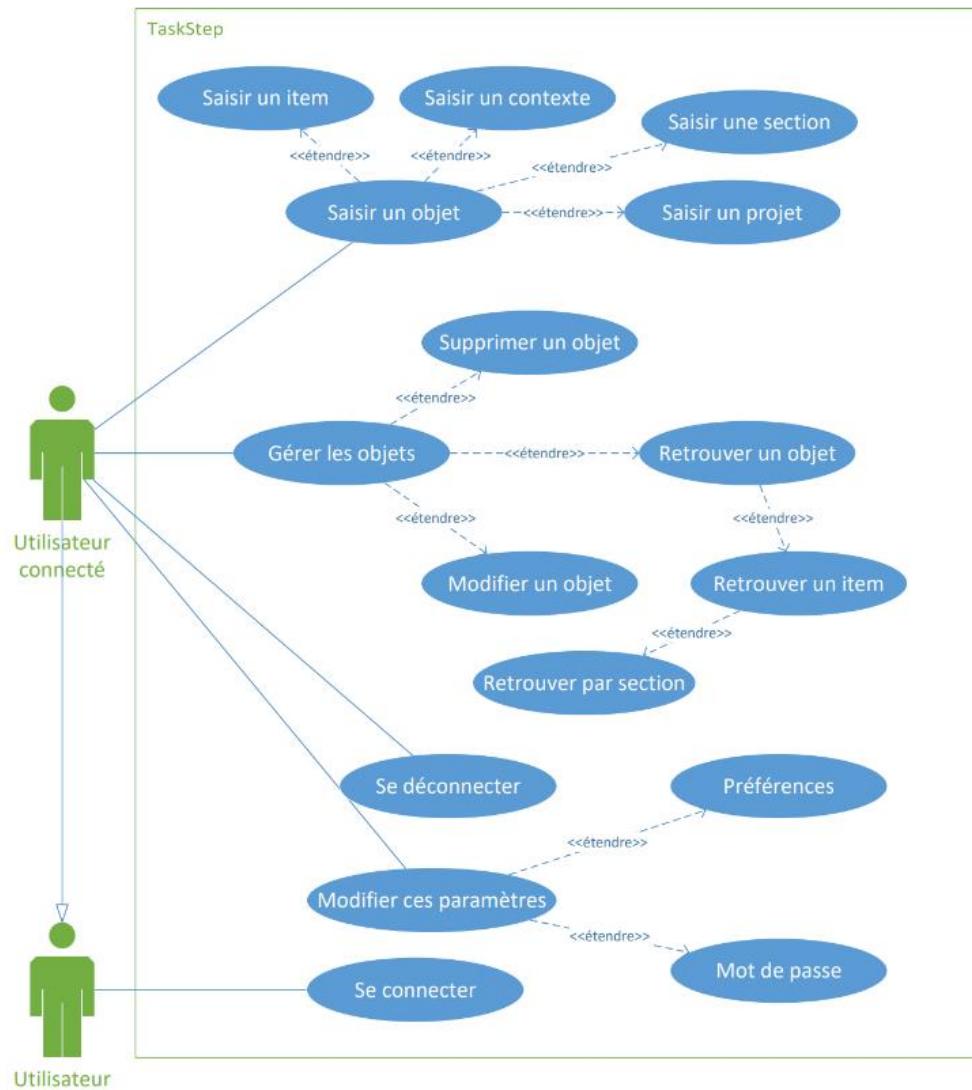


Figure 4 : Diagramme de cas d'usage

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
A – Documentation technique				
A.2 – Documentation interne				
A.2.1 Documentation in-situ du code présente (Code documenté)	Visual Studio Code	Analyse du code existants des différents fichiers présents dans le projet.	CR – Il semble qu'il n'y ait vraiment aucune documentation dans tout le projet. Seulement des commentaires rarement.	
A.2.2 Documentation du code cohérente avec le code et à jour	N/A	Vérification que la documentation a du sens et a une version, que les paramètres y soient ajoutés si ajoutés à une version du site	N/A (Pas de documentation)	Documentation, commentaire, mis en place sur tout le projet (en anglais)
A.2.3 Documentation du code cohérente avec les spécifications fonctionnelles (cf A.1.5., si ça respecte les fonctionnalités décrites)	N/A	Présence de documentation cohérente avec les spécifications	N/A (Pas de documentation)	
A.2.4 Documentation du code claire, lisible, dans la langue souhaitée	N/A	Documentation claire, lisible et dans une même langue	N/A (Pas de documentation)	

Controllers \ Pages \ Home

ControllerHome

in package Application

[ControllerHome.php](#) : 10

The controller for the home page

Table of Contents

Properties

P  \$page : [HomePage](#)

Methods

M  [__construct\(\)](#) : mixed

The class constructor

M  [index\(\)](#) : void

Method showing the home page

Namespaces

Config
Controllers
 Pages
 Router

Logic
 DAO
 Model

PSR4
Service
 Interface

TestUnitaire
 TestBDD

Utils
Composer
 Autoload

Laravel
 SerializableClosure

Invoker
 Exception
 ParameterResolver

DI
 Attribute
 Compiler
 Definition
 Factory
 Invoker

Logic \ DAO \ Interface

I_TaskDAO

in Application

DAO interface for the task table

Table of Contents

Methods

- Ⓜ [AddTask\(\)](#) : void
Method to add a task to the database
- Ⓜ [DeleteTask\(\)](#) : void
Method to delete a task in the database
- Ⓜ [GetTask\(\)](#) : [Task](#)
Method to get a task from the database
- Ⓜ [ListTask\(\)](#) : array<string|int, [Task](#)>
Method to list the tasks
- Ⓜ [ListTaskByContext\(\)](#) : array<string|int, [Task](#)>
Method to list the tasks by section
- Ⓜ [ListTaskByProject\(\)](#) : array<string|int, [Task](#)>
Method to list the tasks by project
- Ⓜ [ListTaskBySection\(\)](#) : array<string|int, [Task](#)>
Method to list the tasks by section
- Ⓜ [TaskDone\(\)](#) : void
Method to set a task as done

Ⓜ [UpdateTask\(\)](#) : void

On this page

Table Of Contents

[Constants](#)

[Methods](#)

Methods

[AddTask\(\)](#)
[DeleteTask\(\)](#)
[GetTask\(\)](#)
[ListTask\(\)](#)
[ListTaskByContext\(\)](#)
[ListTaskByProject\(\)](#)
[ListTaskBySection\(\)](#)
[TaskDone\(\)](#)
[UpdateTask\(\)](#)

Routes

[Routes.php](#) : 57

extends [RoutesAbstract](#)
in package Application

Class listing all the roads of the project

Table of Contents

Properties

- Ⓟ [\\$container](#) : [ContainerInterface](#)
- Ⓟ [\\$ctrlList](#) : array<string|int, mixed>
- Ⓟ [\\$routeList](#) : array<string|int, mixed>

Methods

- Ⓜ [__construct\(\)](#) : mixed
The class constructor
- Ⓜ [createControllerList\(\)](#) : void
Method to initialize the list of controllers
- Ⓜ [createRouteList\(\)](#) : void
Method to initialize the list of routes
- Ⓜ [getConnexionRoute\(\)](#) : string
Method to obtain the name of the connexion page
- Ⓜ [getDefaultRoute\(\)](#) : string
The default route getter for the app
- Ⓜ [getRoute\(\)](#) : [Route](#)
Method returning the asked route or the default one

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
B – Tests de l’application				
B.1 – Tests fonctionnels				
B.1.1 Les cas d’usage de l’application sont tous implémentés (Respect des diagrammes d’usage en conception)	N/A	Vérification que les cas d’usage sont tous respectés	N/A -- Pas de diagrammes de conceptions	/
B.1.2 Les spécifications fonctionnelles sont respectées	Firefox/ Edge	Vérifications que les fonctionnalités de documentation sont bien respectées	CF – La plupart sont conformes, excepté le changement de langue du navigateur (changement de langue possible via le fichier config)	Changement de langue implémenté selon la langue du navigateur directement

Exemple de changement de langue :

Page d'accueil

15/05/2025

Accueil Tous les éléments Par Contexte Par Projet

Formulaire de tâche

Pas de tâches immédiates en attente!

Ajouter une tâche

Astuce: Ajouter une tâche avec le bouton juste ici

Bienvenue sur TaskStep

Bonjour, et bienvenue sur TaskStep. TaskStep est conçu pour vous aider avec vos tâches quotidiennes, vos objectifs à long terme et les listes générales, en les organisant vaguement selon le style avec des contextes et des projets.

Pour ceux d'entre vous qui ne connaissent pas ce concept, tout ce qui nécessite plus d'une étape/action est un projet. Un contexte est l'environnement où vous effectuerez l'action, par exemple sur votre ordinateur.

Aide

Pour plus d'informations cliquez sur ce bouton

Home Page

15/05/2025

Home All items By Context By Project

Paramètres de langue des pages web

Certaines pages web sont proposées dans plusieurs langues. Choisissez les langues d'affichage de ces pages, par ordre de préférence

Anglais (États-Unis) [en-us] Selected	Monter
Français (France) [fr-fr]	Descendre
Français [fr]	Supprimer
Anglais [en]	

Choisir une langue à ajouter... ▼

OK Annuler

Task form

No tasks immediately

Add a task

Tip: Add a task using the button right here

Welcome to TaskStep

Hello, and welcome to TaskStep. TaskStep is designed to help you with your daily tasks, long-term goals, and general lists by loosely organizing them with contexts and projects.

For those of you unfamiliar with this concept, anything that requires more than one step/action is a project. A context is the environment where you will perform the action, such as at your computer.

Help

For more informations click on this button

B.1.3 L'application fonctionne sur les plateformes d'exécution spécifiées	Oracle vm virtualbox	Vérifier que l'application fonctionne sous machine virtuelle en suivant le guide d'installation et d'utilisation.	CF – Tutoriel de mise en place du serveur fonctionnel, à mettre à jour avec la mise à jour du projet	Fichier README mis à jour avec l'optimisation du logiciel
---	----------------------	---	--	---

Installation

- Create database, user and rights. Use your own secure credentials!

```
CREATE DATABASE taskstep;
CREATE USER 'taskstep'@'localhost' IDENTIFIED BY 'taskstep';
GRANT ALL PRIVILEGES ON taskstep.* TO 'taskstep'@'localhost';
```



- Go to installation URL: <https://www.example.com/taskstep/install/install.php>
- Go to the main URL to connect to the app : <https://www.example.com/taskstep/index.php?action=ConnexionPage>
- If you want, and for security reasons, remove `install.php` from the project once your configuration is done
- Create an account and connect to use the application

B.1.4 L'application est robuste aux cas particuliers	Firefox	Vérification qu'aucune mauvaise saisie n'est possible (lettres à la place d'un nombre par exemple)	NC – Insertion possibles avec des champs nécessaires vides, même nom de contexte et de projet possibles, une date peut être entrée en tant que texte et non que date	Formulaire mis à jour, empêchant d'entrer autre chose qu'un lien dans le champs URL ou autre chose qu'une date en date par exemple
--	---------	--	--	--

URL(*)

Pas un URL

Veuillez saisir une URL.

Soumettre

Date limite(*)

22/07/2010



Soumettre

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
B – Tests de l'application				
B.2 – Tests unitaires				
B.2.1 Les tests unitaires sont présents et cohérents avec les spécifications	Visual Studio Code	Vérification que l'application a des tests unitaires cohérents pour les cas généraux et particuliers	NC – Pas de tests unitaires	Tests unitaires mis en place séparément sur le site web, vérifiant les interactions avec la base de données. Seuls les tests en couche DAO ont pu être réalisés, ils sont tous fonctionnels
B.2.2 Les tests unitaires passent tous	Visual Studio Code	Vérification que tous les tests unitaires passent	N/A -- Pas de tests unitaires	
B.2.3 Les tests unitaires sont indépendants les uns des autres	Visual Studio Code	Tests unitaires indépendants entre eux, simulation de couches inférieures, fakes DAO par exemple	N/A -- Pas de tests unitaires	
B.2.4 les cas particuliers (entrées invalides, effets de bord) sont testés	Visual Studio Code	Vérification que les cas particuliers sont testés	N/A -- Pas de tests unitaires	
B.2.5 la couverture des tests est correcte	Visual Studio Code	Couverture des tests correcte (pourcentage éventuel)	N/A -- Pas de tests unitaires	

Tests Unitaires

Unit test for Context

Insert Test : Success

ListContext Test : Success

GetByld Test : Success

Update Test : Success

Delete Test : Success

Unit test for Project

Insert Test : Success

GetByld Test : Success

Update Test : Success

list all Test : Success

Delete Test : Success

Unit test for the Task Table

Insert Test : Success

GetByld Test : Success

Update Test : Success

Delete Test : Success

List all Test : Success

List by Context Test : Success

List by Project Test : Success

List by Section Test : Success

Unit test for Section

TestAddSection : Success

TestGetSection : Success

TestUpdateSection : Success

TestDeleteSection : Success

TestListSection : Success

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
C – Format du code source				
C.1 – Construction du code				
C.1.1 Lors de la construction du code (compilation) il n'y a pas d'avertissement (À modifier si c'est dangereux, peuvent être laissés de côté sinon)	« phpstan » sous Composer et Linux, Visual Studio Code	Commande « phpstan analyse » effectuée sur chaque classe PHP du projet	NC – Plus de 700 erreurs sont trouvées au total, dont beaucoup de méthodes appelées à des éléments	
C.1.2 Lors de la construction du code il n'y a pas d'erreurs	« phpstan » sous Composer et Linux, Visual Studio Code	Commande « phpstan analyse » effectuée sur chaque classe PHP du projet, sous une configuration nous permettant d'obtenir tous les résultats possibles	NC – Plus de 700 erreurs sont trouvées au total, beaucoup de non-typage, de méthodes appelées sur des éléments « mixed »	Après refonte du projet dans une version plus récente de PHP, le test sous phpstan ne renvoie désormais plus aucune erreur

```
root@PC-RCC:/mnt/c/xampp/htdocs/S4_C1_1PeuDinspi# vendor/bin/phpstan analyse Config Logic Controllers Service Views Utils install.php Defines.php index.php Psr4AutoloaderClass.php
113/113 [██████████] 100% [OK] No errors

💡 Tip of the Day:
PHPStan is performing only the most basic checks.
You can pass a higher rule level through the --level option
(the default and current level is 0) to analyse code more thoroughly.
```

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
C – Format du code source				
C.2 – Aspect du code source				
C.2.1 L'indentation du code est correcte	Visual Studio Code, Webimpress Coding Standard	On effectue une analyse du code sous Webimpress Coding Standard afin de vérifier le nombre d'erreurs d'indentation	NC- On remarque donc la présence de presque 2000 erreurs d'indentation dans ce projet (voir figure ci-dessous)	Après analyse avec Webimpress Coding Standards et différents nettoyages du projet, plus aucune erreur d'indentation n'est finalement détectée.

```

analyse.txt
1464
1465
1466 esc[1mFILE: /mnt/c/xampp/htdocs/S4_C1_1PeuDinspi/Views/User/RegisterPage.phpesc[0m
1467 -----
1468 esc[1mFOUND 1 ERROR AND 4 WARNINGS AFFECTING 5 LINESesc[0m
1469 -----
1470 | 22 | esc[31mERROResc[0m | Protected method name "RegisterPage::GenerateContent" is not in
1471 | | camel caps format
1472 | 33 | esc[33mWARNINGesc[0m | Line exceeds 120 characters; contains 155 characters
1473 | 36 | esc[33mWARNINGesc[0m | Line exceeds 120 characters; contains 167 characters
1474 | 39 | esc[33mWARNINGesc[0m | Line exceeds 120 characters; contains 174 characters
1475 | 44 | esc[33mWARNINGesc[0m | Line exceeds 120 characters; contains 186 characters
1476 -----
1477
1478
1479 esc[1mFILE: /mnt/c/xampp/htdocs/S4_C1_1PeuDinspi/index.phpesc[0m
1480 -----
1481 esc[1mFOUND 1 ERROR AFFECTING 1 LINEesc[0m
1482 -----
1483 | 47 | esc[31mERROResc[0m | Default case in switch should be as last; another case found here

```

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
C.2.2 Il n'y a pas d'espace inutiles (blancs) dans le code	Visual Studio Code	Vérification dans chaque fichier qu'aucun grand espace inutile n'est laissé	BP – Aucun réel espace inutile ne se trouve dans le code	/
C.2.3 Le code est facile à lire	Visual Studio Code	On vérifie la présence de commentaires si une fonction n'est pas assez claire sur son propos, la bonne nomenclature de variables et de fonctions, etc.	NC – Les variables sont bien nommées, certains scripts PHP sont correctement commentés (bien que manquant de documentation), mais beaucoup manquent de clarifications et il est difficile de comprendre quelle classe fait quoi (voir figures ci-dessous)	Durant la refonte architecturale du projet, nous avons bien pris le soin, non seulement de rajouter la documentation nécessaire, mais aussi de nommer correctement les différentes variables. La séparation du projet en plusieurs couches rend également plus compréhensible la responsabilité de chacun

```

Views > Task > ListTaskPage.php > PHP > ListTaskPage
11 class ListTaskPage extends AbstractPage {
12 |
13     | reference [0 overrides] prototype
14     | protected function GenerateContent(): string
15     {
16         |
17         $returnValue = "<main class='main-content'>";
18         |
19         $returnValue .= $this->GenerateForm();
20         |
21         if (empty($this->tasks)) {
22             $returnValue .= "<p class='empty-message'>" . $this->languageManager->translate(key: "NoTasksFound") . "</p>";
23         } else {
24             $returnValue .= "<div class='item-list'>";
25             foreach ($this->tasks as $task) {
26                 |
27                 $title = htmlspecialchars(string: $task->getTitle());
28                 $date = $task->getDate() ? $task->getDate()->format(format: "Y-m-d") : "N/A";
29                 $done = $task->getDone();
30                 $colorClass = $done ? "presentation-card" : "delete-card";
31                 $contextTitle = htmlspecialchars(string: $task->getContext()?->getTitle() ?? "No context");
32                 $projectTitle = htmlspecialchars(string: $task->getProject()?->getTitle() ?? "No project");
33                 |
34                 $returnValue .= "
35                     <div class='card $colorClass'>
36                         <h2 class='card-title'$title</h2>
37                         <p><strong>Context:</strong> $contextTitle</p>
38                         <p><strong>Project:</strong> $projectTitle</p>
39                         <p><strong>Date:</strong> $date</p>
40                         <p><strong>Status:</strong> " . ($done ? "✓ " . $this->languageManager->translate(key: "isDone") : "✗ " . $this->languageManager->translate(key: "notDone")) . "</p>
41                         <div class='item-actions'>
42                             <a href='index.php?action=ViewTask&id=' . $task->getId() . "' class='btn btn-view'" . $this->languageManager->translate(key: "View") . "</a>
43                             <a href='index.php?action=TaskDone&id=' . $task->getId() . "' class='btn btn-edit'" . $this->languageManager->translate(key: "GetDone") . "</a>
44                             <a href='index.php?action=UpdateTask&id=' . $task->getId() . "' class='btn btn-edit'" . $this->languageManager->translate(key: "Edit") . "</a>
45                             <a href='index.php?action=DeleteTask&id=' . $task->getId() . "' class='btn btn-delete'" . $this->languageManager->translate(key: "Delete") . "</a>
46                         </div>
47                     </div>
48                 ";
49             }
50             $returnValue .= "</div>";
51         }
52         |
53         $returnValue .= "</main>";
54         return $returnValue;
55     }
56 
```

```

13  class ControllerSettings
14 {
15     /**
16      * 6 references
17      * private SettingsPage $page;
18      * 2 references
19      * private I_UserService $userService;
20
21      /**
22      * The class constructor
23      *
24      * @param I_UserService $userService The user's service
25      */
26
27      0 references|0 overrides
28      public function __construct(I_UserService $userService)
29      {
30         $this->page = new SettingsPage();
31         $this->userService = $userService;
32     }
33
34     /**
35      * Method showing the settings page
36      */
37
38     3 references|0 overrides
39     public function index(): void
40     {
41         echo $this->page->GeneratePage();
42     }
43
44     /**
45      * Method to change a password
46      *
47      * @param string $login The user's login
48      * @param string $password The user's password
49      * @param string $newPassword The new user's password
50      */
51
52     1 reference|0 overrides
53     public function ChangePassword(string $password, string $newPassword, string $confirmPassword): void
54     {
55         $this->userService->ChangePassword(login: $SESSION['username'], password: $password, newPassword: $newPassword, confirmPassword: $confirmPassword);
56         $this->page->setMsg(msg: new Message(message: 'SuccessPassChange', error: false));
57         echo $this->page->GeneratePage();
58     }
59
60     /**
61      * Method to change the style
62      *
63      * @param string $styleName The user's style choice
64      */
65
66     1 reference|0 overrides
67     public function ChangeDisplay(string $styleName): void

```

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
C.2.4 Le code est dans la langue spécifiée	Visual Studio Code	On vérifie que le code du projet entier est écrit en une même langue	BP – Le code entier est en anglais, jusque dans sa documentation et ses commentaires	Le code est toujours en anglais, y compris dans sa documentation
C.2.5 La taille du code est correcte	Visual Studio Code	Vérification que chaque fonction fait moins de 30 lignes et qu'une classe/un fichier fasse moins de 300 lignes	CF – Chaque fichier fait effectivement moins de 300 lignes mais beaucoup de fonctions dépassent les 30 lignes (voir figure ci-dessous), refactoring et séparation des responsabilités nécessaires	Aux classes de visuel près, chaque classe fait en effet moins de 300 lignes pour la plupart (d'autres pouvant encore être redéfinies en plusieurs autres, mais n'ayant pas pu l'être par manque de temps, de même pour différentes fonctions)

```

11  /**
12   * The controller for the website's settings */
13  14 references | 0 implementations
14  ✓ class ControllerSettings
15  {
16      6 references
17      private SettingsPage $page;
18      2 references
19      private I_UserService $userService;
20
21      /**
22       * ...
23       * 0 references | 0 overrides
24      > public function __construct(I_UserService $userService) ...
25      }
26
27      /**
28       * ...
29       * 3 references | 0 overrides
30      > public function index(): void...
31      }
32
33      /**
34       * ...
35       * 1 reference | 0 overrides
36      > public function ChangePassword(string $password, string $newPassword, string $confirmPassword): void...
37      }
38
39      /**
40       * ...
41       * 1 reference | 0 overrides
42      > public function ChangeDisplay(string $styleName): void...
43      }
44
45      /**
46       * ...
47       * 1 reference | 0 overrides
48      > public function ChangeLanguage(string $language): void...
49      }
50
51      /**
52       * ...
53       * 1 reference | 0 overrides
54      > public function ChangeTheme(string $theme): void...
55      }
56
57      /**
58       * ...
59       * 1 reference | 0 overrides
60      > public function ChangeAvatar(string $avatar): void...
61      }
62
63      /**
64       * ...
65       * 1 reference | 0 overrides
66      > public function ChangeEmail(string $email): void...
67      }
68
69

```

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
C.2.6 Le code ne présente pas d'élément redondant	Visual Studio Code	Vérification qu'il n'existe pas 2 classes ou fonctions faisant la même chose (principe DRY), vérifier l'usage d'héritage ou d'interface dans ce cas-là	NC – Les récupérations en BDD depuis du code PHP pourraient être factorisées en un même appel sécurisé	Les différentes récupérations en BDD s'opèrent désormais dans une couche DAO, où des requêtes ont pu être factorisées en une, notamment pour ce qui est des tâches

```

1 reference | 0 overrides
16
17 public function ListTask(User $user, string $sort, string $sortDirection) : array []
18
19     // Verification of the sort parameters
20     $verifiedSorting = $this->verifySorting(sort: $sort, sortDirection: $sortDirection);
21
22     // Request set-up
23     $request = "SELECT T.id AS id, T.title AS title, date, note, url, done, idProject, idSection, idContext, P.title AS project_title, S.title AS section_title
24         FROM TASK AS T
25         JOIN PROJECT AS P ON P.id = T.idProject
26         JOIN CONTEXT AS C ON C.id = T.idContext
27         JOIN SECTION AS S ON S.id = T.idSection
28         WHERE T.idUser=:idUser
29         ORDER BY $verifiedSorting";
30
31     $parameters = array(
32         "idUser" => $user->getId()
33     );
34
35     // Request execution
36     $response = $this->execRequest(sql: $request, params: $parameters);
37
38     if ($response === false) {
39         throw new Exception(message: "Error while fetching the list of tasks");
40     }
41
42     $tasks = $this->setTaskList(response: $response);
43
44     return $tasks;
45
46 }
```



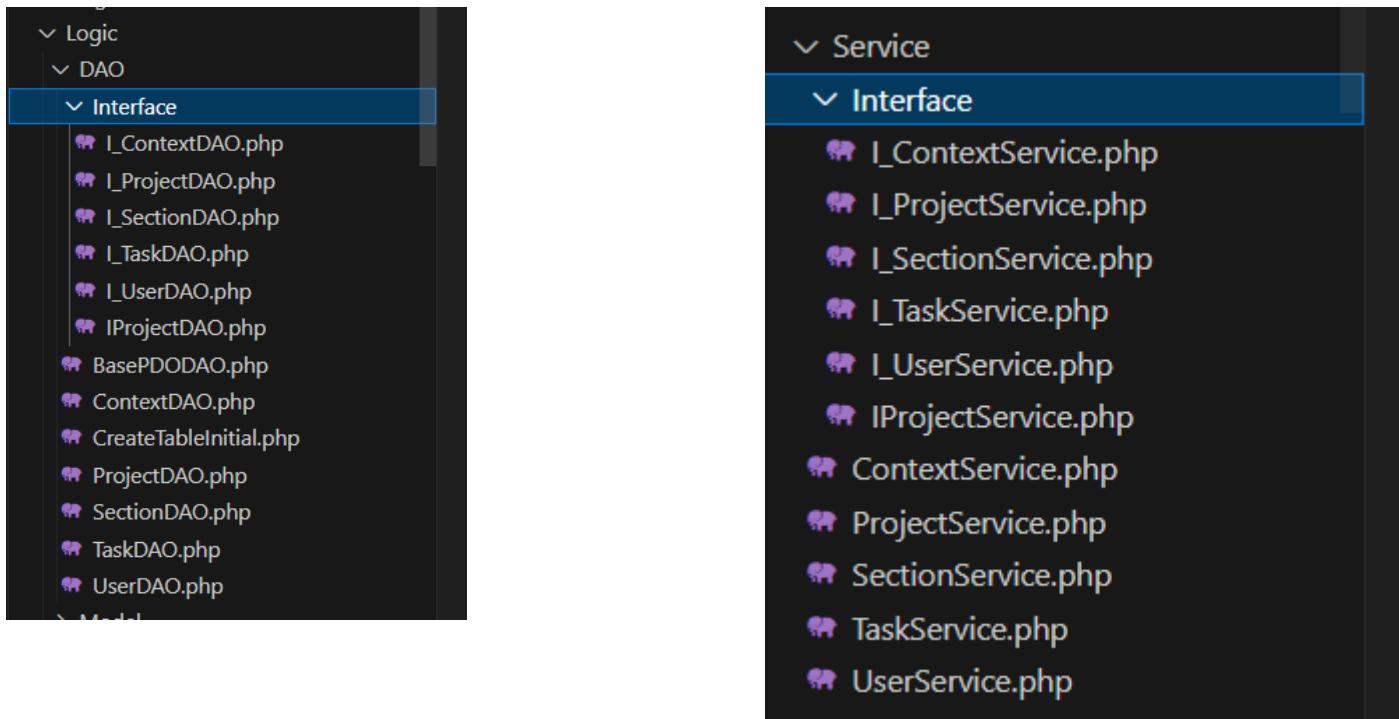
Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
C.2.7 La norme de casse du langage utilisé est respectée	Visual Studio Code, Webimpress Coding Standard	Le camel case est respecté (attributs en minuscule, propriété en majuscule au début, pas d'underscores (ils peuvent être remplacé par une majuscule dans le mot), constantes en majuscules)	NC – Par analyse Webimpress Coding Standard, on détecte déjà presque 600 erreurs de camel case dans le code	N'ayant pas nécessairement suivi toutes les bonnes pratiques durant ce projet, il restera des erreurs de camel case, bien qu'elles soient moins nombreuses que dans le projet original

```

analyse.txt
108
142 | 23 | esc[31mERROResc[0m | [x] Equals sign not aligned correctly; expected 1 space but found
143 |     | 0 spaces
144 | 23 | esc[31mERROResc[0m | [x] Expected 1 space before "="; 0 found
145 | 23 | esc[31mERROResc[0m | [x] Expected 1 space after "="; 0 found
146 | 25 | esc[31mERROResc[0m | [x] Whitespace found at end of line
147 | 28 | esc[31mERROResc[0m | [x] Missing blank line before comment tags
148 | 28 | esc[31mERROResc[0m | [x] Param tag is redundant
149 | 29 | esc[31mERROResc[0m | [x] Return tag with "void" type is redundant
150 | 31 | esc[31mERROResc[0m | [ ] Public method name
151 |     | "ControllerAddUpdateContext::UpdateContextPage" is not in
152 |     | camel caps format
153 | 31 | esc[31mERROResc[0m | [x] There must be exactly 1 space(s) between the closing
154 |     | parenthesis and the colon when declaring a return type for a
155 |     | function
156 | 31 | esc[31mERROResc[0m | [x] Opening brace should be on a new line
157 | 32 | esc[31mERROResc[0m | [x] Expected 1 space before "="; 0 found
158 | 32 | esc[31mERROResc[0m | [x] Expected 1 space after "="; 0 found
159 | 32 | esc[31mERROResc[0m | [x] Equals sign not aligned correctly; expected 1 space but found
160 |     | 0 spaces

```

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
D – Principes de qualité logicielle				
D.1 – Principes SOLID				
D.1.1 Le principe de responsabilité unique (S) est respecté	Vscode	On vérifie si chaque fichier fait une seule tache, grâce à ça il serait plus simple de maintenir l'application dans le futur (Une classe, une responsabilité. On cherche différentes classes faisant la même chose ou une faisant plusieurs choses à la fois)	CR - Le fichier display_type.php ne respecte pas le principe S pour la plupart des fichiers les appels à la base de données et la gestion de l'affichage sont mélangés dans le code ce qui est contraire au principe S : exemple ci-dessous	Une architecture MVC est en place, veillant à ce que chaque fichier ait une responsabilité unique. L'ensemble est structuré avec des classes PHP, ce qui permet de respecter le principe S. De plus, une couche service servira de passerelle entre les couches DAO et Controller, permettant de décharger les responsabilités de l'un ou de l'autre avant d'interagir avec le serveur. (Cf architecture du projet pour plus de détails)
D.1.2 Le principe ouvert-fermé (O) est respecté	Vs code	On vérifie s'il y a une isolation des couches et si pour ajouter une fonctionnalité on peut	CR - il n'y a aucune interface ni de séparation des packages. Si on veut ajouter une fonctionnalité, on est obligé de modifier du code de partout pour que ça marche	La partie métier est isolée à l'aide d'interfaces, ce qui permet d'ajouter de nouvelles fonctionnalités facilement, sans impacter le reste du système. Ceci permet de respecter le principe O. Image d'illustration : Une bibliothèque d'injection de dépendances (PHP-DI) est également installée afin de pouvoir facilement remplacer un service ou un DAO à l'avenir.



```

24
25 // Configuration of the dependecies injection
26 return [
27
28     // DAO
29     I_TaskDAO::class => autowire(className: TaskDAO::class),
30     I_ProjectDAO::class => autowire(className: ProjectDAO::class),
31     I_SectionDAO::class => autowire(className: SectionDAO::class),
32     I_ContextDAO::class => autowire(className: ContextDAO::class),
33     I_UserDAO::class => autowire(className: UserDAO::class),
34
35     // Services
36     I_TaskService::class => autowire(className: TaskService::class),
37     I_ContextService::class => autowire(className: ContextService::class),
38     I_SectionService::class => autowire(className: SectionService::class),
39     I_ProjectService::class => autowire(className: ProjectService::class),
40     I_UserService::class => autowire(className: UserService::class)
41 ];
42
43
44
45 ?>

```

D.1.3	Le principe de substitution de Liskov (L) est respecté	Visual Studio Code	Vérification qu'en cas d'héritage, le module de bas peut remplacer celui de haut niveau sans en changer le fonctionnement (exemple des exceptions levées par les classes filles)	CR – Pas d'objets donc à mettre en place dans le projet	La transformation du projet pour utiliser des classes PHP aurait pu poser un problème vis-à-vis du principe L. Cependant, ce principe SOLID a bien été pris en compte afin d'éviter toute erreur à ce niveau. On retrouve particulièrement des notions d'héritage du côté des vues (ListTaskBySection héritant de ListTask par exemple) et l'un peut aisément remplacer l'autre
-------	--	--------------------	--	---	---

```

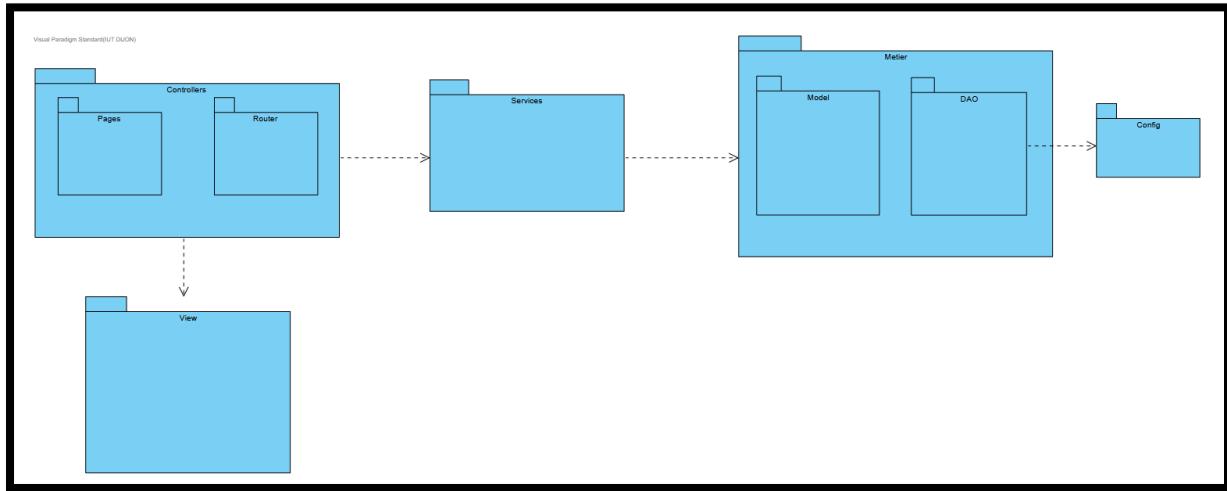
10      6 references | 3 implementations
11  < class ListTaskPage extends AbstractPage {
12
13    > #region Attributes ...
14
15    > /**
16      4 references | 0 overrides
17    > public function setTasks(array $tasks): void ...
18    > }
19
20    > /**
21      18 references | 0 overrides | prototype
22    > public function __construct(): void ...
23    > }
24
25    > /**
26      1 reference | 0 overrides | prototype
27    > protected function GenerateContent(): string
28    > {
29
30      $returnValue = "<main class='main-content'>";
31
32      $returnValue .= $this->GenerateForm();
33
34      if (empty($this->tasks)) {
35        $returnValue .= "<p class='empty-message'>" . $this->languageManager->translate(key: "NoTasksFound") . "</p>";
36      } else {
37        $returnValue .= "<div class='item-list'>";
38        foreach ($this->tasks as $task) {
39          $title = htmlspecialchars(string: $task->getTitle());
40          $date = $task->getDate() ? $task->getDate()->format(format: "Y-m-d") : "N/A";
41
42          $returnValue .= "<div class='item'>" . $title . " - " . $date . "</div>";
43
44        }
45
46      }
47
48    }
49
50  
```

```

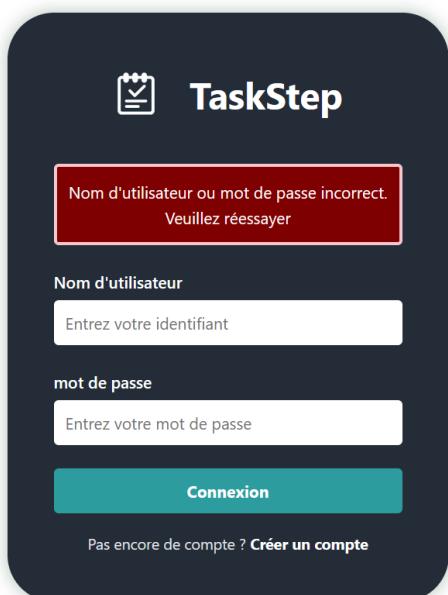
6
7  < /**
8   * Page to list the tasks by context
9   */
10 < 3 references | 0 implementations
11 < class ListTaskByContextPage extends ListTaskPage {
12
13   < 3 references
14   < private Context $context;
15
16   < /**
17     * Set the value of context
18     */
19   < 1 reference | 0 overrides
20   < public function setContext(Context $context): void
21   < {
22     $this->context = $context;
23   }
24
25
26   < 1 reference | 0 overrides | prototype
27   < protected function GenerateForm(): string { ... }
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

```

D.1.4 Le principe de ségrégation des interfaces (I) est respecté	Visual Studio Code	Une classe n'implémente pas une interface dont elle n'a pas besoin de toutes les méthodes, éviter les interfaces pour tout, qui ont trop de méthodes (principe S)	CR – Pas d'objets donc à mettre en place dans le projet	L'ajout d'interface aurait pu poser souci quant au respect du principe I mais nous l'avons bien respecté, comme démontré sur le point D.1.2. sur le principe O
D.1.5 Le principe d'inversion des dépendances (D) est respecté		(Code de haut niveau ne dépend pas de bas niveau (métier qui parle à IHM directement))	CR – Pas d'objets donc à mettre en place dans le projet	<p>Le principe D garantit que les dépendances entre classes vont dans le bon sens et qu'il n'existe pas de dépendances circulaires.</p> <p>Ce point a également été pris en compte dans notre conception (cf. diagramme des classes), ayant une claire séparation des couches : Controllers a accès aux Services qui ont accès aux DAO et aux Models, Controllers ont également accès aux Vues qui ont accès aux Models.</p> <p>Cette séparation des couches évite donc des dépendances circulaires</p>



Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
D – Principes de qualité logicielle				
D.2 – Gestion des erreurs				
D.2.1 – Les cas d'erreurs sont traités à l'aide des exceptions	VsCode	Test via l'IHM et analyse du code pour trouver des erreurs générées	CR – Aucune présence d'exception	
D.2.2 Le point d'entrée du programme capture les exceptions non encore capturées	VsCode	Test via l'IHM et analyse du code pour trouver des erreurs générées et trouver des blocs de try catch	CR – Vu qu'il n'y a aucune exception les seules erreurs affichées viennent d'if/else	Les exceptions sont directement capturées dans le router de sorte que l'on n'ait aucune exception n'étant pas gérée dans l'application
D.2.3 Les exceptions ne sont pas muselées mais capturées pour être traitées (catchs vides, à documenter si c'est volontaire)	VsCode	Test via l'IHM et analyse du code	CR - Aucune présences de try / catch sauf dans la bibliothèque	



TaskStep

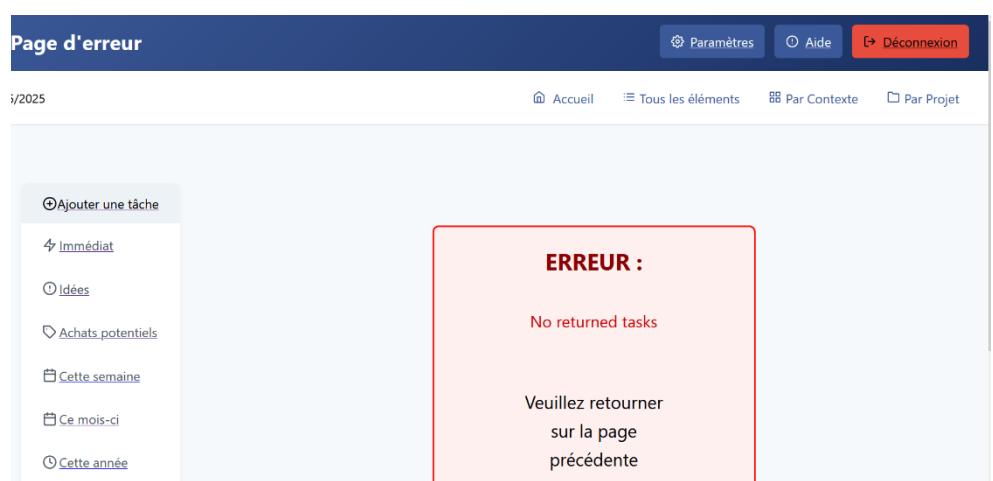
Nom d'utilisateur ou mot de passe incorrect.
Veuillez réessayer

Nom d'utilisateur
Entrez votre identifiant

mot de passe
Entrez votre mot de passe

Connexion

Pas encore de compte ? [Créer un compte](#)



Page d'erreur

1/2025

Paramètres Aide Déconnexion

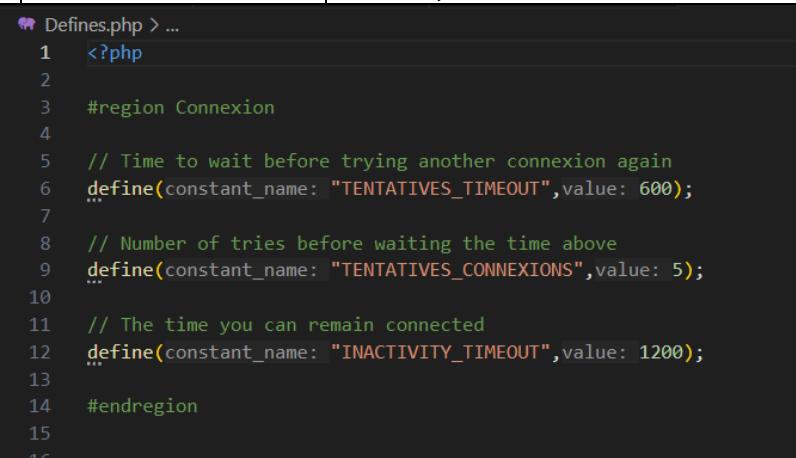
Accueil Tous les éléments Par Contexte Par Projet

Ajouter une tâche

- ↳ Immédiat
- ↳ Idées
- ↳ Achats potentiels
- ↳ Cette semaine
- ↳ Ce mois-ci
- ↳ Cette année

ERREUR :
 No returned tasks
 Veuillez retourner sur la page précédente

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
D – Principes de qualité logicielle				
D.3 – Règles de codage				
D.3.1 – Il n'y a pas de « valeur magique » dans le code	Visual Studio Code	Il n'y a pas de valeurs arbitraires sorties de nulle part dans le code (limites de boucles par exemple)	CF – Une valeur magique se trouve au milieu des fonctions (display_frontpage()) et devrait donc être échappée en dehors de la fonction (variable globale par exemple) ou en paramètre optionnel avec valeur par défaut (voir figures ci-dessous)	Cette valeur magique n'existe maintenant plus et les variables globales de l'application sont contenues dans un fichier séparé pour l'application



```

Defines.php > ...
1  <?php
2
3  #region Connexion
4
5  // Time to wait before trying another connexion again
6  define(constant_name: "TENTATIVES_TIMEOUT", value: 600);
7
8  // Number of tries before waiting the time above
9  define(constant_name: "TENTATIVES_CONNEXIONS", value: 5);
10
11 // The time you can remain connected
12 define(constant_name: "INACTIVITY_TIMEOUT", value: 1200);
13
14 #endregion
15
16

```

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Optimisation
D.3.2 – chaque fonction ne possède qu'un return, sur la dernière ligne de la fonction	Visual Studio Code	Vérification qu'aucune fonction ne possède plus d'un return, ce dernier devant être placé en dernière ligne de fonction	BP – Aucune fonction ne possède plus d'un return dans le projet	De même, après réarrangement du projet, chaque fonction ne possède qu'un return
D.3.3 – aucune procédure ne contient de return	Visual Studio Code	Vérification qu'il n'y a pas de returns en procédure	BP – Il n'y a pas de returns en procédures du projet	Les fonctions ne renvoyant rien ne possèdent pas de return
D.3.4 – il n'y a aucun « goto » ou autre rupture du flux d'exécution	Visual Studio Code	Vérification qu'aucune rupture du flux n'a lieu dans le projet, qu'il n'y a pas de goto renvoyant à une autre ligne de code en particulier, ni de break en milieu de fonction	BP – Il n'y a pas de rupture du flux	Aucune rupture de flux n'a été implémentée du projet

D.3.5 – Le nom des variables est pertinent	Visual Studio Code	On vérifie que les noms donnés sont pertinents comme les noms de variable ou autre	NC – Certains noms ne sont absolument pas pertinents et sont non professionnels parfois par manque d'inspiration le nom est le même avec un numéro ce qui rend le code encore plus illisible	Chaque variable a été nommée de façon cohérente et logique avec l'application
D.3.6 – le nom des opérations (fonctions, procédures) est pertinent	Visual studio code	Vérification que les noms des fonctions est pertinent et clair	NC – Certains noms de fonction ne sont pas très clair et ne possède pas de nomenclature	De même, chaque fonction et procédure est nommée de façon pertinente, les rendant reconnaissables et utilisables

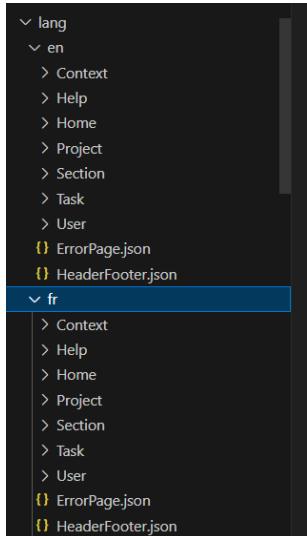
```

6
  18 references | 1 implementation
7 < interface I_SectionService{
8
9 >   /**
10    * ...
11    * 1 reference | 1 override
12    */
13    public function AddSection(string $title);
14
15 >   /**
16    * ...
17    * 1 reference | 1 override
18    */
19    public function UpdateSection(int $id,string $title);
20
21 >   /**
22    * ...
23    * 1 reference | 1 override
24    */
25    public function DeleteSection(int $id);
26
27 >   /**
28    * ...
29    * 1 reference | 1 override
30    */
31    public function ListSection():array;
32
33 >   /**
34    * ...
35    * 3 references | 1 override
36    */
37    public function GetSection(int $id):Section;
38
39 }

```

```
2 references | 0 implementations
8 ▼ class ProjectCreateRoute extends Route {
    3 references
9     private ControllerAddUpdateProject $controllerProject;
10    /**
11     * constructor
12     * @param \Controllers\Pages\Project\ControllerAddUpdateProject $controllerProject project controller
13     */
14    1 reference | 0 overrides
15    public function __construct(ControllerAddUpdateProject $controllerProject){
16        $this->controllerProject = $controllerProject;
17    }
18    #region methodes
19    1 reference | 0 overrides | prototype
20    public function get(array $params=[]): void{
21        $this->controllerProject->AddProjectPage();
22    }
23    1 reference | 0 overrides | prototype
24    public function post(array $params=[]): void{
25        $title = parent::getParam(array: $params, paramName: "title", canBeEmpty: false);
26        $this->controllerProject->AddProject(title: $title);
27    }
28    #endregion
29
30    /**
31     * Abstract class for a route
32     */
33    131 references | 31 implementations
34    abstract class Route {
35
36        /**
37         * Method to act on a road, calling the method GET or POST based on the given parameter
38         * @param array $params Parameters
39         * @param string $method Action method (get by default)
40         * @return void
41         */
42        2 references | 0 overrides
43        public function action(array $params = [], string $method='GET'): void {
44
45            try {
46                switch ($method) {
47                    case 'GET':
48                        $this->get(params: $params);
49                        break;
50
51                    case 'POST':
52                        $this->post(params: $params);
53                        break;
54                }
55            } catch (Exception $e) {
56                $controller = new ControllerError();
57                $controller->index(message: $e->getMessage());
58            }
59
60        }
61
62    }
63
```

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
D.3.7 – il n'y a pas de code « mort » (commenté, non atteignable...) (fonctions jamais appelées, ou alors code qui n'est jamais exécuté du programme)	Visual Studio Code	Vérification qu'il n'y a pas de code commenté ou non atteignable, de fonctions jamais appelées ou de code qui n'est jamais exécuté du programme	CF – Plusieurs zones de code sont commentées	Aucun code mort n'a été laissé dans ce nouveau projet
E – Pile technique				
E.1 – Langage utilisé				
E.1.1 – Le(s) langage(s) utilisés sont pertinents pour le problème donné (Exemple non pertinents : application en Python, IA en C#)	Visual Studio Code, XAMPP	Vérifications que les langages utilisés ne sont pas hors-sujet et sont pertinents à utiliser.	CF – Comme il s'agit d'un site web, le code PHP est adapté dans l'ensemble à de l'interaction côté serveur et à la communication avec une Base de Données Pour la base de données la communication se fait par l'utilisation de MySQL Pour la traduction, quant à elle, elle n'est pas réalisée dans le bon langage, elle est réalisée en PHP alors qu'elle devrait être en .json	Le reste des éléments étant corrects dans l'ensemble, nous avons simplement changé le système de traduction afin qu'il utilise des fichiers .json pour traduire le site en différentes langues



```

10 class LanguageManager {
11     /**
12      * Method to get a traduction from the loaded files
13      * @param string $key The table key
14      * @return string Value associated to the entered key
15     */
16     public function translate(string $key) : string {
17         return htmlspecialchars(string: $this->translations[$key] ?? htmlspecialchars(string: $key));
18     }
19
20     /**
21      * Method to get a translation from the loaded files without special chars
22      * @param string $key The table key
23      * @return string Value associated to the entered key
24     */
25     public function translate_without_specialchars(string $key) : string {
26         return $this->translations[$key] ?? $key;
27     }
28 }

```

The code editor displays a PHP class named 'LanguageManager'. It contains two methods: 'translate' and 'translate_without_specialchars'. Both methods take a string parameter '\$key' and return a string. The 'translate' method uses 'htmlspecialchars' to encode the value from the translations array. The 'translate_without_specialchars' method returns the value from the translations array directly if it exists, otherwise returns the '\$key'.

E.1.2 – La version utilisée du langage est supportée et maintenue (Ex : page en PHP5, failles non corrigées aujourd’hui)	Visual Studio Code, README	Vérification que le langage de programmation est supporté par les navigateurs web et maintenu encore aujourd’hui (lecture du README pour trouver les versions)	CR – La version utilisée est php 7.0 qui n'est plus maintenue, il faut mettre à jour les différentes méthodes n'étant plus d'actualité (PDO pour interactions BDD, POO, ...)	Le code a pu être mis à jour et codé en PHP 8.2, permettant ainsi des interactions plus sécurisées avec la base de données, évitant notamment les injections SQL
--	----------------------------	--	--	--

E.2 – Outils tiers

E.2.1 – Les bibliothèques/frameworks utilisés sont pertinents et réellement utiles	Visual Studio Code	Analyse du code, on vérifie que les bibliothèques utilisées servent réellement et n'ont pas un trop grand nombre de fonctionnalité pour un usage trop petit	NC – L'unique bibliothèque utilisée était utile en 2006 mais n'est plus utiles actuellement est pris en charge nativement par HTML depuis 2015 (voir figure ci-dessous)	La bibliothèque des dates en HTML a donc été retirée au profit d'une simple balise <input type="date"> Une seule bibliothèque sera désormais utilisée, servant à l'injection de dépendances en PHP (PHP-DI 7.0)
E.2.2 – Les bibliothèques/frameworks utilisés sont dans une version supportée et maintenue	Visual Studio Code	Vérification que la ou les bibliothèques du projet sont encore d'actualité et maintenue(s), si besoin, les mettre à jour	NC – La seule bibliothèque utilisée n'est plus maintenue.	PHP-DI est encore actuellement maintenu et supporté, fonctionnant parfaitement bien avec PHP 8.2

X. Tableau des tâches d'optimisation réalisées

Le tableau suivant liste les tâches d'optimisation qui étaient définies par l'audit de qualité logicielle, la dernière colonne indiquera en vert si la tâche a pu être réalisée, en orange si elle ne l'est que partiellement, en rouge si elle n'a pas pu l'être

Recommandations	Priorité	Complexité	Effort	gravité	Fait / Non fait
tutoriel hébergement local	1	1	1	1	
changement de langue	2	3	2	2	
tests unitaires	2	3	2	1	
présence de valeur arbitraire	2	2	2	2	
code mort	2	1	1	1	
bibliothèque pour la date n'est plus utile maintenant	2	1	1	1	
bibliothèque non maintenue	2	1	1	1	
mauvaise gestion des cas particulier	3	3	2	2	
erreur et warning dans le code	3	2	3	2	
non typage	3	1	2	2	
indentation	3	1	1	1	
code difficile à lire	3	3	2	2	
non respect du camel case	3	2	2	1	
traduction en PHP au lieu de json	3	3	2	2	
documentation	4	1	2	1	
fonction trop longue	4	2	2	2	
non présence d'exception	4	3	2	3	
non présence de try catch (sauf dans la bibliothèque)	4	2	3	3	
noms impertinents (variables et fonctions)	4	4	3	3	
mettre à jour la version de PHP	4	4	3	3	
diagramme cas d'usage	5	3	2	2	
mélange requête SQL et code PHP	5	3	2	3	
architecture	6	3	3	3	
principe SOLID non respecté	6	6	4	3	

Par manque de temps, nous n'avons pas pu analyser chaque cas particulier du projet et n'avons pas pu résoudre tous les warnings du projet, mais le reste des éléments, notamment tous ceux à priorité très élevée, a pu être corrigé, rendant donc globalement l'application beaucoup plus qualitative en termes de code.

XI. Synthèse d'optimisation

L'application a pu être jugée selon les principes SOLID, les bonnes pratiques de programmation et le PSR (PHP Standards Recommandations) durant l'audit de qualité logicielle. En découlait donc les soucis d'architecture, de documentations, de tests ou même de respect des principes évoqués précédemment.

Nous avons donc pris la décision de refaire intégralement l'architecture de l'application étant donné que Taskstep n'avait originellement aucune structure, ni même de programmation orientée objet. Ceci se faisant dans le respect des principes SOLID, en documentant chaque classe, interface et fonction publique, et en respectant chacun des critères décrits durant tout cet audit d'optimisation.

Pour ce qui est de la sécurité et de la gestion des données dans le code, il aura été réalisé de sorte qu'aucune injection SQL ne soit possible. Des tests unitaires de la couche DAO, donc pour les interactions avec la base de données, auront également été réalisés afin de pouvoir tester la validité des tables du projet.

La traduction du site est désormais gérée par différents fichiers JSON, rendant la mise en place d'une nouvelle langue beaucoup plus facile.

Globalement, ces changements permettent de rendre l'application beaucoup plus modulaire et évolutive qu'elle ne l'était auparavant, le respect des bonnes pratiques de programmation en étant en grande partie responsable. D'autres optimisations secondaires n'ont pas pu être finalisées dans les temps (comme la gestion de cas particuliers ou l'implémentation de tests unitaires hors-couche DAO) et pourraient être réalisés lors d'une poursuite du projet.

TASKSTEP

C1 – 1PEUDINSPI

- Tristan DAL MOLIN
- Moulay-Wassim ALAOUI
- Jules DUTRION
- Matteo DE MARCO
- Wassim DIOURI

BUT2 Informatique



xii. Remise en contexte

A. Liste des types d'utilisateurs

Il n'existe qu'un type d'utilisateur : ceux qui verront leurs tâches être affichées dans Taskstep

B. Liste des composants de l'application

Nom du composant	Localisation	Restriction d'accès
BDD	Lieu d'hébergement du serveur	Administrateur du service et serveur
Serveur	Lieu d'hébergement du serveur	Administrateur du service
Client	Chez l'utilisateur	Public

À noter que l'application peut être utilisée localement, auquel cas tout se trouvera sur la même machine.

C. Liste des données manipulées

Données	Accès en lecture			Accès en écriture		
	Utilisateurs cibles	Risques	Sensibilité	Utilisateurs cibles	Risques	Sensibilité
Login de l'utilisateur	Utilisateur concerné	X	Pas sensible	X	Perte de fonctionnalité	Sensible
Mot de passe	Utilisateur concerné	Usurpation d'identité	Très sensible	Utilisateur concerné	Perte de l'usage du mot de passe	Très sensible
Item	Utilisateur concerné	X	Sensible	Utilisateur concerné	X	Aucune sensibilité
Paramètre	Utilisateur concerné	X	Pas de risque	Utilisateur concerné	X	Aucune sensibilité

XIII. Analyse des communications

A. Liste des communications entre composants

Communication de ... vers ...	Type de communication	Données transmises	Risques	Sensibilité	Protocole utilisé
Client vers serveur	HTTP	Données web	Interception mot de passe / item	Très sensible	HTTP
Serveur vers BDD	SQL	Données SQL	Interception mot de passe / item	Très sensible	MySQL

B. Analyse des vulnérabilités

La communication se passe sans chiffrement donc les mots de passe se retrouve en clair et peuvent être intercepté lors de la communication. Il pourrait y avoir un problème pour intercepter les requêtes du côté serveur si le mot de passe protégeant la base.

De même, le serveur ne limite en rien les requêtes que l'utilisateur a le droit de faire ou non, impliquant qu'on peut ouvrir n'importe quel fichier sur le serveur. Cela n'est, pour le moment, pas critique car nous ne stockons rien de dangereux, mais si jamais nous venions à stocker des fichiers plus sensibles (base de données sous fichier db par exemple), l'utilisateur pourrait y avoir accès sans même être connecté en trouvant simplement le bon lien.

The screenshot shows a web browser window with three tabs: 'TaskStep', 'TD audit - TD Audit.pdf', and 'localhost/taskstep/rss.php'. The active tab displays an XML document with the following content:

```
<rss version="2.0">
  <channel>
    <title>TaskStep</title>
    <link>http://localhost/taskstep/</link>
    <description>TaskStep Items Feed</description>
    <language>en-us</language>
    <generator>IceMelon RSS Feeder</generator>
    <item>
      <title>Sample task</title>
      <link>http://localhost/taskstep/edit.php?id=1</link>
      <description>2007-08-27 | SampleProject | SampleContext | Notes</description>
    </item>
  </channel>
</rss>
```

The browser interface includes standard navigation buttons (back, forward, search) and a status bar indicating the URL 'localhost/taskstep/rss.php'.

C. Recommandations

Passer les communications entre le client et le serveur par https pourrait donc éviter les différentes failles, niveau de priorité critique.

Limiter l'accès aux fichiers pour l'utilisateur à l'aide de fichier .htaccess. Faille pour le moment peu critique mais pouvant le devenir selon ce qu'on stocke sur le serveur et étant facilement réparable.

D. Optimisation

Par manque de temps nous n'avons pas pu remédier à ce problème, mais il aurait fallu utiliser le principe de chiffrement qu'offre le https par le chiffrement SSL ce qui aurait en cas d'interception des requêtes par exemple sur un serveur public. Actuellement l'utilisation d'un curl permet de lire le contenu du mot de passe.

XIV. Analyse de la gestion de permissions

A. Analyse de l'identification

La connexion est faite par un mot de passe en simple facteur. Le mot de passe n'a pas de règle détaillée (ni de taille imposée, de symboles ou majuscule, de durée de vie, etc.), et rien n'est mis en place dans le cas où l'utilisateur oublierait son mot de passe avant de se connecter.

Au moment de la connexion, on peut essayer autant de mots de passe que l'on veut, ce qui rend donc le site vulnérable aux brutes forces.

Exemple de brute force : on utilisera le logiciel Hydra via invite de commande sur WSL a été utilisé pour faire les tests liés aux brutes force avec une wordlist contenant le mot de passe pour permettre la connexion via le mot de passe :

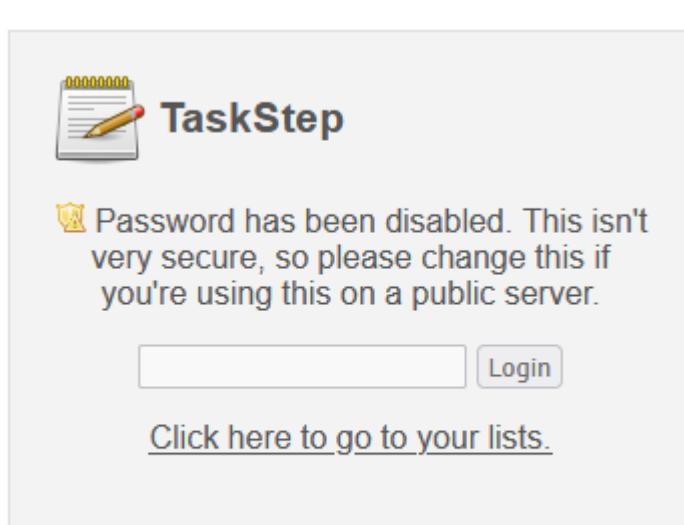
```
jules@DESKTOP-IIISQK5U:/mnt/c/Users/myste/Downloads$ hydra -l "" -P wordlist_fr_5d.txt -f 192.168.56.1 http-post-form "/taskstep-master/Login.php;password^=PASS^&submit=Login:Incorrect password."
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-04-16 09:26:58
[DATA] max 16 tasks per 1 server, overall 16 tasks, 11875 login tries (l:1/p:11875), ~743 tries per task
[DATA] attacking http-post-form://192.168.56.1:80/taskstep-master/login.php;password^=PASS^&submit=Login:Incorrect password.
[80][http-post-form] host: 192.168.56.1 password: taskstep
[STATUS] attack finished for 192.168.56.1 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-04-16 09:27:17
```

Le mot de passe "taskstep" a été placé dans les millièmes lignes et le mot de passe a donc fini par être cracké de cette façon.

Le mot de passe est stocké haché et salé en base de données.

De même, il existe un mode de fonctionnement du site qui ne nécessite aucun mot de passe, ce qui, dans le cadre d'une application multi-utilisateurs, est très dangereux.



Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hooge

B. Analyse de l'authentification

L'utilisateur ne peut pas modifier ses propres droits dans l'application, uniquement son mot de passe.

L'application ne génère rien pour gérer l'authentification de l'utilisateur (ni token, ni session créée pour limiter dans le temps la connexion).

Il n'y a aucune expiration à la connexion de l'utilisateur : une fois connecté, on l'est pour toujours (voir figures ci-dessous pour preuves dans le code)

```
sessioncheck.php
includes > sessioncheck.php
1 <?php
2 //Allow sessions
3 session_start();
4 header("Cache-control: private");
5
6 //Include the configuration
7 include("config.php");
8
9 //Connect and select the database
10 $mysqli = new mysqli($server, $user, $password, $db);
11 if ($mysqli->connect_error) {
12 | die('Connect Error (' . $mysqli->connect_errno . ') ' . $mysqli->connect_error);
13 }
14
15 //Grab the setting for "sessions"
16 $result = $mysqli->query("SELECT value FROM settings WHERE setting='sessions'");
17 if ($result->num_rows > 0)
18 {
19     //Select the results of the query in the format (query, row, column)
20     $r = $result->fetch_row();
21
22     //If sessions are enabled...
23     if ($r[0] == '1')
24     {
25         //and there is no session for "loggedin"...
26         if(!$_SESSION['loggedin'])
27         {
28             //...send them packing to the login page
29             $host = $_SERVER['HTTP_HOST'];
30             $uri = rtrim(dirname($_SERVER['PHP_SELF']), '/\\');
31             $extra = 'login.php';
32             session_write_close();
33             header("Location: http://$host$uri/$extra");
34             exit;
35         }
36     }
37 }
38 ?>
```

```
login.php
1 if (isset($_POST['submit']))
2 {
3     $result = $mysqli->query("SELECT setting,value FROM settings WHERE setting='password' OR setting='salt'");
4     while($r=$result->fetch_assoc())
5     {
6         $setting[$r['setting']] = $r['value']; //Build a multi-dimensional array containing the returned rows
7     }
8
9     $given = $_POST["password"];
10    $secured = md5($given);
11    $total = $secured.$setting['salt'];
12    if ($total == $setting['password'])
13    {
14        //$_SESSION['loggedin'] = true;
15        $host = $_SERVER['HTTP_HOST'];
16        $uri = rtrim(dirname($_SERVER['PHP_SELF']), '/\\');
17        $extra = 'index.php';
18        session_write_close();
19        header("Location: http://$host$uri/$extra");
20        exit;
21    }
22    else
23    {
24        $failed = true;
25        $_SESSION['loggedin'] = false;
26    }
27 }
28 else if (!isset($_GET['action'])) $_SESSION['loggedin'] = false; //If "action" is set, log out
29
30 //If $_SESSION['loggedin'] == true
31
32 ?>
```

C. Recommandation

Concernant les attaques par brutes force, il faudrait mettre un compte à rebours entre chaque erreur de mot de passe pour rendre l'attaque lente et en définitive décourager les attaquants, en plus d'intégrer une limite d'un certain nombre d'essais par heure ou jour afin d'empêcher de telles attaques d'arriver. En effet actuellement le site peut facilement être attaqué par brute force via n'importe quel logiciel, et l'efficacité peut être démultipliée par l'utilisation de "wordlist" complète.

De même, la connexion de l'utilisateur ne s'arrête jamais, il serait nécessaire d'intégrer une expiration à sa connexion afin d'éviter d'éventuelles intrusion sur le site à l'aide d'une simple connexion maintenue ouverte.

Le mode sans mot de passe du site web sera également à retirer pour renforcer sa sécurité.

Enfin, le mot de passe en lui-même pourrait nécessiter des ajustements tels qu'imposer majuscule, chiffre et caractère spécial voire une durée de vie au bout de laquelle il faudrait changer de mot de passe.

Le site rencontre aussi une nouvelle faille. Il y a un cookie sur le site qui permet de stocker l'identifiant d'une session PHP qui n'est pas supprimée. En modifiant la valeur du cookie par un id de session PHP qui est stockée, la page de connexion peut donc être directement contournée et les actions du site deviennent directement accessibles.

The screenshot shows the Network tab of a browser developer tools interface. On the left, there's a sidebar with categories like Application, Storage, and Cookies. Under Cookies, several cookies are listed, including one named "PHPSESSID" with the value "vsfc7r4ctktlkqbo4n7dbud5". Below the table, there's a "Cookie Value" section with a checkbox for "Show URL-decoded" which is unchecked. The main area below the table is mostly empty.

D. Optimisation

Pour la partie de la connexion se déroulant lors de la saisie d'un ID de connexion dans le cookie à l'attribut PHPSESSID, il aurait fallu mettre en place un système de validité de la session et que chaque session soit reliée par une id et donc effective pour une seule machine. Mais par manque de temps nous n'avons pas réussi à le mettre en place.

En revanche, concernant la partie du brute force nous avons pu mettre en place différentes vérifications pour retarder ce processus un maximum mais pas totalement le bloquer, nous aurions eu besoin de davantage de temps et de moyens pour limiter ces cas.

La première chose a été de mettre en place une norme de mot de passe donc celle qu'on a utilisée est une taille de 12 caractères avec au moins une minuscule, une majuscule, un chiffre et un symbole. Ce qui fait donc passer le nombre de combinaison possible à minimum environ $1,32 \cdot 10^{22}$ possibilités.

ERREUR :

The password should have a length equal to 12 and must have numbers with specials characters.

Veuillez retourner sur la page précédente

Ceci se faisait à l'aide d'une vérification par expression régulière comme décrit sur le code suivant :

```
/**  
 * Method to verify a password  
 * @param string $password The password which must be verified  
 * @return bool If a password is valid or not  
 */  
2 references | 0 overrides  
function isValidPassword(string $password) : bool {  
    return preg_match(pattern: '/^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[^#$%^&*-]).{12,}$/', subject: $password);  
}
```

Ce qui prendrait par une attaque par brute force avec un ordinateur traditionnel milieu de gamme environ 1 millier d'années à être craqué sans utiliser un système de wordlist qui pourrait donc éliminer énormément de combinaison rapidement si cette dernière est bien faite, ce qui sera énormément ralenti par le deuxième système que nous avons mis en place.

Le deuxième système qui a été mis en place et un système de pause lors de la saisie d'un mot de passe. La pause ayant été définie sur 2 secondes à chaque tentative pour ne pas nuire au confort utilisateur. L'utilisation de cette mesure permet donc de limiter les essais à donc un essai par seconde, du fait que la réponse indiquant si le test est passé ou pas met 2 secondes à apparaître en cas d'échec de connexion.

Pour obtenir une preuve d'efficacité de cette mesure, des tests ont été réalisés avec le logiciel Hydra avec les mêmes conditions que celle pour l'audit, c'est-à-dire que le mot de passe a été glissé dans une wordlist d'environ 11 000 éléments. Afin d'obtenir des chiffres assez parlants le mot de passe validant la connexion a été placé à la toute fin de la wordlist.

```
jules@DESKTOP-II5QK5U:/mnt/c/Users/myste/Downloads$ hydra -I -l julesDutrion -P wordlist_fr_5d.txt -f -vv 192.168.56.1 http-post-form "/TaskStepV2/index.php?action=ConnexionPage:login^USER^&password^=PASS^&validate=:S=Bienvenue"  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-15 15:06:40  
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting, .../hydra.restore  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 11874 login tries (l:1/p:11874), ~743 tries per task  
[DATA] attacking http-post-form://192.168.56.1:80/TaskStepV2/index.php?action=ConnexionPage:login^USER^&password^=PASS^&validate=:S=Bienvenue
```

Comme affiché sur cette capture, le lancement de l'attaque par brute force a été lancé à 15h 06min 40s et a été fini comme l'est indiqué dans la capture d'écran à 15h 11min 12s.

```
[ATTEMPT] target 192.168.56.1 - login "julesDutrion" - pass "6364AbAc/+/-" - 11874 of 11874 [child 8] (0/0)  
[STATUS] attack finished for 192.168.56.1 (waiting for children to complete tests)  
[80][http-post-form] host: 192.168.56.1 login: julesDutrion password: 6364AbAc/+/-  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-15 15:11:12
```

Ce qui donne en définitive un temps pour les 11 874 essais de 4 min et 32 secondes.

Or en rajoutant ladite pause de deux secondes entre chaque essai, le temps est passé d'environ 4 minutes à plus de 31 minutes.

```
jules@DESKTOP-II5QK5U:/mnt/c/Users/myste/Downloads$ hydra -I -l julesDutron -P wordlist_fr_5d.txt -f -vV 192.168.56.1 http-post-form "/TaskStepV2/index.php?action=ConnexionPage:login^USER^&password^PASS^&validate=:S-Bienvenue"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

[Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-15 16:07:44]
[WARNING] Restorefile (ignored ...) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 11874 login tries (l:1/p:11874), ~743 tries per task
[DATA] attacking http-post-form://192.168.56.1:80/TaskStepV2/index.php?action=ConnexionPage:login^USER^&password^PASS^&validate=:S-Bienvenue

[STATUS] 382.39 tries/min, 11854 tries in 00:31h, 20 to do in 00:01h, 16 active
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "voyons" - 11855 of 11874 [child 12] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "voyou" - 11856 of 11874 [child 2] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "vrac" - 11857 of 11874 [child 7] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "vrai" - 11858 of 11874 [child 8] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "vue" - 11859 of 11874 [child 13] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "wagon" - 11860 of 11874 [child 14] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "western" - 11861 of 11874 [child 5] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "whisky" - 11862 of 11874 [child 15] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "yacht" - 11863 of 11874 [child 8] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "yankee" - 11864 of 11874 [child 6] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "yaourt" - 11865 of 11874 [child 3] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "yeux" - 11866 of 11874 [child 11] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "yoga" - 11867 of 11874 [child 4] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "zen" - 11868 of 11874 [child 10] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "zinc" - 11869 of 11874 [child 9] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "zingaro" - 11870 of 11874 [child 1] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "zizi" - 11871 of 11874 [child 12] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "zone" - 11872 of 11874 [child 2] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "zoo" - 11873 of 11874 [child 7] (0/0)
[ATTEMPT] target 192.168.56.1 - login "julesDutron" - pass "6364AbAc/+/-" - 11874 of 11874 [child 0] (0/0)
[STATUS] attack finished for 192.168.56.1 (waiting for children to complete tests)
[00][http-post-form] host: 192.168.56.1 login: julesDutron password: 6364AbAc/+/-
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-15 16:38:47
```

Même si les conditions ont été différentes entre les deux tests, du fait que le logiciel utilisé Hydra, a utilisé 16 instances en simultanés pour pouvoir contourner au mieux cette restriction ce qui donnait 16 tests toutes les deux secondes et non 1 seuls.

```
public function Login(string $login, string $password): void
{
    $connected = false;

    if(!isset($_SESSION["connexionAttempts"]))
        $_SESSION["connexionAttempts"] = 0;

    if (isset($_SESSION["lastConnexionAttempts"]) && ($_SESSION["lastConnexionAttempts"] + TENTATIVES_TIMEOUT <= time()))
        unset($_SESSION["lastConnexionAttempts"]);

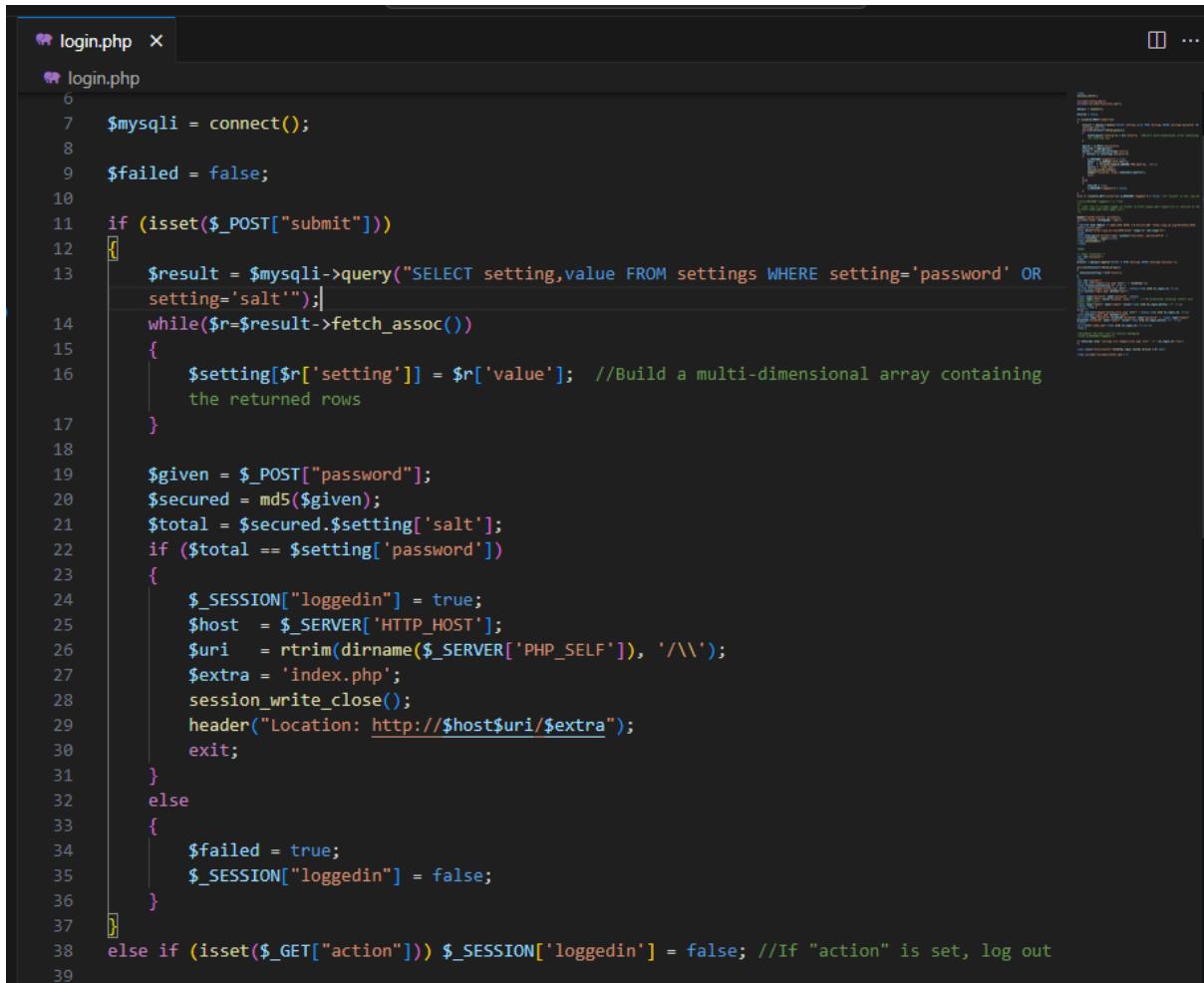
    if(!isset($_SESSION["lastConnexionAttempts"]))
    {
        if ($this->userService->Connexion(login: $login,password: $password))
        {
            $_SESSION["username"] = $login;
            $_SESSION["userID"] = $this->userService->getId(login: $login);
            $connected = true;
        }
        else
        {
            sleep(seconds: 2);
            $_SESSION["connexionAttempts"] += 1;
            if ($_SESSION["connexionAttempts"] >= TENTATIVES_CONNEXIONS)
            {
                $_SESSION["lastConnexionAttempts"] = time();
                $_SESSION["connexionAttempts"] = 0;
            }
        }
    }

    if ($connected)
        (new ControllerHome())->index();
    else
        $this->page->setMsg(msg: new Message(message: "ConnectionFailed",error: true));
        echo $this->page->GeneratePage();
}
```

XV. Analyse du stockage des informations

A. Vérification des données stockées

Une connexion est mise en place pour ce qui est de la connexion : étant donné qu'il n'existe qu'un seul mot de passe à l'application, il est récupéré en BDD et ensuite comparé au mot de passe entré, ce qui évite une possible injection SQL. Il est cependant haché en MD5, qui n'est plus un algorithme de hachage sécurisé.



The screenshot shows a code editor window with a dark theme. The file is named 'login.php'. The code is a PHP script for a login system. It starts by connecting to a MySQL database using \$mysqli = connect();. It then checks if the 'submit' button has been pressed via if (isset(\$_POST["submit"])). If true, it runs a query to select 'setting' and 'value' from the 'settings' table where 'setting' is either 'password' or 'salt'. The result is stored in \$result. A while loop iterates through the results, assigning each row's 'setting' value to \$setting[\$r['setting']] and its 'value' to \$r['value']. This builds a multi-dimensional array containing the returned rows. The script then retrieves the password input from \$_POST["password"] and hashes it using md5(\$given). It also retrieves the salt from \$setting['salt']. If the hashed password matches the stored one, session variables are set to indicate successful login (\$_SESSION["loggedin"] = true) and the user is redirected to the homepage (\$host = \$_SERVER['HTTP_HOST'], \$uri = rtrim(dirname(\$_SERVER['PHP_SELF']), '/\\'), \$extra = 'index.php', session_write_close(), header("Location: http://\$host\$uri/\$extra"), exit;). If the password does not match, \$failed is set to true and the session is marked as not logged in (\$_SESSION["loggedin"] = false). Finally, there is an else if block that logs out the user if the 'action' parameter is set in the GET request (\$_SESSION['loggedin'] = false).

```
login.php
login.php
6
7 $mysqli = connect();
8
9 $failed = false;
10
11 if (isset($_POST["submit"]))
12 {
13     $result = $mysqli->query("SELECT setting,value FROM settings WHERE setting='password' OR
14     setting='salt'");
15     while($r=$result->fetch_assoc())
16     {
17         $setting[$r['setting']] = $r['value']; //Build a multi-dimensional array containing
18         //the returned rows
19     }
20
21     $given = $_POST["password"];
22     $secured = md5($given);
23     $total = $secured.$setting['salt'];
24     if ($total == $setting['password'])
25     {
26         $_SESSION["loggedin"] = true;
27         $host = $_SERVER['HTTP_HOST'];
28         $uri = rtrim(dirname($_SERVER['PHP_SELF']), '/\\');
29         $extra = 'index.php';
30         session_write_close();
31         header("Location: http://$host$uri/$extra");
32         exit;
33     }
34     else
35     {
36         $failed = true;
37         $_SESSION["loggedin"] = false;
38     }
39
40 else if (isset($_GET["action"])) $_SESSION['loggedin'] = false; //If "action" is set, log out
```

Pour le reste des requêtes, aucune requête de sélection n'est protégée face aux injections et est donc à sujet de nombreuses failles dans la base de données.

```

$display = (isset($_GET["display"])) ? $_GET["display"] : '';
$sortby = (isset($_GET["sort"])) ? $_GET["sort"] : 'date';
$section = (isset($_GET["section"])) ? $_GET["section"] : '';
$tid = (isset($_GET["tid"])) ? $_GET["tid"] : '';

switch ($display)
{
    case "section":
        //Massively cleaned up section which obtains section titles
        //from the language file
        foreach($l_sectionlist as $key=>$value){
            if($section==$key){
                $currentsection = $key;
                $sectiontitle = $value;
            }
        }
        $result = $mysqli->query("SELECT * FROM items WHERE
        section='".$currentsection' ORDER BY $sortby");
        echo "<div id='sectiontitle'><h1>$sectiontitle</h1></div>";
        $noresultsurl = '?section=' . $section;
        break;
    case "project":
    case "context":
        $idresult = $mysqli->query("SELECT title FROM {$display}s
        WHERE id='".$tid"');
        $disptitle = $idresult->fetch_row()[0];
        $result = $mysqli->query("SELECT * FROM items WHERE
        $display='".$disptitle' ORDER BY $sortby");
        echo "<div id='sectiontitle'><h1>$disptitle</h1></div>";
        $noresultsurl = '?tid=' . $tid;
        break;
    case "all":
        $result = $mysqli->query("SELECT * FROM items ORDER BY
        $sortby");
        echo "<div id='sectiontitle'><h1>{$l_nav_allitems}</h1></
        div>";
        $noresultsurl = '';
        break;
    case "today":
        $today = date("Y-m-d");
        $todayf = date($menu_date_format);
        $result = $mysqli->query("SELECT * FROM items WHERE
        date='".$today' ORDER BY $sortby");
        echo "<div id='sectiontitle'><h1>{$l_nav_today} : $todayf</
        h1></div>";
        $noresultsurl = '';
        break;
}

```

On peut donc effectuer n'importe quelle injection SQL si on le souhaite à partir de l'URL du site :

The screenshot shows a browser window with the following details:

- Address bar: localhost/taskstep/display.php?display=all&sort=date;DROP TABLE items;
- Title bar: TD audit - TD Audit.pdf, TaskStep, TaskStep, +
- Content area:
 - Page title: TaskStep 1.1
 - Header: Today: 16th April 2025, Home, All Items, By context, By project, Settings, Help, Logout
 - Left sidebar: Add item, Ideas (0) (1), Might Want to Buy (0) (0)
 - Right sidebar: Fatal error: Uncaught mysqli_sql_exception: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'DROP TABLE items' at line 1 in C:\xampp\htdocs\taskstep\display.php:61 Stack trace: #0 C:\xampp\htdocs\taskstep\display.php(61): mysqli->query('SELECT * FROM i...') #1 {main} thrown in C:\xampp\htdocs\taskstep\display.php on line 61

MySQL protège par défaut des requêtes multiples dans un même élément mais il reste possible d'exploiter cette faille d'une façon ou d'une autre afin d'attaquer la base de données du site.

B. Chiffrement des données stockées

Données	Sensibilité	Chiffrement utilisé	Accès à la donnée	Niveau d'adéquation
Tâches	Peu sensible	Chiffré par MySQL	Sur les pages listant les tâches	Bon car nous n'avons pas besoin de plus de chiffrement de données que celui de la BDD de base
Items	Peu sensible	Chiffré par MySQL	Sur les pages listant les items	Bon car nous n'avons pas besoin de plus de chiffrement de données que celui de la BDD de base
Login	Peu sensible	Chiffré par MySQL	Par la base de données et lors de la connexion	Bon car nous n'avons pas besoin de plus de chiffrement de données que celui de la BDD de base
Mot de passe	Très sensible	MD5 + salage	/	Non car MD5 plus sécurisé
Sel	Très sensible	Chiffré par MySQL	/	Mauvais car pourrait être accessible sous injection SQL, peut être stocké à part de la BDD

C. Recommandations

Toutes les requêtes du programme seront à sécuriser si on souhaite éviter les injections SQL, cela n'est actuellement pas le cas et il est possible de trouver des informations sensibles de la base de données à l'aide de la bonne requête. En mettant le code à jour, il serait donc préférable d'échapper les paramètres et de les vérifier avant d'exécuter des requêtes SQL.

Toutefois, les failles SQL via le mot de passe sont empêchées lors de la connexion.

Le sel, servant à renforcer la sécurité derrière l'authentification, se trouve également en BDD, ce qui implique qu'une fuite de la base de données nous donnerait accès, non seulement au mot de passe haché, mais aussi à son sel et n'empêchera donc pas l'authentification. Il devrait donc se trouver dans un fichier séparé ou purement en code afin d'éviter une telle faille de sécurité.

De même, la fonction de hachage MD5 est aujourd'hui sujette à de majeures vulnérabilités et continuer de l'utiliser est un risque pour les utilisateurs de cette application. Il serait donc recommandable de modifier l'algorithme de hachage pour un plus robuste, le SHA256 étant plus actuel et optimisé dans notre cas.

D. Optimisation

Concernant la sécurité des mots de passe il a été choisi de les chiffrer en base de données et d'utiliser l'algorithme de chiffrement bcrypt qui est implanté de manière native et plus lent que SHA256, ce qui permet de nuire aux attaques par brute force en prolongeant le temps de calcul.

En plus d'avoir utilisé cet algorithme de chiffrement d'utiliser un salage généré de manière aléatoire pour garantir une plus grande sécurité. En effet le même mot de passe chiffré grâce à la méthode de hashage ne redonnera théoriquement jamais la même empreinte.

Pour parvenir à cela nous avons utilisé la méthode password_hash de PHP.

```
1 référence | 0 overrides
public function Register(string $login, string $password1, string $password2): void
{
    if ($password1 != $password2)
        throw new Exception(message: "The two passwords aren't the same.");
    if (!$this->userDAO->IsLoginAvailable(login: $login))
        throw new Exception(message: "This login is not available.");
    if (!$this->isValidPassword(password: $password1))
        throw new Exception(message: "The password should have a length equal to 12 and must have numbers with special characters.");
    $user = new User();
    $user->setLogin(login: $login);
    $user->setPassword(hashedPassword: password_hash(password: $password1, algo: PASSWORD_DEFAULT));
    $this->userDAO->CreateUser(user: $user);
}
```

Le problème ayant été aussi remonté durant la phase d'audit concernant les données stockées en base de données était les failles d'injections SQL. Ce problème a été pu être endigué grâce à l'utilisation de requête préparée ce qui veut dire que la requête de base est écrite de dur mais les paramètres sont rajoutés après par l'utilisation de chaîne de caractères ce qui empêche au SGBD d'exécuter la deuxième partie qui peut donc nuire à l'intégrité et à la sécurité des données.

```
1 référence | 0 overrides
public function DeleteUser(int $identifiant): void
{
    $request = "DELETE FROM USERS WHERE id = :id;";
    $params = [
        ":id" => $identifiant
    ];
    $this->execRequest(sql: $request, params: $params);
}

3 références | 0 overrides
public function GetUser(string $login) : ?User
{
    $user = null;
    $request = "SELECT id, password, login FROM USERS WHERE login = :login";
    $params = [
        ":login" => $login
    ];
    $response = $this->execRequest(sql: $request, params: $params);
    if (($row = $response->fetch(mode: PDO::FETCH_ASSOC)) != null)
    {
        $user = new User();
        $user->hydrate(data: $row);
    }
    return $user;
}
```

XVI. Résultats finaux

Le tableau suivant liste les tâches d'optimisation qui étaient définies par l'audit de gestion de données, la dernière colonne indiquera en vert si la tâche a pu être réalisée, en orange si elle ne l'est que partiellement, en rouge si elle n'a pas pu l'être

Recommandation	Priorité	Complexité	Effort	Gravité	Avancement
Passage Bcrypt	6	1	1	3	
Passage https	6	5	4	3	
Limite tentatives mot de passe	6	4	3	3	
Limite de temps connecté	5	3	3	2	
Gérer injection SQL	6	2	2	3	

XVII. Synthèse

L'objectif du projet était d'identifier les vulnérabilités et failles de sécurité du site web, chose qui avait été faite durant l'audit de sécurité durant lequel nous avons pu déterminer que non seulement des injections SQL étaient possibles, mais aussi que le site pouvait aisément être brut force, en plus de ne pas avoir de mots de passes sécurisés (aucune contrainte imposée par exemple).

Ainsi, nous avons non seulement mis à jour la méthode de hachage des mots de passes, celui de Taskstep étant originellement dépassé, tout en imposant un caractère majuscule, minuscule, un caractère, un chiffre ainsi qu'une taille minimum de 12 caractères.

De même, les injections SQL sont maintenant nativement gérées en PHP et ont pu être gérées durant la restructuration de ce projet, évitant donc des failles de sécurité du côté de la base de données.

Enfin, nous avons sécurisé le système de connexion à travers une variable de session limitant le nombre de tentatives pour un temps donné (après 5 tentatives échouées, on doit attendre 10 minutes par exemple) mais aussi en limitant le temps de connexion d'un utilisateur.

La seule optimisation que nous n'avons pas pu réaliser était celle du passage au HTTPS, étant donné que cela nécessite un certificat et que nous n'en avons pas les moyens. Ceci constitue une faille et dépend donc de la personne hébergeant le site web et non de nous.

Finalement, nous avons donc pu optimiser la quasi-totalité des failles de sécurité du site web, beaucoup peuvent nécessiter de bien meilleures fonctionnalités (limite par session pouvant être évitée par suppression du cookie de session par exemple) mais pour un site web ne servant qu'à lister des tâches de façon locale ou interne à une entreprise, nous estimons avoir apporté des ajustements suffisants à la sécurité du site web.

TASKSTEP

C1 – 1PEUDINSPI

- Tristan DAL MOLIN
- Moulay-Wassim ALAOUI
- Jules DUTRION
- Matteo DE MARCO
- Wassim DIOURI

BUT2 Informatique



XVIII. Analyse du projet

Les critères d'analyse suivants seront le sujet des tableaux qui suivent :

- Utilisation de clé primaire (contrainte d'intégrité)
- Utilisation clé(s) étrangère(s) si utile (contrainte d'intégrité)
- Données nullable/non nullable ((contrainte d'intégrité)
- Données uniques (contrainte d'unicité)
- Pertinence des noms des attributs
- Pertinence des types des attributs
- Cohérence des associations entre les tables

A. Table context

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/>	2 title	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
A.1 Contrainte d'intégrité				
Utilisation de clé primaire	PhpMyAdmin	Vérifier la présence de clé primaire	Conforme aux attentes	N.a.
Utilisation de clé(s) étrangère(s)	PhpMyAdmin	Vérifier la présence de clé étrangère	Clé étrangère inutile dans cette table donc conforme	N.a.
Données nullable	PhpMyAdmin	Vérifier si les champs sont nullable	id et title sont non null ce qui est cohérent	N.a
A.2 Contrainte d'unicité				
Données uniques	PhpMyAdmin	Vérifier si les données sont uniques	A amélioré car un title doit être unique, on pourrait confondre différentes taches si elles ont le même titre	5 min
A.3 Pertinent/Cohérent				
Pertinence noms attributs	PhpMyAdmin	Vérifier le nom de chaque attribut et sa pertinence	Conforme	N.a
Pertinence types attributs	PhpMyAdmin	Vérifier le type de chaque attribut et sa pertinence	Conforme	N.a
Cohérence des associations	PhpMyAdmin	Aucune association présente	N.a	N.a

B. Table projects

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/>	2 title	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

B.1 Contrainte d'intégrité				
Utilisation de clé primaire	PhpMyAdmin	Vérifier si la clé primaire est présente	Conforme aux attentes	N.a.
Utilisation de clé(s) étrangère(s)	PhpMyAdmin	Vérifier si la clé étrangère est présente	Clé étrangère inutile dans cette table donc conforme	N.a.
Données nullable	PhpMyAdmin	Vérifier si les données sont nullable a juste titre	Conforme, les deux attributs sont non-nullable	N.a
B.2 Contrainte d'unicité				
Données uniques	PhpMyAdmin	Vérifier si les données sont uniques a juste titre	A amélioré car le title doit être unique	5 min
B.3 Pertinent/Cohérent				
Pertinence noms attributs	PhpMyAdmin	Vérifier le nom de chaque attribut et sa pertinence	Conforme	N.a
Pertinence types attributs	PhpMyAdmin	Vérifier le type de chaque attribut et sa pertinence	Conforme	N.a
Cohérence des associations	PhpMyAdmin	Vérifier la cohérence des associations	Conforme	N.a

C. Table sections

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/>	2 title	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	3 fancytitle	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

C.1 Contrainte d'intégrité				
Utilisation de clé primaire	PhpMyAdmin	Vérifier si la clé primaire est présente	Conforme aux attentes	N.a.
Utilisation de clé(s) étrangère(s)	PhpMyAdmin	Vérifier si la clé étrangère est présente	Clé étrangère inutile dans cette table donc conforme	N.a
Données nullable	PhpMyAdmin	Vérifier si les données sont nullable a juste titre	Conforme, tout est non null, ce qui est cohérent	N.a
C.2 Contrainte d'unicité				
Données uniques	PhpMyAdmin	Vérifier si les données sont uniques a juste titre	A amélioré car le title doit être unique Moyen	5 min
C.3 Pertinent/Cohérent				
Pertinence noms attributs	PhpMyAdmin	Vérifier le nom de chaque attribut et sa pertinence	Title et fancytitle, incohérence 2 fois des titres	5 min
Pertinence types attributs	PhpMyAdmin	Vérifier le type de chaque attribut et sa pertinence	Conforme	N.a
Cohérence des associations	PhpMyAdmin	Vérifier la cohérence des associations	Aucune association présente Conforme	N.a

D. Table items

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/>	2 title	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	3 date	date			Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	4 section	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	5 notes	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	6 url	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	7 done	tinyint(4)			Non	0			Modifier Supprimer Plus
<input type="checkbox"/>	8 context	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	9 project	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

D.1 Contrainte d'intégrité				
Utilisation de clé primaire	PhpMyAdmin	Vérifier si la clé primaire est présente	Conforme aux attentes	N.a.
Utilisation de clé(s) étrangère(s)	PhpMyAdmin	Vérifier si la clé étrangère est présente	Les clés étrangères idContext et idProject devraient être présent au lieu d'utiliser uniquement le nom du context et project Non conforme	10 min
Données nullable	PhpMyAdmin	Vérifier si les données sont nullable a juste titre	id et title sont non null ce qui est cohérent Conforme	N.a
D.2 Contrainte d'unicité				
Données uniques	PhpMyAdmin	Vérifier si les données sont uniques a juste titre	A amélioré car un title doit être unique Moyen	5 min
D.3 Pertinent/Cohérent				
Pertinence noms attributs	PhpMyAdmin	Vérifier le nom de chaque attribut et sa pertinence	Conforme, mais il faut changer context et project pour stocker la clé étrangère de chaque table (idContext et idProject)	1 min
Pertinence types attributs	PhpMyAdmin	Vérifier le type de chaque attribut et sa pertinence	Il faut changer le type de context et project en int pour stocker la clé étrangère Moyen	5 min
Cohérence des associations	PhpMyAdmin	Vérifier la cohérence des associations	Plusieurs associations sont présente, un lien avec la table context et project Critique car non utilisation de clés étrangères	5 min

E. Table settings

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	🔑 int(11)			Non	Aucun(e)		AUTO_INCREMENT	Modifier Supprimer Plus
<input type="checkbox"/>	2 setting	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus
<input type="checkbox"/>	3 value	text	utf8mb4_general_ci		Non	Aucun(e)			Modifier Supprimer Plus

E.1 Contrainte d'intégrité				
Utilisation de clé primaire	PhpMyAdmin	Vérifier si la clé primaire est présente	Conforme aux attentes	N.a.
Utilisation de clé(s) étrangère(s)	PhpMyAdmin	Vérifier si la clé étrangère est présente	Conforme car inutile ici	N.a
Données nullable	PhpMyAdmin	Vérifier si les données sont nullable a juste titre	Tout doit être not null Conforme	N.a
E.2 Contrainte d'unicité				
Données uniques	PhpMyAdmin	Vérifier si les données sont uniques a juste titre	Nom des champs incohérent Critique	5 min
E.3 Pertinent/Cohérent				
Pertinence noms attributs	PhpMyAdmin	Vérifier le nom de chaque attribut et sa pertinence	Settings et value ne doivent pas être des noms colonnes, cela devrait être le nom des settings (password) Et les informations de la colonne value devraient correspondre aux nouveaux attributs (valeurDuPassword) Suppression des attributs tips, style, salt et session	20 min
Pertinence types attributs	PhpMyAdmin	Vérifier le type de chaque attribut et sa pertinence	Non pertinent, un entier et stocké dans une chaîne de caractère (tips et sessions) Il faudrait juste les supprimer car inutile	5 min
Cohérence des associations	PhpMyAdmin	Vérifier la cohérence des associations	Aucune association présente Conforme	N.a

Afin de rajouter plusieurs utilisateurs, il faut rajouter un attribut login en primary key, pour avoir la possibilité de se connecter avec différent utilisateur.

Il faudrait changer le nom de la table settings en user pour plus de clarté.

Pour dissocier chaque tâche selon l'utilisateur il faut ajouter une clé étrangère dans les tables projects, items et contexts.

Le temps estimé afin d'avoir plusieurs utilisateurs est de 20 min.

Le RGPD est respecté, aucune information personnelle n'est stockée.

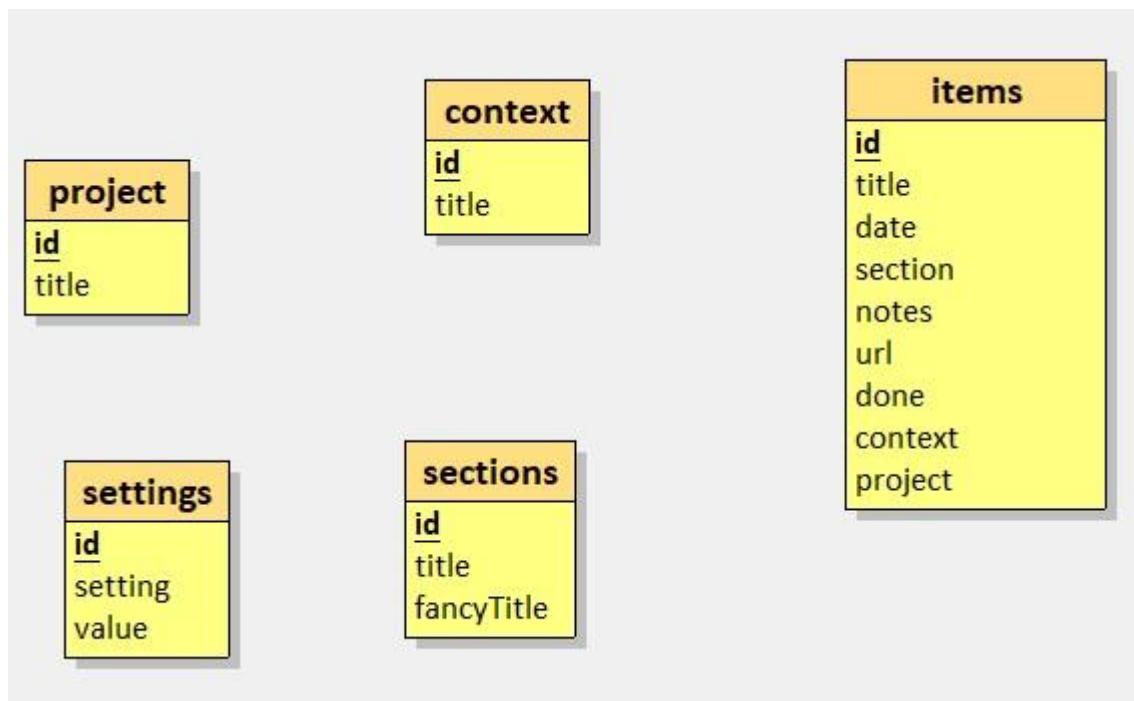
XIX. MCD recommandé et conclusion

Les normes du RGPD et les standards de gestion de données sont aux bases de l'analyse de la gestion de données du projet. Pour ce faire, une réflexion approfondie sur la gestion des données de l'application a pu être réalisée, relevant plusieurs défauts de l'application.

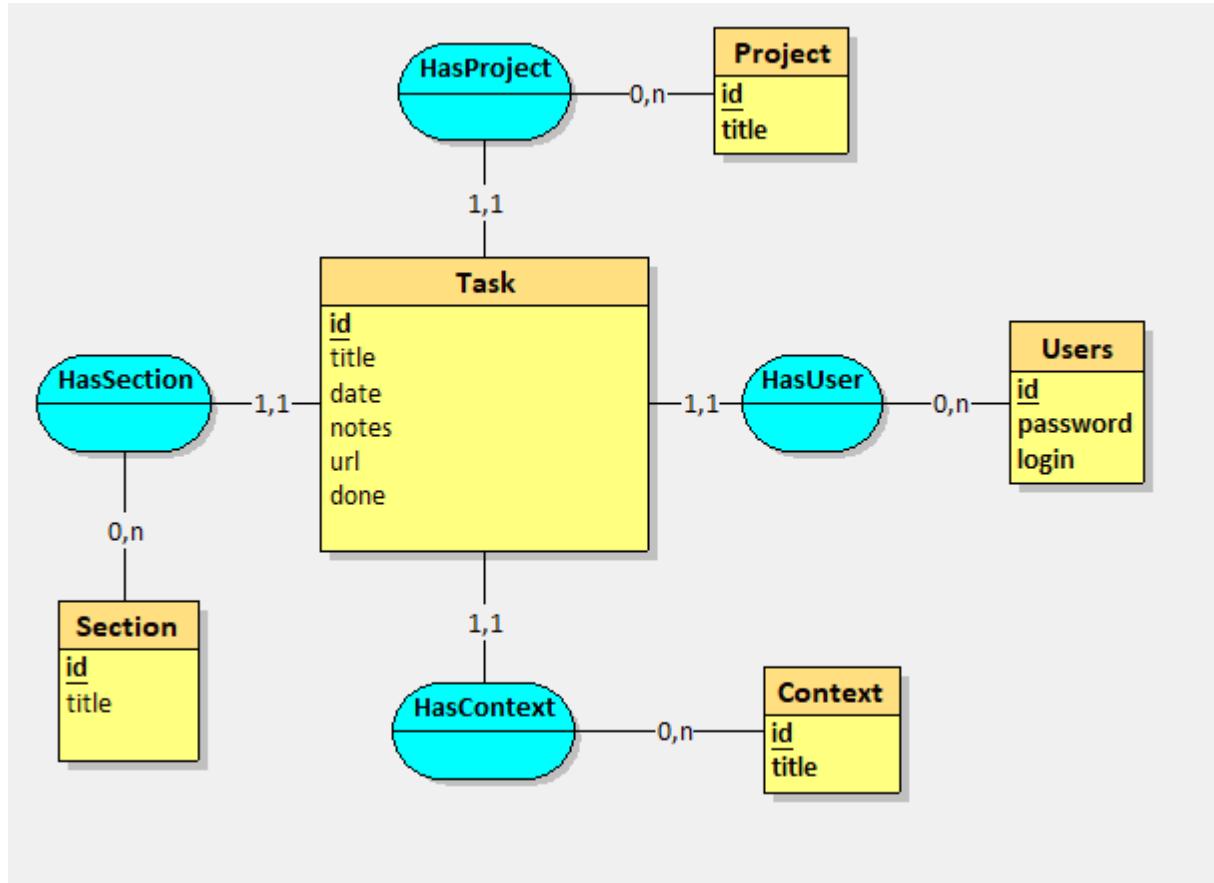
Cet audit a permis de remarquer une non-conformité de cette base de données. En effet, plusieurs attributs peuvent avoir la même valeur ce qui ne correspond pas à l'utilisation de cette base de données car chaque titre de tâche doit être unique, pour cela il faut spécifier l'attribut comme unique. De plus, aucune association via des clés étrangères n'est présente, ce lien est fait avec un attribut de l'autre table et non une clé étrangère permettant une association conforme.

La table settings stocke des informations inutiles, l'application devra être multiutilisateur, pour cela l'ajout d'attribut login est nécessaire.

Voici le MCD initial :



Ainsi, le MCD suivant serait recommandable à utiliser pour une optimisation du projet :



Des liens ont été ajoutés entre les tables task et project, sections, context permettant une meilleure liaison qu'auparavant qui était réalisé avec des attributs dans la table principale qui n'étaient pas des clés étrangères. La table settings a été renommé en user permettant une meilleure compréhension. Les défauts évoqués précédemment seraient donc résolus.

XX. Tableau des tâches réalisées

Les tâches en rouges n'ont pas été faites, en orange faites en partie et en vert totalement finies.

Recommandation	Priorité	Complexité	Effort	gravité	Optimisation
Mise en place de clé étrangère	6	1	1	3	Orange
Ajout d'association	6	2	2	3	Orange
Cohérence des noms d'attributs	6	1	1	3	Orange
Attributs uniques	6	1	1	3	Orange
Mise en place multiutilisateur	6	4	3	3	Orange

À travers un nouveau MCD, nous avons pu régler tous ces problèmes à la fois et bien optimiser l'utilisation de notre base de données.

XXI. Optimisation

Afin d'optimiser la base de données tous les conseils énoncés précédemment ont été appliqués, voici donc les changements :

- Restructuration de la base de données pour toutes les tables :

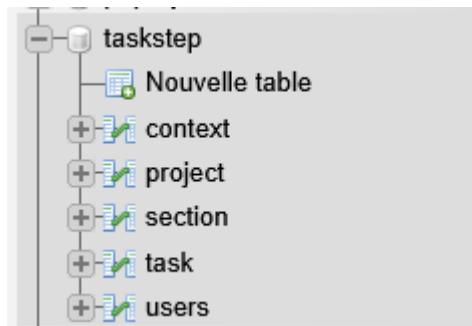


Table Context :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Plus
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			Plus

Table Project :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Plus
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			Plus

Table Section :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Plus
<input type="checkbox"/>	2 title	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			Plus

Table Task :

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Plus
2	title	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			Plus
3	date	datetime			Non	Aucun(e)			Plus
4	note	varchar(5000)	utf8mb4_general_ci		Non	Aucun(e)			Plus
5	url	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			Plus
6	done	int(11)			Non	Aucun(e)			Plus
7	idUser	int(11)			Non	Aucun(e)			Plus
8	idContext	int(11)			Non	Aucun(e)			Plus
9	idProject	int(11)			Non	Aucun(e)			Plus
10	idSection	int(11)			Non	Aucun(e)			Plus

Table Users :

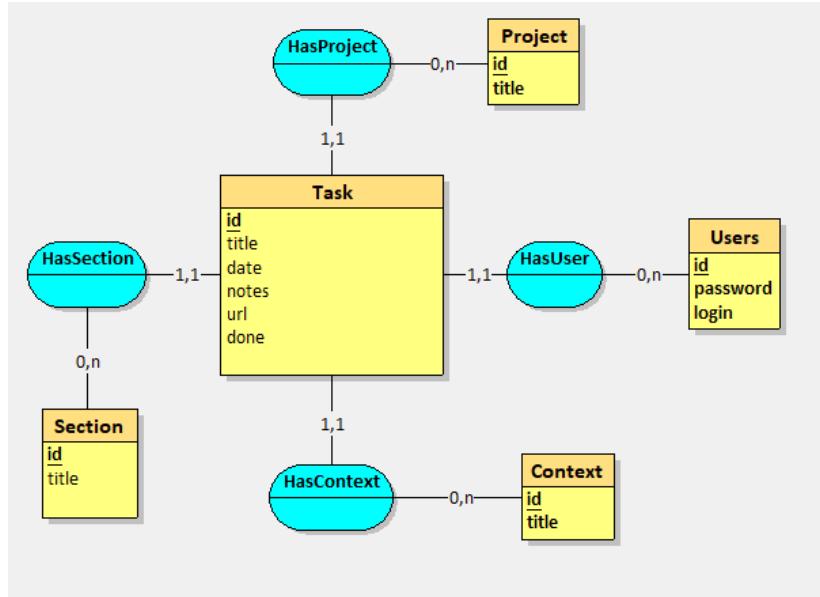
#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	id	int(11)			Non	Aucun(e)		AUTO_INCREMENT	Plus
2	password	varchar(500)	utf8mb4_general_ci		Non	Aucun(e)			Plus
3	login	varchar(255)	utf8mb4_general_ci		Non	Aucun(e)			Plus

De plus des index sont réalisés afin d'optimiser davantage les différentes requêtes :

```
CREATE INDEX idx_task_idUser ON Task(idUser);
CREATE INDEX idx_task_idContext ON Task(idContext);
CREATE INDEX idx_task_idProject ON Task(idProject);
CREATE INDEX idx_task_idSection ON Task(idSection);
```

Ces différents index ciblent chacun une clé étrangère permettant de faire le lien plus rapidement.

A cause de la contrainte des utilisateurs et pour des raisons d'optimisations des requêtes, des clefs étrangères ont été rajoutées dans la base de données. Ce qui permet en somme de pouvoir obtenir des résultats mixant les données de plusieurs tables en une seule requête. Ce qui nous a donner ce MCD qui était celui recommandé :



Comme peut l'être constaté sur cette capture, les noms n'ayant pas vraiment de signification comme l'attribut "fancyTitle" on était enlevé du fait qu'il n'était pas cohérent à la nouvelle structure et que même si l'ancienne il n'avait pas vraiment d'utilité.

La table setting s'est donc vu totalement supprimée du fait que le mot de passe a été déplacé et le display a été enlevé du stockage en base de données. Les différentes entrées de la table settings servaient à conserver les préférences liées à l'utilisateur.

La partie des settings ont été aussi enlevé du fait que le password a été déplacé dans l'entité User du fait qu'il s'agit du mot de passe de l'utilisateur et plus d'un mot de passe unique pour accéder à l'application. Ce dernier devient donc relié à un seul utilisateur.

XXII. Synthèse de l'optimisation

Comme vu lors de cette partie d'optimisation, la base de données a donc été revue pour correspondre aux nouvelles exigences de la deuxième version de Taskstep. La base de données a été revue pour pouvoir intégrer des clefs étrangères pour rendre plus efficace les recherches. Pour une meilleure compréhension d'éventuels futurs développeurs de l'application, du fait que leur nom est directement relié à leurs fonctions au sein de l'organisation des données.

Globalement, nous avons donc pu apporter des clefs étrangères à un modèle conceptuel de données qui en manquait sérieusement, les attributs et tables inutiles ont pu être retirée du projet et différents index sur des clefs étrangères ont pu être créés pour améliorer les performances. À travers ce nouveau MCD, nous avons donc su améliorer l'usage de notre base de données.

TASKSTEP

C1 – 1PEUDINSPI

- Tristan DAL MOLIN
- Moulay-Wassim ALAOUI
- Jules DUTRION
- Matteo DE MARCO
- Wassim DIOURI

BUT2 Informatique



XXIII. Introduction

Ce rapport présente l'audit d'optimisation des performances et de l'impact environnemental de l'application TaskStep. Il repose sur l'analyse de deux fonctionnalités clés, testées dans des contextes d'utilisation variés (application légère et lourde). L'audit s'appuie sur des référentiels reconnus, notamment GreenIT pour la partie environnementale, et Lighthouse pour les mesures de performance. Une comparaison entre l'application dans son état original et après optimisation sera effectuée durant cet audit afin de déterminer à quel point nous avons pu faire gagner cette application en performances.

XXIV. Évaluation de l'empreinte environnementale

Pour l'analyse qui suit, nous nous appuyons sur le référentiel GreenIT, utilisé comme checklist d'évaluation. Ce référentiel constitue également la base de mesure de l'empreinte environnementale de l'application web TaskStep.

A. Contexte

Afin d'évaluer l'empreinte environnementale de l'application web TaskStep, nous avons choisi d'analyser deux parcours utilisateurs correspondant à des fonctionnalités essentielles de l'application. Ce choix permet de représenter des cas d'usage concrets et fréquents, vécus au quotidien par les utilisateurs.

Pour garantir la pertinence des résultats, chaque scénario a été testé dans deux contextes distincts :

- Une application légère, avec très peu de tâches en base (application "quasi vierge")
- Une application conséquente, contenant un volume important de données (~3000 à 3600 tâches)

Cette approche comparative permet de juger si l'application reste performante et éco-efficiente en fonction de sa charge réelle. Elle reflète à la fois des usages standards et des cas plus extrêmes, afin de garantir un audit complet et réaliste.

B. Présentation des critères d'audit

Pour réaliser cet audit, nous nous sommes appuyés sur les critères proposés par l'extension GreenIT Analysis, outil de référence dans l'évaluation de la performance environnementale des applications web.

Cette extension se base sur un ensemble de bonnes pratiques techniques, regroupées dans un référentiel d'éco-conception. Ces pratiques couvrent différents aspects de l'optimisation numérique tels que :

- La réduction du poids des ressources (compression, minification)
- Le contrôle des requêtes HTTP (quantité, redirections, domaines)
- L'optimisation des images (format, taille, absence de redimensionnement navigateur)
- L'usage de techniques de mise en cache, de headers adaptés, ou encore de ressources externes bien gérées
- L'application de règles d'accessibilité mobile (ex. : meta viewport)

Ces critères ont servi de base pour identifier les forces de l'application (c'est-à-dire les bonnes pratiques déjà en place) ainsi que les axes d'amélioration (vulnérabilités environnementales).

C. Fonctionnalité : Ajouter une tâche

Scénario d'utilisation : Un étudiant décide d'ajouter une tâche « Rendre le rapport » avec une date limite fixée au 20 Avril.

Résultat d'analyse de la fonctionnalité :

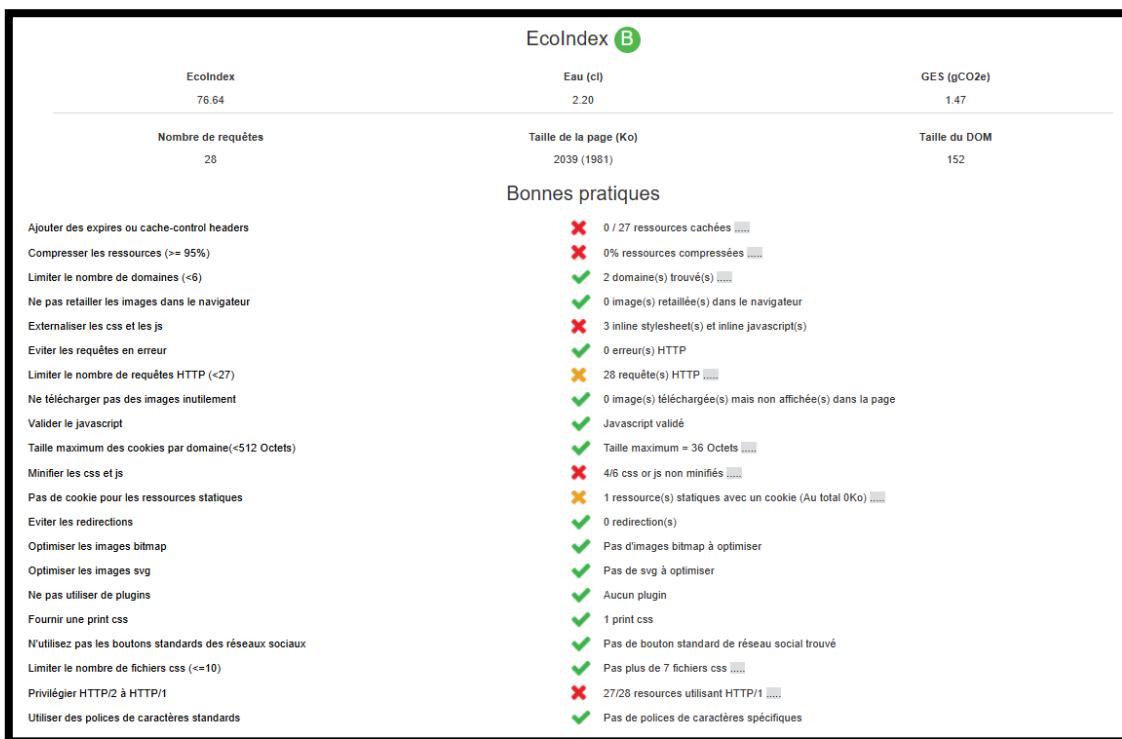


Figure 5 : Note EcolIndex Ajout d'une tâche

Dans le cadre de ce scénario d'utilisation, l'analyse de l'outil GreenIT a révélé plusieurs bonnes pratiques d'éco-conception déjà respectées :

- Aucune image redimensionnée côté navigateur
- Aucune erreur HTTP détectée
- Aucune image inutilement téléchargée

- JavaScript correctement validé
- Taille des cookies respectée (36 octets, bien en dessous du seuil de 512)
- Aucune redirection détectée
- Aucune image bitmap à optimiser
- Aucune image SVG à optimiser
- Aucun plugin tiers utilisé
- Feuille de style dédiée à l'impression présente (print.css)
- Aucun bouton standard de réseau social détecté
- Nombre de fichiers CSS limité (moins de 10)
- Polices de caractères standards utilisées
- Très bonne note ExoIndex : B

Ces bonnes pratiques montrent que, dans un contexte simple, l'application est globalement sobre et bien pensée d'un point de vue environnemental.

L'analyse a également permis de relever des vulnérabilités :

Vulnérabilité observée	Degré de risque	Solution proposée
Absence de cache HTTP (0/27 ressources cachées)	Élevé	Ajouter des expires ou headers cache-control
Aucune ressource compressée	Élevé	Compresser les ressources
Ressources non externalisées (3 styles inline)	Moyen	Externaliser les CSS et JS dans des fichiers dédiés
CSS et JS non minifiés (4/6)	Moyen	Minifier les fichiers CSS et JS
Utilisation partielle de HTTP/2 (27/28)	Moyen	Privilégier HTTP/2 pour toutes les ressources
Nombre de requêtes élevé (28)	Faible	Réduire le nombre de requêtes HTTP (<27)
Présence d'un cookie sur une ressource statique	Faible	Éviter les cookies pour les ressources statiques

Ces vulnérabilités, bien que pour certaines d'entre elles mineures, peuvent avoir un impact significatif sur la performance environnementale de l'application à grande échelle. Les plus critiques concernent l'absence de mise en cache et l'absence de compression des ressources, qui augmentent inutilement les échanges réseaux à chaque chargement. D'autres points comme la non minification des fichiers ou l'usage partiel de HTTP/2 sont relativement simples à corriger et permettraient de réduire la consommation de bande passante et d'énergie.

Scénario d'utilisation : Un étudiant décide d'ajouter une tâche « Rendre le rapport » avec une date limite fixée au 20 Avril parmi un nombre conséquent de tâches.

Résultat d'analyse de la fonctionnalité :

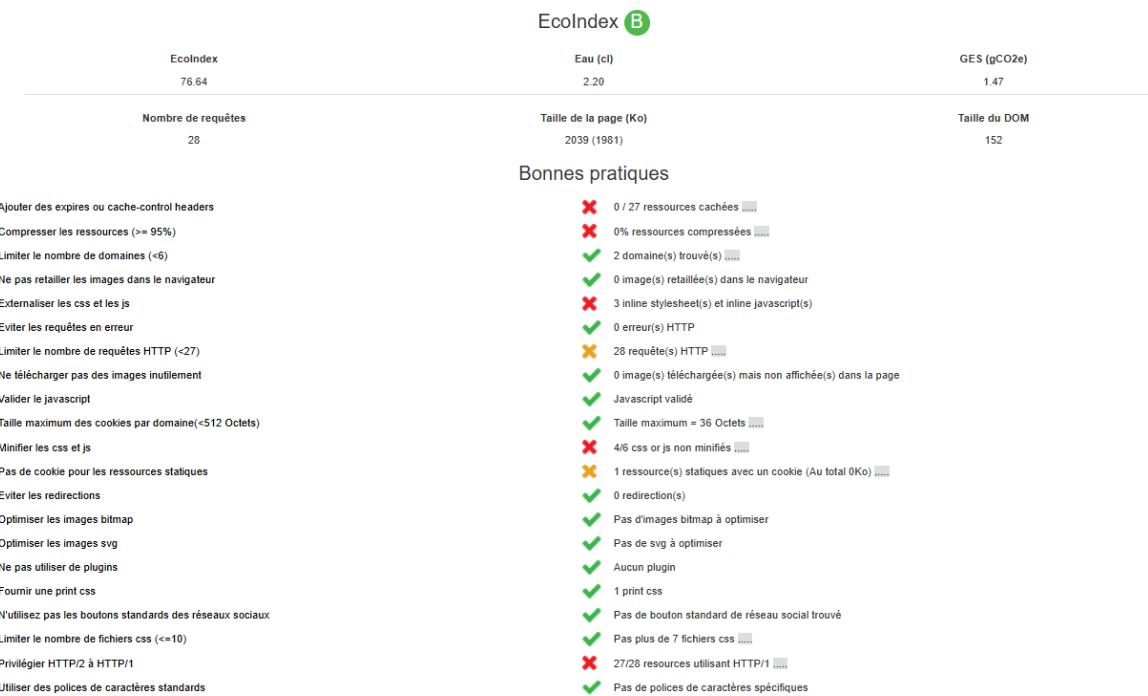


Figure 2 : Note EcoIndex ajout d'une tâche parmi de nombreuses tâches

L'audit révèle que les résultats en matière d'éco-conception sont strictement identiques entre l'ajout d'une tâche dans un environnement léger et dans un environnement lourd (plusieurs milliers de tâches en base).

Cela s'explique par le fait que la page concernée (edit.php) reste techniquement inchangée quelle que soit la volumétrie des données. Elle se contente d'afficher un formulaire simple, sans interagir avec l'ensemble des éléments stockés. Ainsi :

- Le DOM généré est identique
- Les ressources chargées sont les mêmes
- Le nombre de requêtes et les scripts exécutés ne varient pas

Cette stabilité est un bon indicateur d'éco-conception : l'application ne surcharge pas inutilement ses pages en fonction du contexte ou de la charge serveur pour cette fonctionnalité.

Toutefois, il est important de noter que les vulnérabilités détectées restent présentes dans les deux cas, ce qui indique qu'elles sont liées à la structure globale de l'application, et non à un état spécifique de la base de données.

D. Fonctionnalité : Visualisation des tâches

Scénario d'utilisation : Un étudiant veut consulter les tâches qu'il a dans la rubrique « this week » représentant ses tâches pour le week-end avec très peu de tâches présentes.

Résultat d'analyse de la fonctionnalité :



Figure 3 : Note EcoIndex de la visualisation des tâches

De la même façon que pour la tâche "Ajout d'une tâche", l'analyse de l'outil GreenIT a permis de relever, pour le scénario de visualisation des tâches dans un environnement léger, plusieurs bonnes pratiques d'éco-conception respectées :

- Limitation du nombre de domaines (2 domaines trouvés)
- Aucune image redimensionnée dans le navigateur
- Aucune erreur HTTP détectée
- Aucune image inutilement téléchargée
- JavaScript correctement validé
- Taille des cookies respectée (36 octets)
- Aucune redirection détectée
- Aucune image bitmap à optimiser
- Aucune image SVG à optimiser
- Aucun plugin tiers utilisé
- Présence d'une feuille de style dédiée à l'impression (print.css)
- Aucun bouton standard de réseau social détecté
- Moins de 10 fichiers CSS utilisés
- Polices de caractères standards employées

Cela montre que dans un environnement minimal, l'application respecte plusieurs principes essentiels de sobriété numérique.

Cependant notre analyse a quand même permis de relever des vulnérabilités :

Vulnérabilité observée	Degré de risque	Solution proposée
Aucune ressource mise en cache (0/27 ressources)	Élevé	Ajouter des headers cache-control ou expires
Aucune ressource compressée	Élevé	Activer la compression des ressources
Présence de 3 fichiers CSS/JS inline non externalisés	Moyen	Externaliser les fichiers dans des ressources dédiées
Nombre de requêtes élevé (28 requêtes)	Faible	Réduire ou regrouper les requêtes pour rester sous le seuil de 27
Fichiers CSS/JS non minifiés (3/3 non minifiés)	Moyen	Minifier les fichiers CSS et JavaScript
Cookie présent sur une ressource statique	Faible	Éviter d'envoyer des cookies sur les fichiers statiques
Utilisation partielle de HTTP/2 (27/28 ressources en HTTP/1)	Moyen	Passer l'ensemble des ressources au protocole HTTP/2

Ces vulnérabilités, bien qu'en partie mineures, révèlent des axes d'amélioration concrets pour optimiser l'impact environnemental de l'application. Leur correction permettrait de réduire significativement la consommation de ressources réseau sans altérer l'expérience utilisateur

Scénario d'utilisation : Un étudiant veut consulter les tâches qu'il a dans la rubrique « this week » représentant ses tâches pour le week-end avec déjà un nombre important de tâches présentes.

Résultat d'analyse de la fonctionnalité :

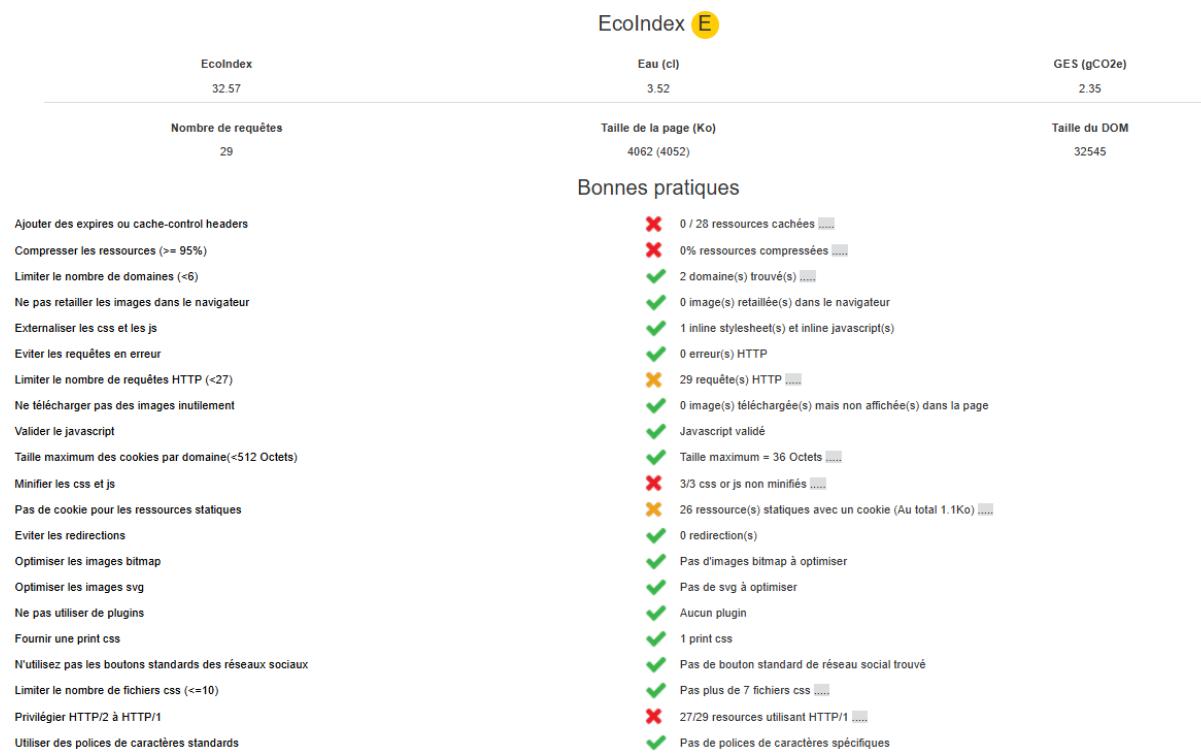


Figure 4 : NoteEcoIndex visualisation de nombreuses tâches

Bien que les bonnes pratiques techniques soient globalement respectées et identiques à celles observées dans un environnement léger, les indicateurs environnementaux se dégradent fortement lorsque la base de données contient un grand nombre de tâches (Dans notre cadre on a testé avec ~3600 tâches).

On constate notamment :

- Une taille de page extrêmement élevée (plus de 4 Mo, soit le double de quand on était en environnement léger), ce qui est problématique pour une simple liste textuelle.
- Un DOM surchargé avec plus de 32 000 éléments, ce qui entraîne une consommation excessive de ressources pour l'affichage.
- Un EcoIndex qui chute à E, traduisant une expérience particulièrement énergivore.

Ces résultats indiquent que l'architecture de l'application n'est pas adaptée à une montée en charge. L'absence de pagination, de chargement différé ou de limitation de contenu affiché provoque une explosion des ressources utilisées côté client, ce qui est contraire aux principes de sobriété numérique.

Des optimisations sont indispensables pour garantir la soutenabilité de l'application à long terme.

E. Conclusions d'audit

L'audit environnemental de l'application TaskStep met en évidence une base technique globalement stable et conforme à plusieurs bonnes pratiques d'éco-conception. Ces points positifs sont observés de manière constante, quel que soit le scénario ou la charge de données.

Cependant, des vulnérabilités récurrentes sont identifiées (absence de cache, compression désactivée, fichiers non minifiés, HTTP/2 partiel) et, surtout, une forte dégradation des indicateurs apparaît en environnement chargé : poids de la page, complexité du DOM, et chute de l'EcoIndex.

Cela souligne une absence d'adaptation à la montée en charge, en contradiction avec les principes de sobriété numérique. Des optimisations techniques ciblées sont nécessaires pour limiter l'impact environnemental de l'application à long terme.

F. Optimisation de l'application

Suite à l'audit initial, plusieurs optimisations ont été mises en place afin de pouvoir corriger les vulnérabilités identifiées et ainsi améliorer à la fois l'impact environnemental et les performances de l'application.

Afin de réaliser une comparaison très cohérente, après avoir réalisé tout le développement du site nous avons à nouveau réaliser des analyses avec l'outil GreenIT des 2 scénarios d'utilisation les plus fréquents pour l'application c'est-à-dire visualiser une tâche ainsi qu'ajouter une tâche (dans un premier temps dans un environnement léger ~3/5 tâches déjà présentes ainsi que dans un environnement lourd ~3000 tâches) afin de pouvoir juger si l'application est adaptée pour une monté de données.

En ce qui concerne l'ajout d'une tâche que ce soit dans un environnement léger ou lourd l'application de base Taskstep présentait déjà des données satisfaisantes (score EcoIndex B, 76.64 note EcoIndex) cependant le nombre de requêtes restaient très élevées par rapport à la tâche demandée. Notre implémentation a donc pu permettre d'améliorer l'impact environnemental de l'application :

EcoIndex A					
EcoIndex	90.84	Water Consumption (cl)	1.77	Greenhouse Gases Emission (gCO2e)	1.18
Request number	4	Page Size (KB)	27 (26)	Dom Size	117

Figure 6 : Ajout d'une tâche sur un environnement léger

EcoIndex A					
EcoIndex	90.84	Water Consumption (cl)	1.77	Greenhouse Gases Emission (gCO2e)	1.18
Request number	4	Page Size (KB)	27 (26)	Dom Size	117

Figure 5 : Ajout d'une tâche au sein d'un environnement lourd

En ce qui concerne la seconde tâche, c'est-à-dire visualiser des tâches que ce soit dans un environnement léger ou lourd, l'application avait un score satisfait dans un environnement léger 77.43 et un score B EcoIndex . Cependant on a pu remarquer l'application n'était pas adapté pour une montée en charge car sous un environnement lourd le score EcoIndex était à 32.57 et E montrant ainsi un impact environnemental de l'application très mauvais sur un environnement lourd. À la suite de notre implémentation nous avons nettement pu améliorer ces résultats :

EcoIndex A		
EcoIndex	Water Consumption (cl)	Greenhouse Gases Emission (gCO2e)
90.01	1.80	1.20
Request number	Page Size (KB)	Dom Size
3	28 (27)	149

Figure 6 : Visualisation d'une tâche dans un environnement léger

EcoIndex A		
EcoIndex	Water Consumption (cl)	Greenhouse Gases Emission (gCO2e)
89.23	1.82	1.22
Request number	Page Size (KB)	Dom Size
0	0	253

Figure 7 : Visualisation d'une tâche dans un environnement lourd

G. Conclusion comparative

L'évolution des scores EcoIndex obtenus après optimisation témoigne d'un changement structurel profond dans le fonctionnement de l'application. Là où TaskStep se contentait auparavant d'être « sobre » en environnement léger, elle est désormais performante et éco-responsable quel que soit le volume de données.

Avant les améliorations, l'application atteignait un niveau critique dès qu'elle devait gérer une grande quantité de tâches. L'affichage sans limitation provoquait une explosion du DOM, un poids de page supérieur à 4 Mo, et un EcoIndex de niveau E, indiquant une consommation de ressources excessive — donc peu soutenable écologiquement.

La mise en œuvre de techniques simples mais efficaces (pagination, regroupement des requêtes, réduction du contenu chargé) a permis de stabiliser l'affichage et de maîtriser les ressources consommées, même en environnement très chargé. Le passage à un EcoIndex A dans tous les scénarios testés, y compris les plus lourds, ne signifie pas seulement un bon score il permet de démontrer que la charge affichée est désormais proportionnée à l'usage réel et que l'application est adaptée à une utilisation intensive tout en restant responsable.

XXV. Mesures de performances

Afin de tester les performances de l'application de manière pertinente, nous avons choisi de suivre la même approche que pour l'évaluation de l'empreinte environnementale. Cette approche consiste à s'appuyer sur deux fonctionnalités essentielles et couramment utilisées :

- Ajout d'une tâche (dans une application vierge, puis dans une application contenant déjà un grand nombre de tâches)
- Visualisation des tâches (dans une base quasiment vide, puis avec une base très chargée)

Pour chaque scénario, nous avons mesuré des indicateurs de performance tels que le temps d'exécution et la vitesse de rendu, en utilisant l'outil Lighthouse. L'objectif est de déterminer si l'application reste fluide et performante en conditions réelles, aussi bien en environnement léger que sous forte charge.

Premier scénario : Un étudiant souhaite rajouter une tâche au sein de son application contenant pour l'instant aucune tâche pour se rappeler qu'il doit rendre son rapport pour ce week-end.

Indicateur utilisé	Valeur obtenue	Commentaire
Score de performance	100/100	Performance excellente
First Contentful Paint	0,3 s	Affichage initial très rapide
Largest Contentful Paint	0,4 s	Élément principal visible quasi instantanément
Total Blocking Time	0 ms	Excellent aucun blocage n'est détecté
Speed Index	0,3 s	Temps de chargement optimal

Deuxième scénario : L'étudiant souhaite faire la même chose que précédemment mais cette fois-ci il a une application avec un nombre important de tâches déjà présentes.

Indicateur utilisé	Valeur obtenue	Commentaire
Score de performance	100/100	Performance excellente
First Contentful Paint	0,3 s	Affichage initial très rapide
Largest Contentful Paint	0,4 s	Élément principal visible quasi instantanément
Total Blocking Time	0 ms	Excellent aucun blocage n'est détecté
Speed Index	0,3 s	Temps de chargement optimal

Troisième scénario : L'étudiant souhaite consulter les tâches qu'il a ajoutées au sein d'un environnement léger (qu'une seule tâche) :

Indicateur utilisé	Valeur obtenue	Commentaire
Score de performance	100/100	Performance excellente
First Contentful Paint	0,3 s	Affichage initial très rapide
Largest Contentful Paint	0,4 s	Élément principal visible quasi instantanément
Total Blocking Time	0 ms	Excellent aucun blocage n'est détecté
Speed Index	0,3 s	Temps de chargement optimal

Quatrième scénario : L'étudiant souhaite consulter ses tâches au sein d'une application avec un nombre important de tâches déjà présentes (~3600 tâches).

Indicateur utilisé	Valeur obtenue	Commentaire
Score de performance	59/100	Performance dégradée avec une forte charge
First Contentful Paint	2,0 s	Chargement initial très lent
Largest Contentful Paint	2,1 s	Élément principal visible avec un délai
Total Blocking Time	440 ms	Temps de blocage modéré, à optimiser
Speed Index	2,0 s	Rendu général ralenti

A. Conclusion sur les résultats obtenus

L'analyse des performances de l'application TaskStep, menée à travers quatre scénarios utilisateurs distincts, met en évidence un comportement très satisfaisant en environnement léger, mais une dégradation importante dès lors que l'application traite un grand volume de données.

Dans les trois premiers scénarios : ajout d'une tâche et visualisation de tâches dans une base quasi vide les résultats sont constants et excellents :

- Score de 100/100 sur Lighthouse
- Temps de chargement très courts (FCP à 0,3 s, LCP à 0,4 s)
- Aucun blocage détecté (Total Blocking Time : 0 ms)

Ces chiffres traduisent une application fluide, rapide et bien optimisée pour une utilisation simple.

En revanche, lors du scénario de visualisation avec une base très chargée (~3600 tâches), on observe une chute de performance significative :

- Score réduit à 59/100
- First Contentful Paint : 2,0 s, Largest Contentful Paint : 2,1 s
- Total Blocking Time : 440 ms

Ces dégradations sont principalement dues à un affichage en masse de toutes les tâches, sans découpage, sans limitation de rendu, et sans mécanisme d'adaptation. Cela entraîne un DOM très lourd et des sollicitations importantes du navigateur, ce qui dégrade l'expérience utilisateur.

En conclusion, l'application est bien dimensionnée pour des contextes légers, mais nécessite des ajustements techniques afin de maintenir ses performances dans des situations réelles à forte charge.

B. Recommandations techniques pour améliorer les performances

Suite aux résultats obtenus lors des différents scénarios, plusieurs pistes d'optimisation peuvent être envisagées afin de garantir la stabilité des performances de l'application, notamment en cas de forte charge.

Les recommandations principales sont les suivantes :

- Mettre en place un système de pagination : Plutôt que d'afficher toutes les tâches simultanément, il serait pertinent de charger les tâches page par page. Cela permettrait de limiter la quantité de données affichées et d'améliorer significativement les temps de chargement.
- Optimiser les requêtes vers la base de données : Actuellement, toutes les tâches sont probablement récupérées sans filtre. Il serait utile de revoir le fonctionnement des requêtes pour ne récupérer que les éléments nécessaires à l'instant donné (par exemple, les tâches de la semaine uniquement).
- Limiter le contenu affiché par défaut : Une autre solution complémentaire serait d'afficher par défaut un nombre restreint de tâches (ex. : 10 ou 20), avec la possibilité de charger la suite manuellement.
- Éviter le rechargement inutile de certaines ressources : Les scripts et fichiers CSS semblent rechargés à chaque interaction. La mise en place de mécanismes de mise en cache ou l'amélioration de la gestion des ressources côté client pourrait réduire la pression sur le navigateur.
- Prévoir une architecture plus modulable : Même si l'application reste légère dans sa logique, son comportement montre qu'elle n'est pas conçue pour monter en charge. Une réflexion globale sur l'optimisation du traitement et de l'affichage des données serait pertinente si l'application devait évoluer.

Ces recommandations visent à prévenir la dégradation des performances dans des situations plus proches d'un usage réel à long terme, et à garantir une expérience utilisateur fluide, quelle que soit la quantité de données stockées.

C. Résultats obtenus après optimisation

Suite à l'audit initial, plusieurs optimisations ont été mises en place afin de corriger les vulnérabilités identifiées et d'améliorer les performances générales de l'application, en particulier dans les cas de forte charge.

Afin de garantir une comparaison cohérente, nous avons effectué de nouvelles analyses avec l'outil Lighthouse après finalisation du développement. Les tests ont été réalisés sur les deux scénarios d'utilisation principaux : l'ajout d'une tâche et la visualisation des tâches, dans deux contextes distincts :

- Environnement léger : base contenant 3 à 5 tâches,
- Environnement lourd : base contenant environ 3000 tâches.

Les résultats ont confirmé que l'application présentait déjà des performances optimales pour la majorité des cas testés, avec des scores Lighthouse de 100/100, aussi bien pour l'ajout que pour la visualisation de tâches dans un environnement léger.

Cependant, en environnement lourd, nous avons constaté une chute notable des performances lors de la visualisation des tâches. Ce résultat était attendu : en l'absence de pagination, l'application générait une seule page contenant

l'intégralité des 3000 tâches, entraînant un DOM extrêmement volumineux, une charge mémoire importante, et un ralentissement sensible du rendu. Pour remédier à cette situation, nous avons mis en place :

- Un système de pagination pour répartir l'affichage des tâches sur plusieurs pages,
- Une limitation du contenu affiché par défaut,
- Une meilleure gestion du DOM et des requêtes SQL plus ciblées.

Ces mesures ont permis à l'application de maintenir sa fluidité même sous forte charge, avec des temps de chargement réduits, une interface plus réactive, et une remontée du score Lighthouse dans les scénarios complexes :

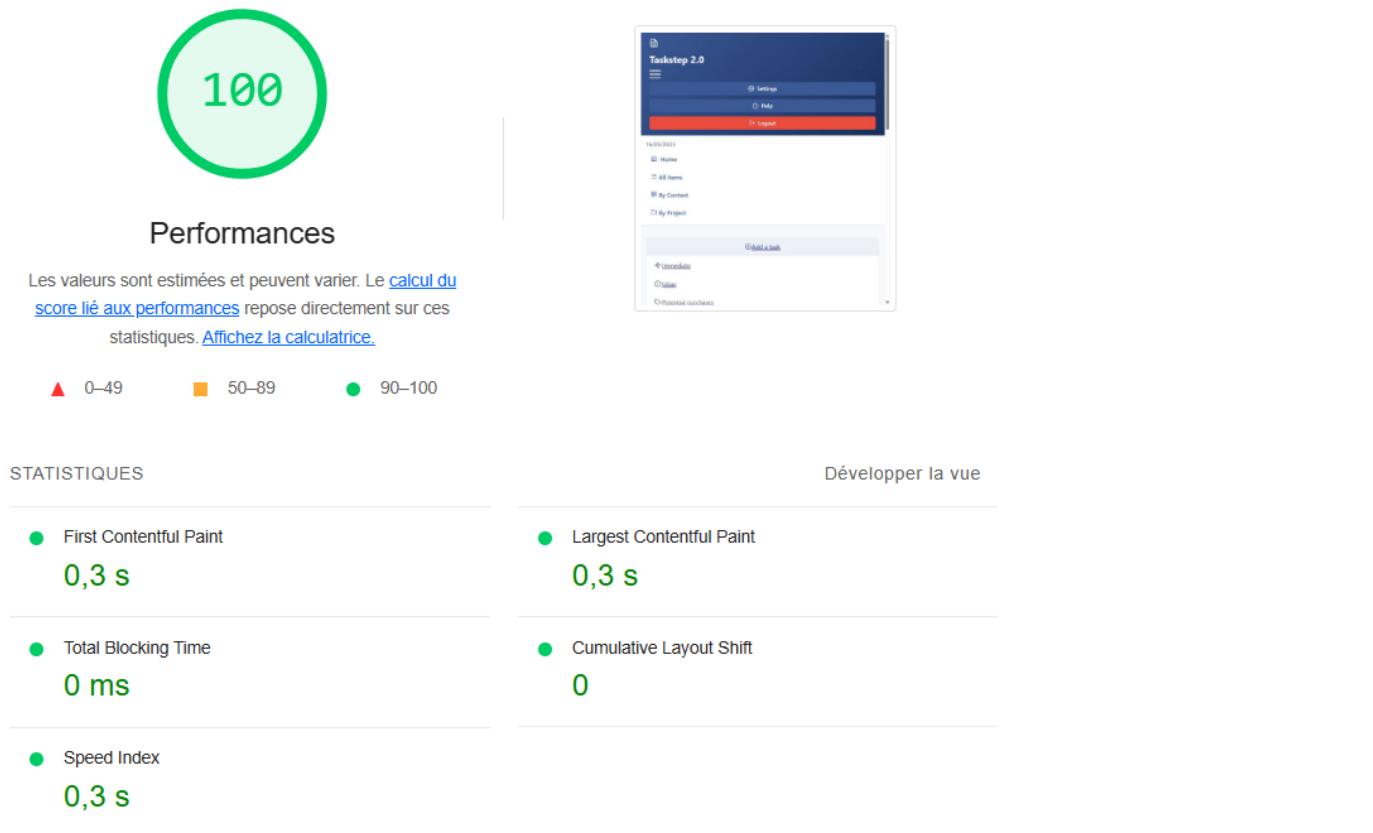


Figure 7 : Ajout d'une tâche au sein d'un environnement léger

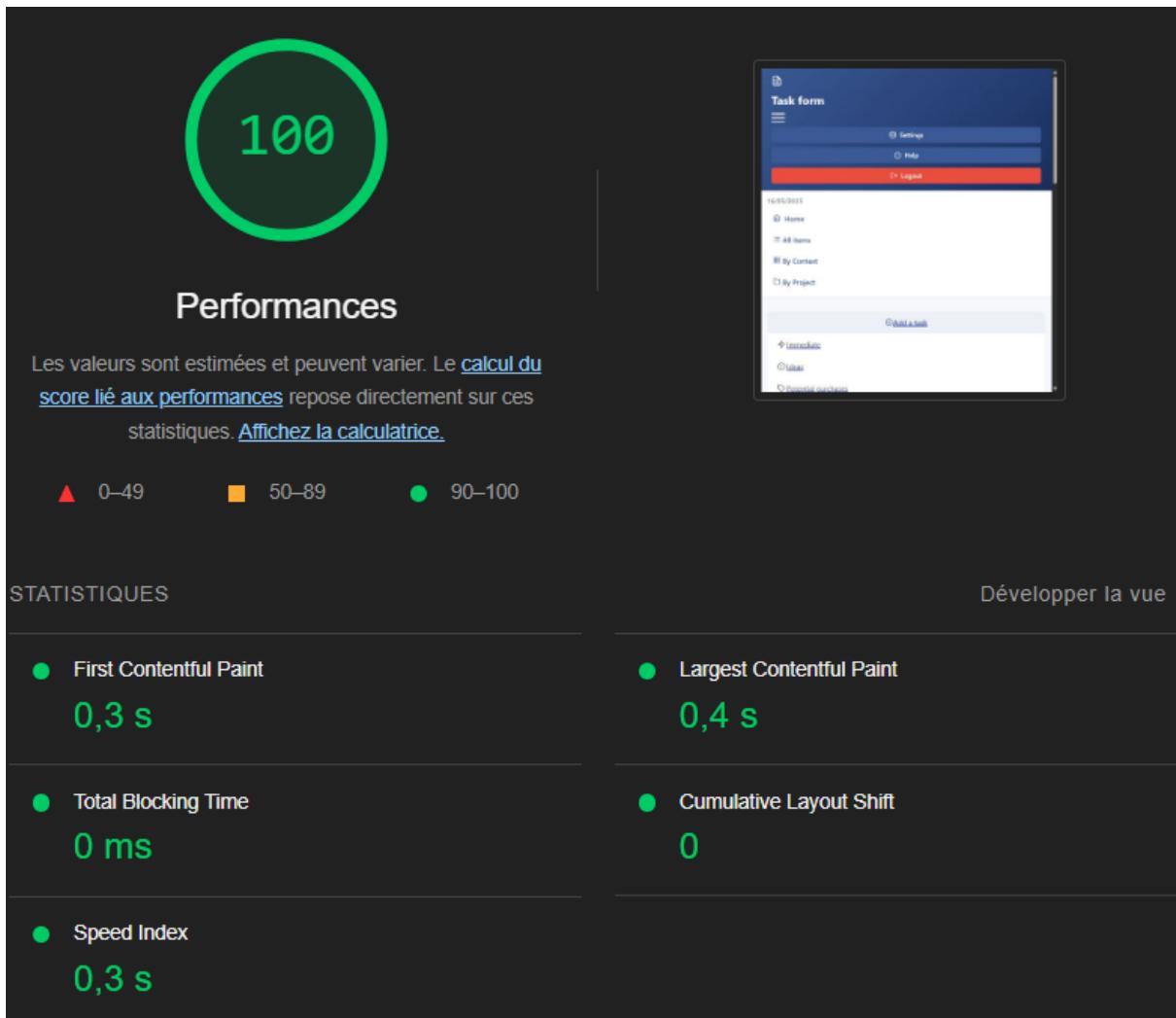


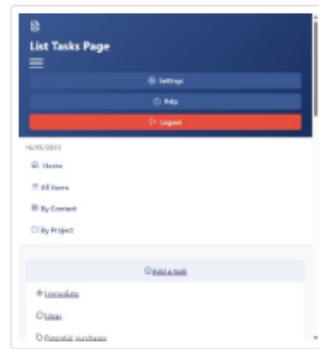
Figure 8 : Ajout d'une tâche au sein d'un environnement lourd



Performances

Les valeurs sont estimées et peuvent varier. Le [calcul du score lié aux performances](#) repose directement sur ces statistiques. [Affichez la calculatrice.](#)

▲ 0–49 ■ 50–89 ● 90–100



STATISTIQUES

● First Contentful Paint

0,3 s

● Total Blocking Time

0 ms

● Speed Index

0,3 s

Développer la vue

● Largest Contentful Paint

0,3 s

● Cumulative Layout Shift

0

Figure 9 : Visualisation d'une tâche au sein d'un environnement léger

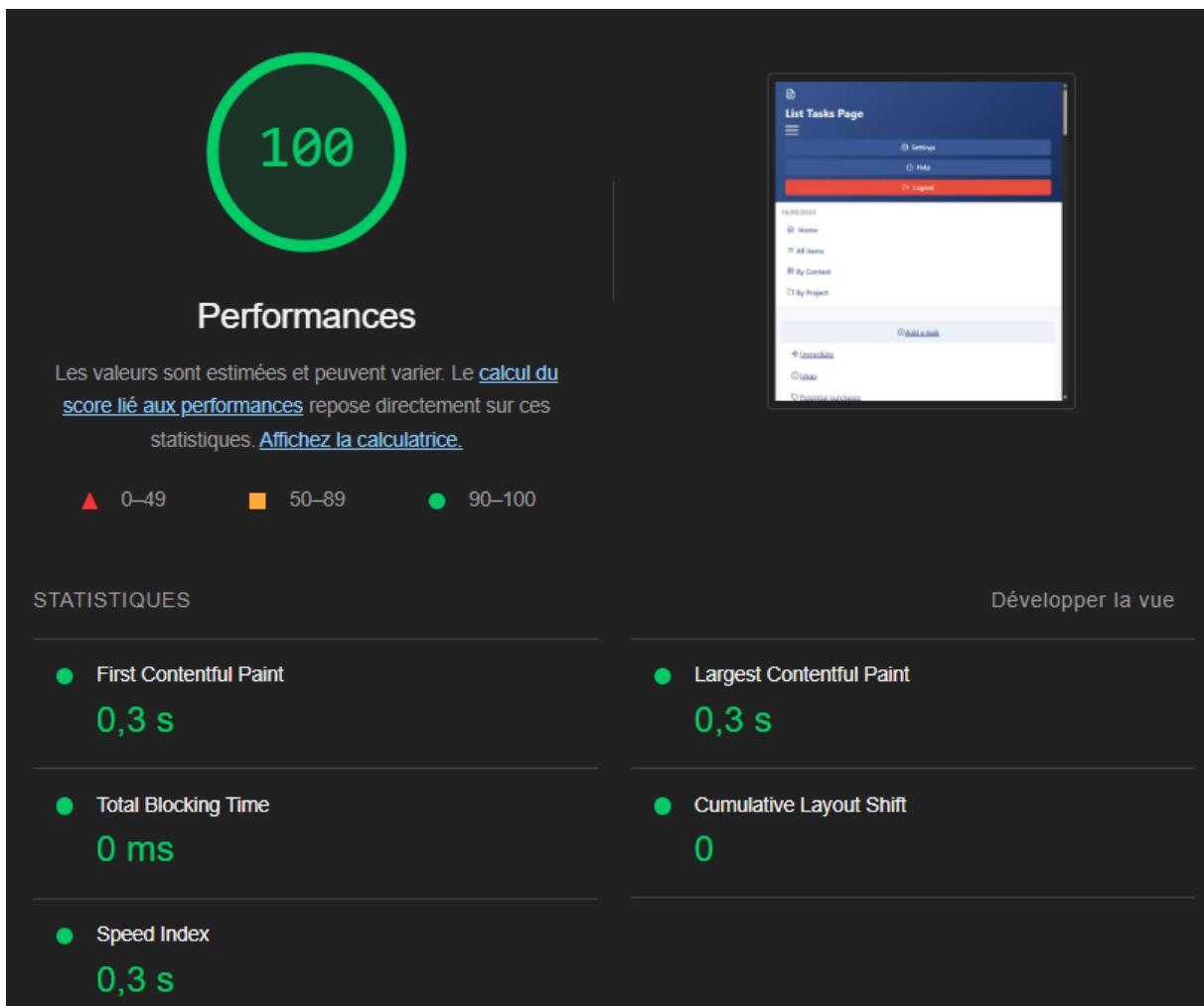


Figure 10 : Visualisation d'une tâche dans un environnement lourd

D. Conclusion comparative

L'évolution des scores Lighthouse après optimisation témoigne d'une transformation en profondeur des performances techniques de l'application TaskStep. Si elle se montrait déjà rapide et fluide en environnement léger, elle est aujourd'hui capable de maintenir un comportement optimal même sous des charges importantes.

Avant les améliorations, l'application rencontrait ses premières limites lors de la visualisation des tâches dans une base très chargée (~3000 tâches). L'absence de pagination entraînait une explosion du DOM, une charge mémoire importante, et un ralentissement notable. Le score Lighthouse chutait alors à 59/100, révélant une interface peu adaptée à un usage intensif.

Pour y remédier, nous avons mis en place des optimisations ciblées :

- Un système de pagination pour limiter le nombre de tâches affichées simultanément,
- Une réduction du contenu chargé par défaut,
- Une meilleure structuration du DOM et des requêtes SQL plus sélectives.

Ces mesures ont produit des effets immédiats : tous les scénarios, y compris les plus lourds, affichent désormais un score Lighthouse parfait de 100/100. Les captures d'écran réalisées après optimisation témoignent de temps de chargement excellents, d'un First Contentful Paint à 0,3s, et d'un Total Blocking Time nul, même avec plusieurs milliers de tâches.

Cela prouve que TaskStep est désormais parfaitement optimisée, non seulement pour une utilisation simple, mais aussi pour des contextes complexes. L'interface reste fluide, stable et réactive, quel que soit le volume de données.

XXVI. Tableaux des tâches

A. Environnement

Recommandations	Priorité	Complexité	Effort	Gravité
Éviter les cookies pour les ressources statiques	2	2	2	1
Privilégier HTTP/2 pour toutes les ressources	3	5	3	2
Réduire le nombre de requêtes HTTP (<27)	3	4	3	2
Externaliser les CSS et JS dans des fichiers dédiés	4	3	2	2
Minifier les fichiers CSS et JS	5	2	1	2
Ajouter des expires ou headers cache-control	6	2	1	3
Compresser les ressources	6	2	1	3

B. Performances

Recommandations	Priorité	Complexité	Effort	gravité
Activer la mise en cache des ressources statiques	3	2	1	2
Limiter le nombre d'éléments affichés par défaut	4	2	1	2
Optimiser les requêtes pour cibler les données utiles	5	4	3	3
Mettre en place un système de pagination	6	3	2	3
Adapter l'architecture pour supporter la montée en charge	6	6	4	3

XXVII. Synthèse

Cette analyse a été réalisée dans le respect des principes d'éco-conception numérique et en s'appuyant sur des outils reconnus tels que Lighthouse et GreenIT Analysis. Les tests ont été menés sur plusieurs scénarios représentatifs des usages utilisateurs afin d'évaluer à la fois la rapidité d'exécution de l'application et son empreinte environnementale.

Cette analyse a permis de mettre en évidence des performances satisfaisantes en environnement léger, mais aussi des dégradations notables en contexte de forte charge. L'étude environnementale, quant à elle, a révélé plusieurs bonnes pratiques déjà respectées, tout en pointant des vulnérabilités structurelles liées au chargement des ressources, à l'absence de mise en cache, ou encore à l'architecture non adaptée à la montée en charge.

Même en prenant en compte la refonte architecturale du projet et le respect actuel des principes SOLID, une séparation des couches est par défaut de mise et va donc en partie diminuer les performances du site (routing qui mène au controller appelant Service appelant DAO retournant jusqu'au controller qui injecte dans la vue par exemple). Ça n'est malheureusement pas une optimisation que nous pourrons ajuster en ce qui concerne les performances, mais une telle structure nous permettra tout de même d'améliorer les performances plus facilement sur tous les autres aspects du site web.

TASKSTEP

C1 – 1PEUDINSPI

- Tristan DAL MOLIN
- Moulay-Wassim ALAOUI
- Jules DUTRION
- Matteo DE MARCO
- Wassim DIOURI

BUT2 Informatique



XXVIII. Optimisation

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
A.1 Utilisation appropriée des balises sémantiques	On remarque qu'aucune balise sémantique n'est utilisée et qu'à la place, de simples div avec un id header/footer sont utilisées. Des divs sont utilisées partout pour la mise en forme du site. Un div à l'id container est utilisé à la place d'un <main> par exemple	Des balises sémantiques appropriées ont été intégrées sur l'ensemble des pages du site pour structurer correctement le contenu.

```
<!DOCTYPE html>
<html lang="fr"> scroll
  > <head>... </head>
... ▼ <body> == $0
  > <header class="header">... </header> flex
  > <nav class="nav ">... </nav> flex
  > <main class="container">... </main> grid
    <footer class="footer"> TaskStep 2.0 - Par Un peu d'inspi (1peuD'inspi) </footer>
  </body>
</html>
```

Figure 11 : Balise sémantiques de la page d'accueil

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
A.2 Mention de la langue principale et le cas échéant des langues secondaires	Aucune mention de la langue principale, ni d'éventuels changements de langue alors qu'elles étaient mentionnées sur le README. Les paramètres ne permettent pas de changer la langue non plus	Nous avons mis en place une page d'aide dans laquelle l'ensemble de l'application est expliqué, nous y avons aussi mentionné les langues disponibles sur l'application.

Présentation de TaskStep

TaskStep est une application gratuite et open-source de gestion de tâches qui vous aide à organiser efficacement vos activités quotidiennes et vos projets à long terme. Inspirée par la méthodologie GTD (Getting Things Done), TaskStep vous permet de structurer vos tâches selon différentes échéances et contextes.

Disponible en anglais et français.

Figure 12 : Extrait de la page d'aide avec mention des langues disponibles

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
A.3 Un titre pertinent par page	Voir screenshots ci-dessous, seule la page de login a un titre pertinent, le reste des pages ont pour nom « Taskstep » accompagné de rien d'autre, elles n'ont pas de titre réellement pertinent vis-à-vis de leur usage	Des titres explicites ont été définis pour chaque page, en cohérence avec l'action principale que l'utilisateur peut y effectuer.
A.4 Un titre différent pour chaque page	Même chose que le critère précédent, chaque page, sauf celle de login, a pour titre « Taskstep »	Des titres différents et pertinents ont été mis en place pour chaque page.

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
A.5 Un titre de niveau 1 pour chaque page	Chaque page a pour balise <h1> le logo de retour à la page d'accueil du site web. Elle n'est pas utilisée de façon pertinente ou en guise de titre et est donc à redéfinir pour des titres de pages plus pertinents (voir screenshots ci-dessous où chaque page manque d'un réel titre, certaines en ayant une en h1 aussi qui sont à conserver)	Les balise h1 sont utilisé de manière pertinente

Figure 13 : Titre de niveau 1 de la page

Figure 14 : Page affichant tous les items

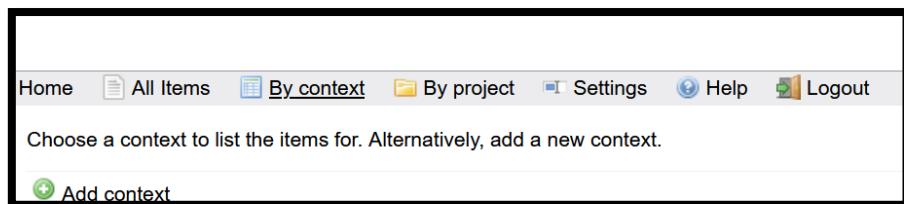


Figure 15 : Page listant les contextes

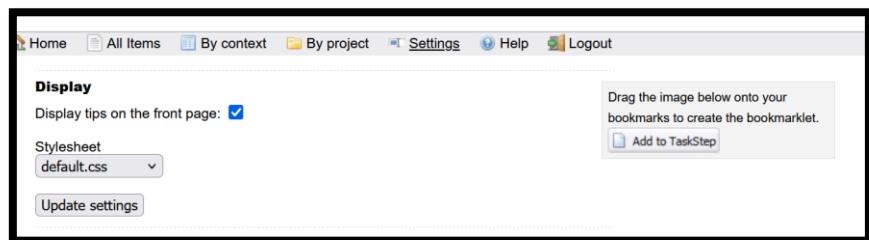


Figure 16 : Page de paramètres

The screenshot shows the "TaskStep 1.1" page. The top navigation bar includes Today: 15th April 2025, Home, All Items, By context, By project, Settings, Help, and Logout. On the left, a sidebar lists categories: Ideas (0), Might Want to Buy (0), Immediate (0), This week (0), This month (0), This year (0), and Some day maybe (0). The main form area has fields for Title (with placeholder "Task or step title"), Notes, Section (dropdown with options: Ideas, Might Want to Buy, Immediate, This week, This month, This year, Some day maybe), Context (dropdown with option: SampleContext), Project (dropdown with option: SampleProject), Due date, Url, and an "Add item" button. Below the form, a footer note says: "Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hooge".

Figure 17 : Page d'ajout d'un item

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
A.6 Respect de l'arborescence des titres	Certaines pages contiennent les titres qu'il faut en h2, en-dehors du défaut de h1 mentionné plus tôt, l'arborescence des titres semble respectée sur le site	L'arborescence est respecté dans le nouveau site

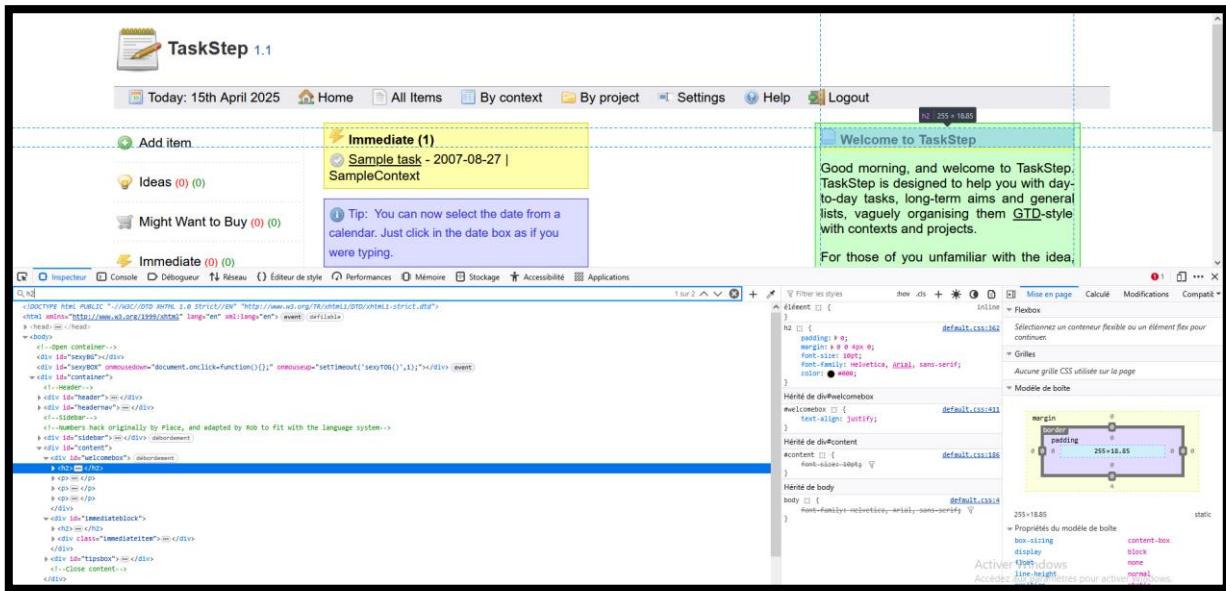


Figure 18 : Titres de niveau 2 de la page d'accueil

B.1 Navigation au clavier opérationnelle	La navigation fonctionne correctement sur le site web et tous les éléments sont accessibles au clavier sur le site web.	Cela reste de mise après l'amélioration
B.2 Retour vers la page d'accueil accessible sur chaque page	Toutes les pages ont le même header contenant non seulement l'icône renvoyant vers l'accueil mais aussi un bouton Home y renvoyant	Toujours d'actualité après les améliorations



Figure 19 : Header de l'application

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
B.3 Absence de liens vides	Un lien vers le bouton help renvoie vers un site indisponible aujourd'hui dans la barre de navigation. Il faut retirer le bouton ou ajouter la page d'aide	Le bouton help renvoi maintenant vers une nouvelle page du site
B.4 Utilisation d'attributs d'assistance aux lecteurs d'écran (title, aria-label...) pour les liens et boutons	Aucun lien n'a d'aria-label, de même pour les title	Ceci a été fait tous les formulaire possède des labels pour chaque champs
B.5 Utilisation des listes (ul/ol) pour structurer les menus	Les listes sont respectées sur la page listant les items par catégories et la barre de navigation, mais pas pour les pages listant contexte ou projets	Les liste sont bien respecté sur toute les pages

The screenshot shows a web browser displaying the TaskStep application. The main content area is titled "This month" and lists items sorted by date. The items include "Ideas (0) (0)", "Might Want to Buy (0) (0)", and "Immediate (0) (0)". A red box highlights the "Immediate" category. The bottom of the screen shows the browser's developer tools, specifically the "Inspector" tab, which displays the HTML structure of the "Immediate" section.

Figure 21 : Liste pour les types d'items

The screenshot shows a list of contexts available in the TaskStep application. The contexts listed are "Add item", "Ideas (0) (0)", "Might Want to Buy (0) (0)", and "Immediate (0) (0)". Below this, there is a section titled "Choose a context to list the items for. Alternatively, add a new context." with options like "Add context", "SampleContext", and "Edit context". The "Immediate" context is highlighted with a red box.

Figure 20 : Liste de contextes de l'application

The screenshot shows the top navigation bar of the TaskStep application. It includes links for "Home", "All Items", "By context", "By project", "Settings", "Help", and "Logout". A red box highlights the "Logout" link.

Figure 22 : Liste de barre de navigation

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
C.1 Tous les contrastes respectent au minimum le critère AA du WCAG	Plusieurs couleurs ne sont pas adaptées aux critères de contraste, il faut changer là où cela sera nécessaire (voir screenshots ci-dessous), et sur chaque fiche de style de l'application	Toutes les pages sont correctement accessible et n'ont aucun problème de contraste

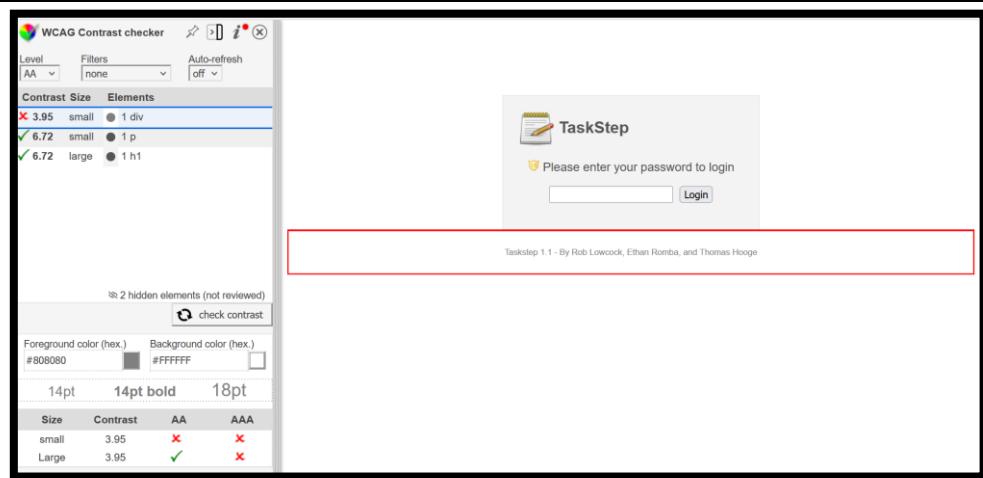
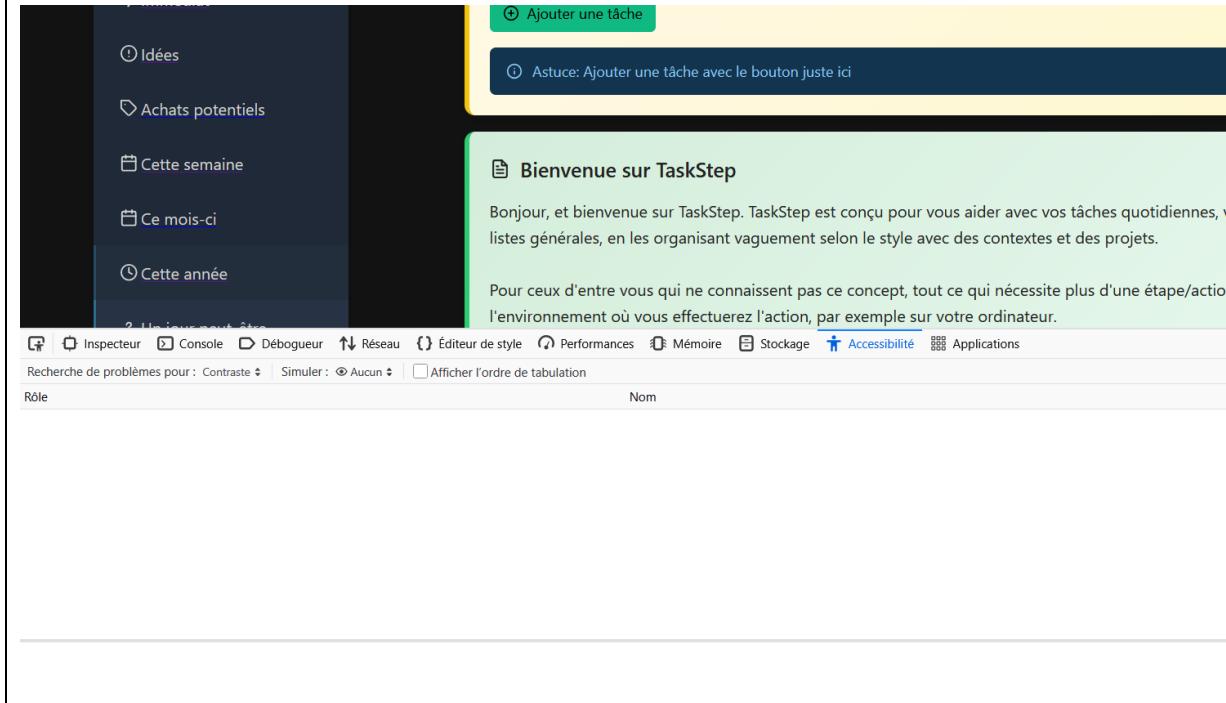


Figure 23 : Contraste de la page de connexion

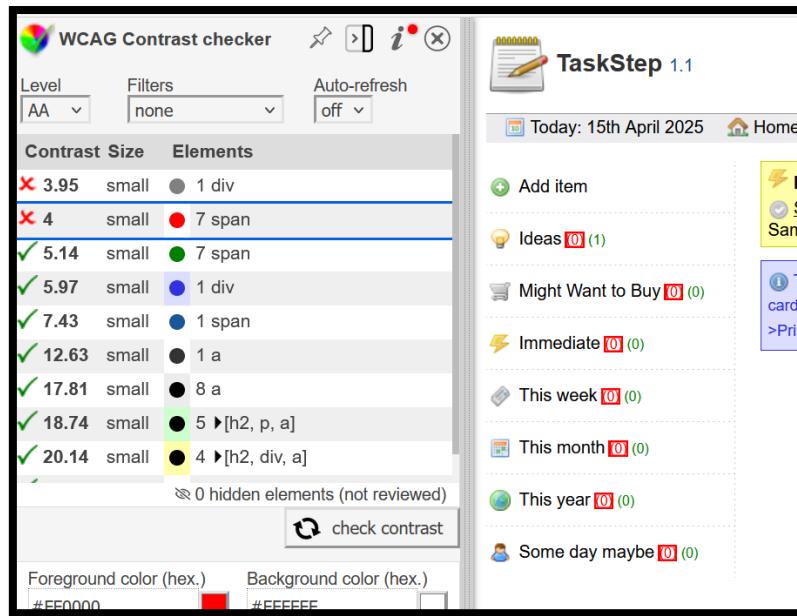


Figure 24 : Contraste de la page d'accueil

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
C.2 Aucune information ne passe uniquement par la couleur	Pour ce qui est des idées ou des tâches, celles validées sont affichées en vertes et celles qui ne le sont pas sont en rouges. L'information de la validation d'une tâche ne passe donc que par la couleur, dans la liste des tâches comme celle des items à gauche de la page	Après optimisation c'est toujours le cas il y a un texte pour dire si la tâche est faite ou non soutenu par la couleur

The figure consists of three vertically stacked screenshots from a web application:

- Top Screenshot:** A task card with a black header. It displays the title "El titro" and the following metadata: Context: El Contexto, Project: El proyecto, Date: 2005-12-01, Status: X Non Faite. Below the metadata are four buttons: View (green), Marquer comme Faite/Non Faite (blue), Modifier (blue), and Supprimer (red).
- Middle Screenshot:** A task card with a pink header. It displays the same metadata as the first card, but the status is now checked (green) and labeled "Fait". The buttons are the same: View (green), Marquer comme Faite/Non Faite (blue), Modifier (blue), and Supprimer (red).
- Bottom Screenshot:** A list titled "Ideas" with a light green header. On the left is a sidebar with a "Add item" button and a list of filters: Ideas (1), Might Want to Buy (0), Immediate (0), This week (0), This month (0), This year (0), and Some day maybe (0). The main area shows one item: "Sample task - 2007-08-27 | SampleProject | SampleContext" with a red background. To the right of the item are edit and delete icons.

Figure 25 : Information passant uniquement par la couleur

D.1 Utilisation d'une typographie de police lisible et sans-serif	3 feuilles de style peuvent être utilisées sur le site web, la première (celle par défaut) étant sans-serif, les 2 autres non (modern et professional)	Toutes les police du site sont sans-serif
---	--	---

```

# default.css X
styles > # default.css > ...
4   body{
5     font-family:Helvetica, Arial, sans-serif;
6   }

```

Figure 26 : Police de la page par défaut

```

# modern.css X
styles > # modern.css > ...
4   body{
5     font-family:Helvetica;
6     background-color: #004aa6;
7     background-image:url(images/modern/spl
8     color: #fff;
9   }

```

Figure 27 : Police de la feuille de style moderne

```

# professional.css X
styles > # professional.css > ...
4   body{
5     font-family:Helvetica;
6   }
7

```

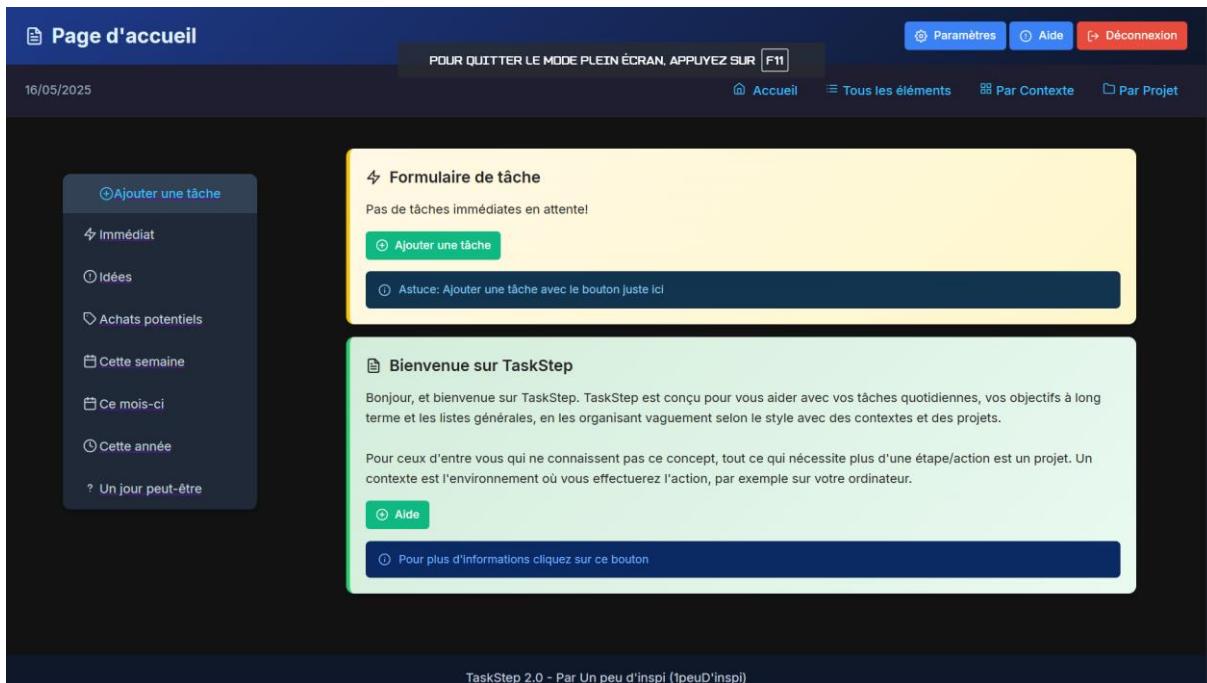
Figure 28 : Police de la feuille de style professionnelle

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
D.2 Utilisation d'unités relatives pour les polices	Beaucoup de textes sont définis en pt (unité fixe)	Toute les mesure du site sont faite en em
<pre> padding: 0.5em; font-size: 1em; width: 100%;</pre>		

```
h2{  
    padding:0;  
    margin:0 0 4px 0;  
    font-size:10pt;  
    font-family:Helvetica, Arial, sans-serif;  
    color:#000;  
}  
  
#tipsbox{  
    width:255px;  
    background-color:#ddf;  
    border:1px solid #33d;  
    padding:5px;  
    margin:10px 0 0 0;  
    color:#33d;  
    font-size:9pt;  
    line-height:1.6em;  
}
```

Figure 29 : Tailles de police en pt

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
D.3 Taille de fonte suffisante	En ayant la page à 100% de zoom, le texte est bien trop petit et illisible.	Apres optimisation quand on zoom le texte est bien visible



D.4 Paragraphes de texte ne comportant pas plus de 80 caractères par ligne	On remarque que le texte d'une tâche va jusqu'à environ 100 caractères par ligne	Nous n'avons pas eu le temps pour corriger ce point
D.5 Paragraphes de texte alignés à gauche	Quelques éléments de CSS sont justifiés et sont à remettre alignés à gauche	Apres optimisation tous les paragraphe de texte sont aligné a gauche

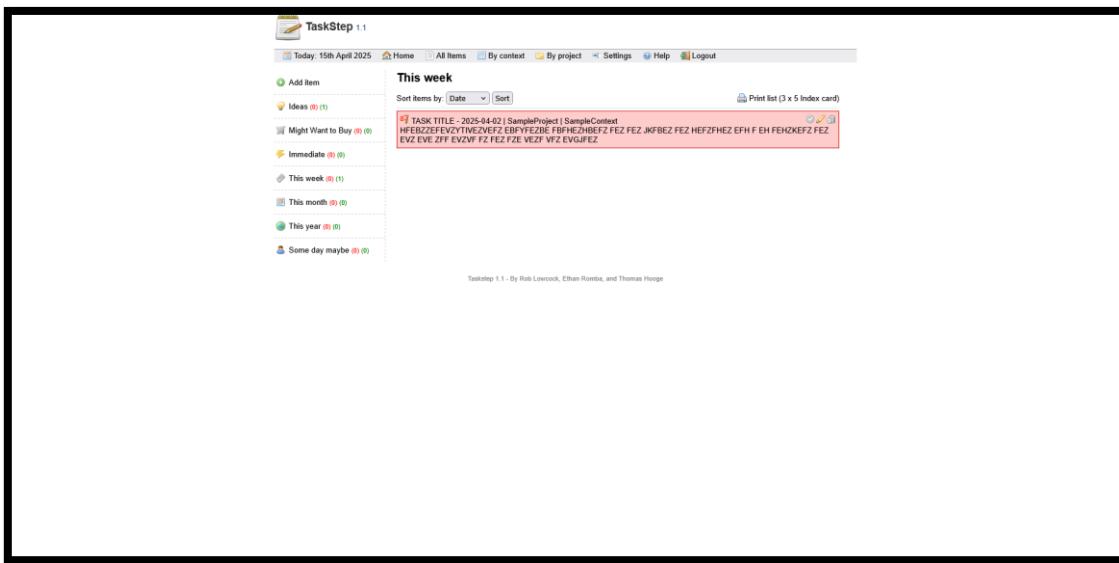


Figure 30 : Exemple de page du site web

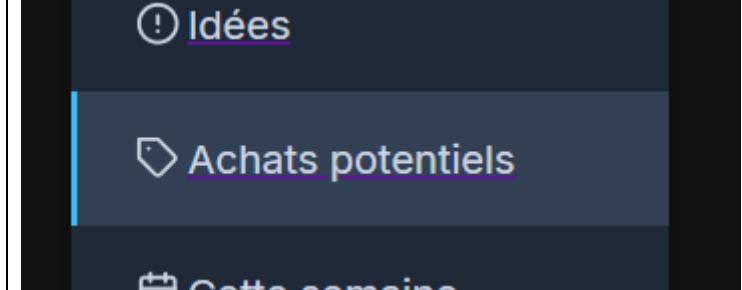
```

1  #welcomebox{
2      background-color:#cfc;
3      border:1px solid #0c0;
4      float:right;
5      width:255px;
6      padding:5px;
7      text-align:justify;
8  }
9

```

Figure 31 : Texte justifié

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
D.6 Soulignement réservé aux liens	La feuille de style professionnelle ne fait pas souligner les liens	Le soulignement est appliqué uniquement aux liens.
D.7 Mise en forme différenciée pour les liens actifs et ceux ayant un focus	Les liens ayant le focus ont un arrière-plan grisé pour les boutons de gauche et un soulignement actif pour ceux de la barre de navigation, sauf dans la feuille de style professionnelle	Tous les liens ont un hover et change de couleurs de fond si le curseur passe dessus



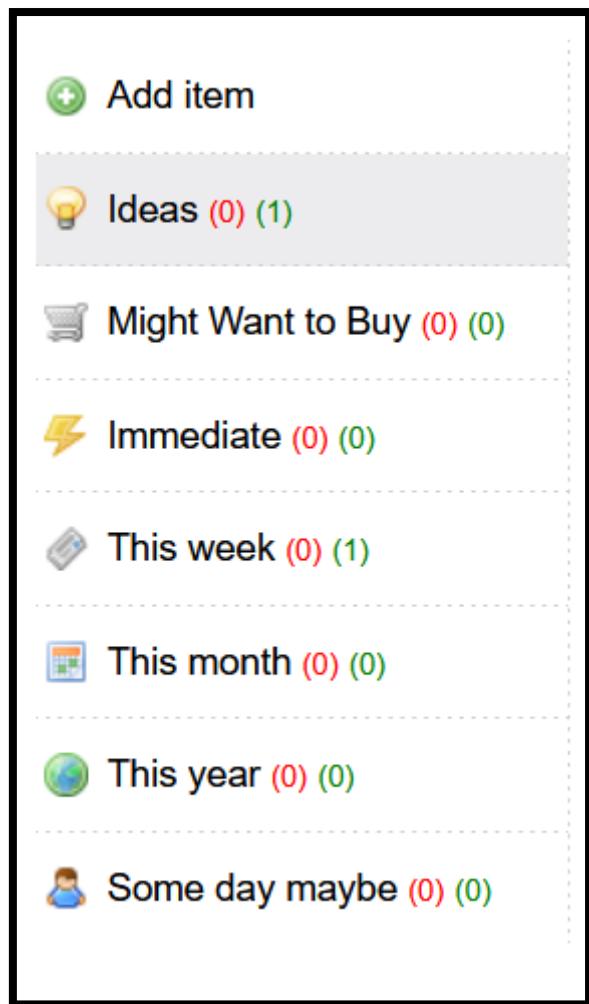


Figure 32 : Focus des liens d'items

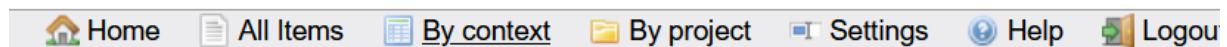


Figure 33 : Focus des liens de barre de navigation

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
E.1 Utilisation d'étiquette liée pour les champs de saisie	Aucun label n'existe dans ces formulaires, il faut remplacer le texte simple par des label	Apres optimisation tous les champs du formulaire ont des labels
	<pre> ▼ <form action="index.php?action=CreateTask" method="POST" class="base-form"> flex ► <p class="form-note">...</p> ▼ <div class="form-group"> flex == \$0 ► <label for="title">...</label> <input type="text" id="title" name="title" placeholder="Entrez un titre à la tâche" required> ... </pre>	
E.2 Pas de placeholder en guise d'étiquette	Placeholder placé en guise de label pour le titre de la tâche (figure 28)	Apres optimisation on a refait les formulaires pour y inclure des place Holder
E.3 Les étiquettes sont alignées à droite quand placées à gauche d'un champ de saisie, ou placées en haut du champ de saisie	Elles sont toutes alignées à gauche (peu importe la feuille de style)	Apres optimisation le formulaire est bien plus logique et sa disposition plus intuitive

The screenshot shows a user interface for creating a new task. At the top, there are input fields for 'Title' and 'Notes'. Below these are three dropdown menus labeled 'Section', 'Context', and 'Project', each with a list of items. Underneath the dropdowns are fields for 'Due date' and 'Url', followed by a 'Add item' button. At the bottom of the form, there is a small note: 'Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hooge'.

Figure 34 : Formulaire d'ajout d'un item

Display
Display tips on the front page:
Stylesheet

Password
Current password*:
New password:
Confirm new password:
Use passwords and sessions (Recommended):
Fields marked with a * are necessary for changes to be made..

Tools

Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hooge

Figure 35 : Formulaire de paramètres

Critères d'audit	Résultats de l'audit (constats)	Résultat d'optimisation
F.1 Attribut alt pour chaque image	Aucune image décorative n'existe sur l'application	Après optimisation le même constat est fait il n'y a pas d'image non décorative
F.2 Alt vide pour les images décoratives	Aucune image décorative n'a d'alt	Même constat
F.3 Alt renseigné et pertinent pour les images informatives et les images fonctionnelles	Aucune image non décorative n'existe sur l'application	Même constat

TaskStep 1.1

Today: 15th April 2025 Home All Items By context By project Settings Help Logout

Add item

- Ideas (0) (1)
- Might Want to Buy (0) (0)
- Immediate (0) (0)
- This week (0) (1)
- This month (0) (0)
- This year (0) (0)
- Some day maybe (0) (0)

Display

Display tips on the front page:

Stylesheet

Password

Current password*:

New password:

Confirm new password:

Use passwords and sessions (Recommended):

Fields marked with a * are necessary for changes to be made..

Tools

Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hooge

Figure 36 : Page de paramètres sans images

XXIX. Synthèse

Recommandations	Priorité	Complexité	Effort	gravité	Fait / Non fait
Pas plus de 80 caractères par ligne	2	2	2	1	Non fait
Paragraphes alignés à gauche	2	1	1	2	Fait
Ajout des langues à l'écran	2	1	1	1	Non fait
Lien help à corriger	2	1	1	2	Fait
Attributs d'assistance aux liens/boutons	2	1	1	2	Fait
Contraste	3	2	2	1	Fait
Textes en taille relatives + taille suffisante	3	2	2	2	Fait
Utiliser des listes pour les menus	3	1	1	2	Fait
Police sans serif	3	1	1	2	Fait
Liens soulignés et avec focus	3	1	1	2	Fait
Étiquettes de formulaires	3	2	1	2	Fait
Balises sémantiques	4	2	1	2	Fait
Titre h1 par page	4	1	1	2	Fait
Informations passant par la couleur	4	2	1	2	Fait

L'audit d'accessibilité a été réalisé dans le respect des normes WCAG et des lois européennes sur l'accessibilité. Des tests de contrastes WCAG, d'autres visuels ou encore des analyses du code source du projet.

Ceci a permis de déduire que le site souffrait d'un manque de balises sémantiques organisant la structure du site web, de contrastes et de couleurs mal gérées ou encore de très mauvaise gestion des liens du site web.

XXX.Synthèse d'optimisation

L'application a pu être évaluée selon les principes d'accessibilité numérique, en se basant notamment sur le RGAA (Référentiel Général d'Amélioration de l'Accessibilité). Cet audit a permis d'identifier plusieurs points d'amélioration

touchant aussi bien à la structure du code qu'à la présentation visuelle et à l'expérience utilisateur, notamment pour les personnes en situation de handicap.

De nombreuses recommandations ont été mises en œuvre pour améliorer l'accessibilité globale de l'application. Cela inclut :

- L'utilisation de balises sémantiques HTML pour structurer correctement les contenus
- La présence d'un seul titre de niveau H1 par page, garantissant une hiérarchie claire
- L'ajout d'étiquettes pour les champs de formulaire, facilitant la navigation pour les technologies d'assistance
- L'amélioration du contraste entre le texte et l'arrière-plan, ainsi que la taille relative des textes, assurant une bonne lisibilité
- Le remplacement des polices par une police sans serif, plus lisible
- L'utilisation de listes structurées pour les menus
- L'ajout de focus visibles et de soulignement sur les liens pour en assurer la distinction
- La correction du lien "help" devenu accessible et fonctionnel
- L'ajout d'attributs d'assistance (aria-label, alt, etc.) sur les boutons et les liens
- La prise en compte de l'information transmise uniquement par la couleur, remplacée ou doublée par des éléments textuels ou visuels accessibles.

Ces correctifs ont été mis en œuvre selon leur priorité et leur gravité, afin d'optimiser l'impact sur l'expérience utilisateur. Certains points à priorité moyenne ou faible restent à traiter, notamment :

- L'ajout des attributs de langue sur les balises HTML principales
- Le respect strict des 80 caractères par ligne, qui n'a pas pu être systématisé sur toutes les pages.

Ces derniers pourront être pris en compte dans une phase ultérieure du projet.

Grâce à l'application de ces recommandations, l'application est désormais nettement plus accessible, inclusive et conforme aux standards actuels. Elle offre une meilleure prise en charge des lecteurs d'écran, une navigation clavier fluide, une hiérarchie visuelle claire et une structure sémantique cohérente.

Certaines optimisations secondaires restent à prévoir, notamment l'audit des contenus dynamiques et la prise en compte de cas spécifiques complexes. Une poursuite du projet permettrait de finaliser ces aspects et d'atteindre un niveau de conformité RGAA encore plus élevé.



TASKSTEP

C1 – 1PEUDINSPI

- Tristan DAL MOLIN
- Moulay-Wassim ALAOUI
- Jules DUTRION
- Matteo DE MARCO
- Wassim DIOURI

BUT2 Informatique



Évaluation ergonomique selon les critères de Bastien et Scapin *

Consignes :

L'analyse ergonomique étudie les défauts de conception d'une interface utilisateur. Elle peut se faire avant la phase de développement, sur la base des maquettes ou prototypes, afin d'éviter des modifications de code chronophages, ou bien après le développement lors de la reprise d'une application existante.

L'analyse critique d'une interface utilisateur existante est délicate dans la mesure où elle fait souvent appelle à des considérations subjectives. Plusieurs personnes n'auront pas forcément le même avis sur l'interface d'une application.

Il est donc nécessaire de fixer un cadre et des critères qui vont donner une grille de lecture afin de tendre vers une exhaustivité et une reproductibilité.

Chacune des remarques que vous ferez dans ce document lors de l'analyse des critères doit être illustrée par une annotation sur une capture d'écran.

Sommaire

Description générale et présentation de la séquence d'écrans permettant de réaliser l'action étudiée 117

1. Le guidage 117

 1.1 Incitation 118

 1.2 Groupement/Distinction entre Items 126

 1.3 Feedback Immédiat 129

 1.4 Lisibilité 132

2. Charge de travail 134

 2.1 Brièveté 134

 2.2 Densité Informationnelle 138

3. Contrôle explicite 140

 3.1 Actions Explicites 140

 3.2 Contrôle Utilisateur 143

4. Adaptabilité 145

 4.1 Flexibilité 145

 4.2 Prise en Compte de l'Expérience de l'Utilisateur 147

5. Gestion des Erreurs 149

 5.1 Protection Contre les Erreurs 149

 5.2 Qualité des Messages d'Erreurs 153

 5.3 Correction des Erreurs 155

6. Homogénéité/Cohérence 158

7. Significance des Codes et Dénotiations 161

8. Compatibilité 163

Synthèse 167

Tableau des tâches 169

* Bastien, J.M.C., Scapin, D. (1993) *Ergonomic Criteria for the Evaluation of Human-Computer interfaces*. Institut National de recherche en informatique et en automatique, France

Audit réalisé par : l'équipe 1PeuD'inspi (Wassim DIOURI, Matteo DE MARCO, Jules DUTRION, Tristan DAL MOLIN, Moulay-Wassim ALAOUI)

Date : 15/05/2025

Identification du logiciel/site testé : Taskstep

Identification du navigateur et du système d'exploitation avec lequel le test s'est déroulé :

Testé fait avec le navigateur Opera GX le tout sur le système d'exploitation Windows 11

Scénarios utilisateurs testé :

Ajouter une nouvelle tâche

1. L'utilisateur accède à la page de connexion
2. L'utilisateur accède à l'interface principale de l'application.
3. Il clique sur le bouton "Add item" situé en haut à gauche de la page principale
4. Une interface de saisie s'ouvre : l'utilisateur entre un titre de tâche, une description (facultative), une section, un projet et un context.
5. Il clique sur "add item"
6. La tâche apparaît dans la liste des items

Optimisation de l'application :

Ce document reprend l'audit d'ergonomie de sorte qu'à chaque défaut ayant pu être trouvé à l'application, nous apporterons une correction en rouge ainsi que des précisions si besoin en ce qui concerne l'optimisation.

XXXI. Description générale et présentation de la séquence d'écrans permettant de réaliser l'action étudiée

Description :

L'objectif ici est de présenter de manière concise et globale la procédure testée. On peut y mettre des captures d'écran permettant de comprendre le parcours utilisateur nécessaire pour réaliser l'action étudiée. Les captures d'écran présentées doivent avoir en incrustations des indications graphiques permettant de savoir les actions utilisateurs qui ont permis de passer d'un écran à l'autre. Des indications additionnelles telles que le nombre de clics et le temps nécessaire pour mener à terme l'action sont de première importance.

XXXII. 1. Le guidage

Description :

- Ensemble des moyens mis en œuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec l'ordinateur.
- D'un point de vue général, il s'agit d'évaluer la façon dont le système prend en charge l'utilisateur.
- L'utilisateur doit comprendre clairement ce qu'il peut faire et comment il peut le faire. Le système doit donc le prendre en main pour le conduire efficacement.
- Un guidage trop présent peut devenir gênant lorsque la population à laquelle on s'adresse est experte avec l'outil informatique ou avec le système. En effet, certains des procédés permettant d'améliorer le guidage vont de pair avec une réduction de la vitesse d'exécution des actions.
- On ne peut cependant pas remettre en cause l'objectif du guidage. Il s'agit seulement de réfléchir en fonction de la population cible. Si elle est experte, on devra veiller à ne pas entraver ses actions et à lui permettre de les exécuter rapidement.

A. 1.1 Incitation

Note : 2 /4

Nouvelle note :4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

- Inciter l'utilisateur à effectuer des actions spécifiques en lui fournissant des indices. Par exemple, guider les entrées de données en indiquant le format adéquat et les valeurs acceptables (ex : date (jj/mm/aa) __ / __ / __). Le mieux étant de proposer des contrôles spécifiques à l'entrée désirée (ex : une entrée calendrier pour choisir une date).
- Le critère d'incitation permet de juger des moyens mis en œuvre pour faire connaître à l'utilisateur le contexte dans lequel il se trouve et les actions qu'il peut effectuer. Par exemple, dans le domaine du web, il s'agit par exemple d'indiquer à l'utilisateur où il se trouve dans le site web et quels éléments sont cliquables.

Exemples de questions à vous poser :

- Quelles sont les zones interactives ? Sont-elles bien différencierées des autres éléments non interactifs de l'interface ?
- Ce que doit faire l'utilisateur pour réaliser ce qu'il veut est-il bien compréhensible ?
- La présentation des liens suit-elle bien la hiérarchie d'importance des fonctionnalités proposées (ex : cas d'une page très dense qui propose de très nombreuses options dont beaucoup sont très peu utilisées) ?

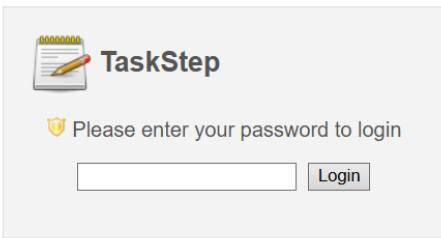
Page de connexion :

Les + :

- Le curseur se transforme lorsque on passe sur un champ où il faut écrire ou lorsqu'on passe sur un champ cliquable

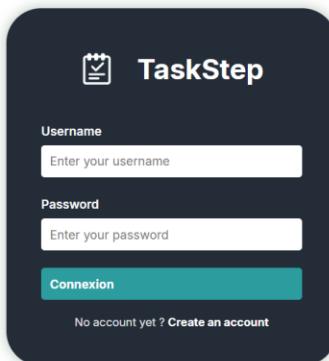
Les - :

- --- pas de remarque ---



Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hooge

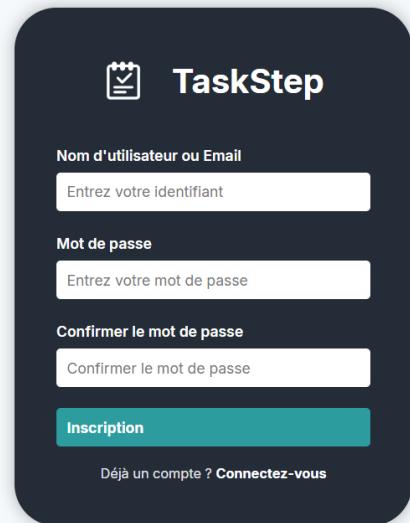
Après amélioration :



The image shows a dark-themed login form for TaskStep. At the top is a logo consisting of a clipboard icon with a checkmark and the text "TaskStep". Below the logo are two input fields: "Username" and "Password", each with a placeholder text "Enter your username" or "Enter your password". A teal-colored "Connexion" button is positioned below the password field. At the bottom of the form is a link "No account yet ? [Create an account](#)".

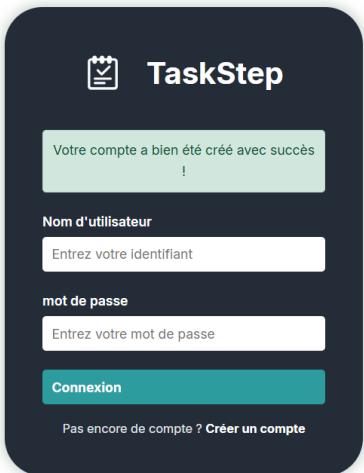
TaskStep 2.0 - By Un peu d'inspi (1peuD'inspi)

Ajout de la possibilité de créer un compte avec le texte en gras pour faire comprendre que l'on peut cliquer dessus ce qui nous ramène sur cette page :



The image shows a dark-themed registration form for TaskStep. It features three input fields: "Nom d'utilisateur ou Email" (placeholder: "Entrez votre identifiant"), "Mot de passe" (placeholder: "Entrez votre mot de passe"), and "Confirmer le mot de passe" (placeholder: "Confirmer le mot de passe"). Below these fields is a teal-colored "Inscription" button. At the bottom of the form is a link "Déjà un compte ? [Connectez-vous](#)".

A la création du compte on est automatiquement redirigé vers la page d'accueil avec un message de succès



Page d'accueil :

Les + :

- Le curseur se transforme lorsque on passe sur un champ où il est possible de cliquer

Les - :

- Lorsque l'on passe le curseur sur le bandeau du haut avec le style professionnel il n'y a pas d'effet de plus que le changement de curseur alors qu'avec les autres styles on a les boutons qui se souligne



Voici les améliorations apportées :

A screenshot of the TaskStep 2.0 homepage. The top navigation bar is dark blue with white text, featuring links for 'Paramètres', 'Aide', 'Déconnexion', 'Accueil', 'Tous les éléments', 'Par contexte', and 'Par projet'. The main content area has a light blue header with the text 'Ajouter une tâche'. Below this is a yellow section titled 'Formulaire de tâche' containing a button labeled 'Ajouter une tâche'. A tooltip 'Astuce: Ajouter une tâche avec le bouton juste ici' points to this button. The bottom of the page has a green footer with the text 'Bienvenue sur TaskStep', a message about the application's purpose, and a summary: '2 tâches restantes à faire aujourd'hui'. At the very bottom, a dark blue footer bar displays the text 'TaskStep 2.0 - Par Un peu d'inspi (tpeuD'inspi)'.

On peut par ailleurs remarquer que lorsque l'on passe la souris sur un bouton il y a bien un effet ici la souris était sur le bouton accueil

Page d'ajout :

Les + :

- --- pas de remarque ---

Les - :

- Il n'y a aucune mention si les champs sont obligatoires ou non
- Le champ titre est prérempli mais lorsque l'on écrit dedans cela n'efface pas le titre déjà écrit

A screenshot of a web application interface. At the top, there is a navigation bar with links: Home, All Items, By context, By project, Settings, Help, and Logout. Below the navigation bar is a form for creating a task or step. The form fields include:

- Title: A text input field containing "Task or step title".
- Notes: A text input field.
- Section: A dropdown menu with options: Ideas, Might Want to Buy, Immediate, This week, This month, This year, Some day maybe.
- Context: A dropdown menu set to "SampleContext".
- Project: A dropdown menu set to "SampleProject".
- Due date: A text input field.
- Url: A text input field.
- Add item: A button.

Below the form are two buttons: "Edit contexts" and "Edit projects".

- Lorsqu'on sélectionne un contexte ou un projet puis on clique sur le bouton edit cela nous renvoie vers la page qui liste tous les contextes ou projets. De plus aucun texte ne s'affiche en survolant les boutons on ne sait donc pas ce qu'ils font à part les icônes pour nous donner un indice



- Le champ url n'est pas explicite, on ne comprend pas son utilité
- Le nom des sections n'est pas explicite on ne comprend pas leur objectif et utilité
- Noms de sections peu professionnels ("some day maybe", "might want to buy")

Améliorations apportées

Ajouter une tâche

Les champs marqués d'un (*) sont nécessaires.

Titre(*)

Entrez un titre à la tâche

Notes(*)

Ajoutez quelques notes

Section(*)

(Sélectionnez une section)

Contexte(*)

title1

Projet(*)

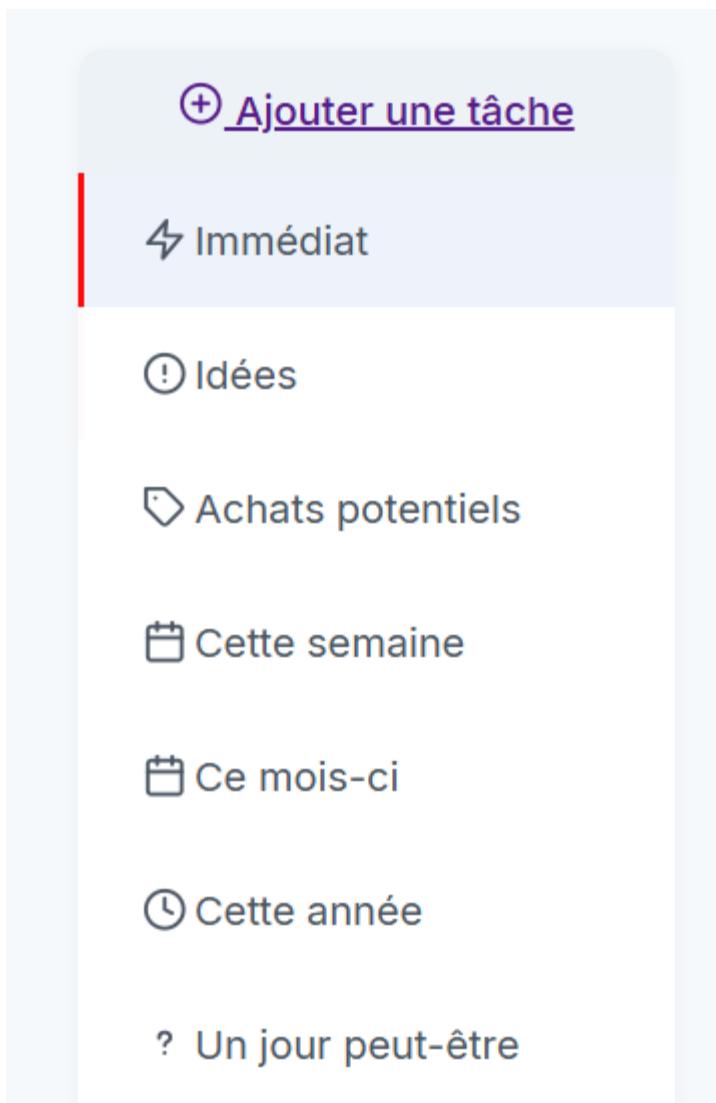
(Sélectionnez un projet)

Date limite(*)

Les champs obligatoires sont marqués, les titres des sections ont été gardés pour la plupart sauf la section might want buy qui devient Potential purchases.

Amélioration faite :

Les icônes ont été remplacées pour être plus cohérents et mieux correspondre aux sections. De plus, la barre possède désormais un effet au survol de la souris.



- [Add item](#)
- [Ideas \(0\) \(1\)](#)
- [Might Want to Buy \(0\) \(1\)](#)
- [Immediate \(0\) \(0\)](#)
- [This week \(0\) \(0\)](#)
- [This month \(0\) \(0\)](#)
- [This year \(0\) \(0\)](#)
- [Some day maybe \(0\) \(0\)](#)

B. 1.2 Groupement/Distinction entre Items

Note : 1 /4

Nouvelle note :4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

- Groupement des différents éléments visuels de façon cohérente et ordonnée.

On peut grouper les items de 2 manières :

1.3.1 Groupement/Distinction par la Localisation

Positionner les items les uns par rapport aux autres afin d'indiquer leur appartenance, ou non, à une classe donnée d'objets. Par exemple, grouper les options de menus en fonction des objets sur lesquels elles s'appliquent.

1.3.2 Groupement/Distinction par le Format

Donner aux éléments des caractéristiques graphiques particulières afin d'indiquer leur appartenance, ou non, à une classe donnée d'objets. Par exemple : utiliser un symbole et la couleur rouge pour les boîtes de dialogue d'alerte ou d'erreur.

Exemples de questions à vous poser :

- Comment (par quels éléments graphiques) les choses sont-elles rassemblées.
- L'appartenance de plusieurs items à la même catégorie fonctionnelle est-elle bien indiquée (spatialement ou graphiquement) ?
- Les items regroupés spatialement ou graphiquement appartiennent-ils bien à la même catégorie fonctionnelle ?

Les + :

- --- pas de remarque ---

Les - :

- La disposition des sections et leur arrangement n'est pas logique. Dans le bandeau à gauche, le bouton add item se trouve aux côtés de toutes les sections ce qui n'a pas de rapport direct.
- Des noms illogiques se trouvent associés aux liens de la barre de navigation : un lien "today" avec la date du jour qui renvoie aux tâches du jour, un bouton "by context" et un "by project" qui ne sont pas assez clairs sur ce qu'ils font, ... Peu de pages sont reconnaissables pour ce qu'elles font vraiment ce qui donne un groupement très hasardeux

The screenshot shows the TaskStep 1.1 application. On the left is a sidebar with a dark header 'TaskStep 1.1' and a logo. It contains links for 'Add item', 'Ideas (0) (0)', 'Might Want to Buy (0) (0)', 'Immediate (0) (0)', 'This week (0) (0)', and 'This month (0) (0)'. The main area has a yellow box titled 'Immediate (0)' with the message 'No immediate items left to-do! [Add some!](#)'. Below it is a purple box with a tip: 'Tip: Don't try this on a public server! It's not very secure!'. To the right is a green box titled 'Welcome to TaskStep' with text about the app's purpose and a note about security. At the bottom right of the main area is a green box stating 'There are currently 0 tasks left to do.'

Améliorations :

The screenshot shows the TaskStep 2.0 application. The sidebar on the left has a single button 'Ajouter une tâche'. The main area features a yellow box titled 'Formulaire de tâche' with the message 'Pas de tâches immédiates en attente!' and a green 'Ajouter une tâche' button. Below it is a blue box titled 'Bienvenue sur TaskStep' with text about the app's purpose and a 'Aide' button. The top navigation bar includes 'Paramètres', 'Aide', and 'Déconnexion'.

Les boutons ont été déplacé pour être à des endroits plus ergonomiques tout en gardant le bandeau de gauche

- Sur le formulaire d'ajout d'un item, le champ URL est très proche du bouton "Add Item"

Url:

Résolution : Les 2 champs sont beaucoup plus séparés visuellement maintenant

URL(*)

C. 1.3 Feedback Immédiat

Note : 3/4

Nouvelle note : 4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

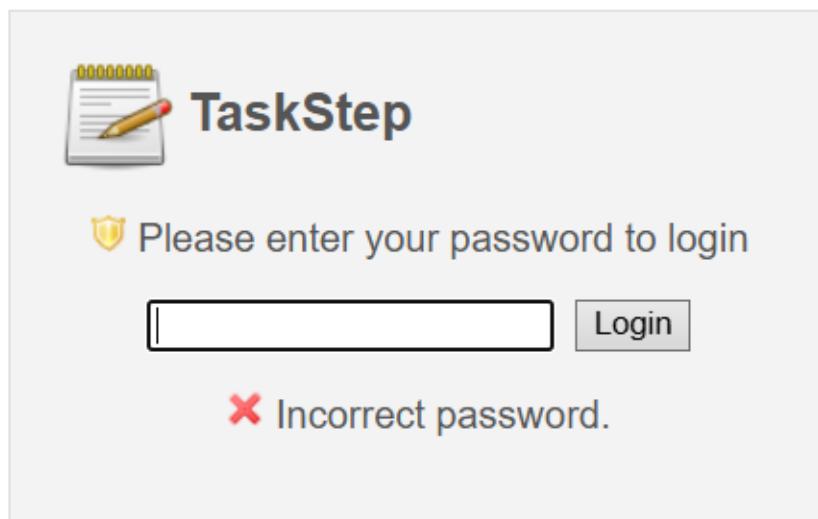
Dans tous les cas, l'ordinateur doit répondre à l'utilisateur en fonction des actions et des requêtes de ce dernier. Par exemple, dans les cas où les traitements sont longs, une information indiquant à l'utilisateur que les traitements sont en cours devrait lui être fournie.

Exemples de questions à vous poser :

- Est-ce que le site me montre tout de suite que mon action a été prise en compte ?
- Lorsque l'utilisateur vient de réaliser une action, ce qu'il vient de se passer au niveau du logiciel est-il bien compréhensible ?
- Lorsqu'une commande utilisateur prends du temps, le pointeur de souris indique-t-il à l'utilisateur que son action a bien été prise en compte et qu'il doit patienter ?

Les + :

- Feedback présent lors de la connexion si on tape un mauvais mot de passe



- Lorsqu'on ajoute une task on a bien un signal visuel pour nous montrer que l'item a été ajouté

 Item added!

Title:	<input type="text" value="Task or step title"/>		
Notes:	<input type="text"/>		
Section:	<input type="button" value="Ideas"/>	Context:	<input type="button" value="SampleContext test contexte"/>
	<input type="button" value="Might Want to Buy"/>		<input type="button" value="Project: SampleProject"/>
	<input type="button" value="Immediate"/>		
	<input type="button" value="This week"/>		
	<input type="button" value="This month"/>		
	<input type="button" value="This year"/>		
	<input type="button" value="Some day maybe"/>		
		<input type="button" value="Edit contexts"/>	<input type="button" value="Edit projects"/>
Due date:	<input type="text"/>		
Url:	<input type="text"/>		
<input type="button" value="Add item"/>			

- Message d'erreur si on essaie d'ajouter une task

 Sorry, you need to specify a context, project and section.

Title:	<input type="text" value="Task or step title"/>		
Notes:	<input type="text"/>		
Section:	<input type="button" value="Ideas"/>	Context:	<input type="button" value="SampleContext test contexte"/>
	<input type="button" value="Might Want to Buy"/>		<input type="button" value="Project: SampleProject"/>
	<input type="button" value="Immediate"/>		
	<input type="button" value="This week"/>		
	<input type="button" value="This month"/>		
	<input type="button" value="This year"/>		
	<input type="button" value="Some day maybe"/>		
		<input type="button" value="Edit contexts"/>	<input type="button" value="Edit projects"/>
Due date:	<input type="text"/>		
Url:	<input type="text"/>		
<input type="button" value="Add item"/>			

Les - :

- Sur la page d'accueil, lorsqu'on clique sur un bouton, aucun signal visuel n'est présent ce qui empêche de vraiment savoir si un click est bien pris en compte

Résolution :

Maintenant tous les boutons du site change de couleurs lorsqu'ils sont survolés et donne un feedback visuel s'ils sont cliqués

The screenshot shows a user interface with a dark blue header bar containing three buttons: "Paramètres" (Settings), "Aide" (Help), and "Déconnexion" (Logout). Below the header is a navigation bar with four items: "Accueil" (Home), "Tous les éléments" (All elements), "Par Contexte" (By Context), and "Par Projet" (By Project). A cursor is hovering over the "Tous les éléments" button, which has a light blue background and a white border. To the right of the navigation bar, there is a dropdown menu titled "+Ajouter une tâche" (Add a task). This menu lists several options: "Immédiat" (Immediate), "Idées" (Ideas), "Achats potentiels" (Potential purchases), "Cette semaine" (This week), "Ce mois-ci" (This month), "Cette année" (This year), and "Un jour peut-être" (One day maybe). The "Idées" option is highlighted with a red border and a black outline, and a cursor is hovering over it. The rest of the page is mostly blank white space.

1.4 Lisibilité

Note : 1.5/4

Nouvelle note : 4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Le critère Lisibilité concerne les caractéristiques de présentation des informations (éventuellement multimodales) dans un environnement visuel pouvant entraver ou faciliter la lecture de ces informations (luminance, contraste, dimension des objets, espacement entre les objets, fréquence sonore, intensité, timbre, etc.). Par convention, le critère Lisibilité ne concerne ni le feedback ni les messages d'erreurs.

Exemples de questions à vous poser :

- Peut-on facilement lire ce qui est écrit à l'écran (taille, contraste...) ?
- Peut-on facilement voir les pictogrammes ?

Les + :

- ---pas de remarque---

Les - :

- Le site entier est au format 4:3 et non en plein écran ce qui fait que les icônes, le texte et les éléments du site sont illisibles pour une personne qui renconterait des difficultés visuelles
- La taille du site en elle-même est très petite à un niveau de zoom 100%, rendant sa lecture très difficile de façon générale sans zoomer

Résolution :

Tout le site est désormais utilisable en plein écran de plus il est responsive et utilisable sur téléphone

Formulaire de Contexte

16/05/2025

Paramètres Aide Déconnexion

Accueil Tous les éléments Par Contexte Par Projet

Ajouter un Contexte

Titre : ss

Valider

Actions :

- Ajouter une tâche
- Immédiat
- Ideas
- Achats potentiels
- Cette semaine
- Ce mois-ci
- Cette année
- Un jour peut-être

Page des paramètres

● Affichage

Choisir un style

Default

Valider



Mot de passe

Mot de passe courant*

Nouveau mot de passe*

Description :

L'ensemble des éléments de l'interface qui a un rôle dans la réduction de la charge perceptive ou mnésique des utilisateurs, de même que dans l'augmentation de l'efficacité du dialogue.

A. 2.1 Brièveté

Note : 2.5 /4

Nouvelle note :3.5/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Limiter le travail de lecture, d'entrée et les étapes par lesquelles doivent passer les usagers.

Cela se réalise en faisant attention essentiellement à 2 points :

2.1.1 Concision

Réduire la charge de travail au niveau perceptif et mnésique pour ce qui est des éléments individuels d'entrée ou de sortie. Par exemple, lorsqu'une unité de mesure est associée à un champ de données, celle-ci doit faire partie du label du champ plutôt qu'être saisie par l'utilisateur.

Exemples de questions à vous poser :

- Peut-on rendre les messages textuels plus concis (brefs), tout en restant compréhensible ?
- Les intitulés pour des actions banales sont-ils remplacés par des pictogrammes ?

2.1.2 Actions Minimales

Limiter les étapes par lesquelles doivent passer les utilisateurs. Par exemple, ne pas demander aux utilisateurs d'entrer des données qui peuvent être déduites par le système.

Exemples de questions à vous poser :

- Toutes les actions demandées à l'utilisateur sont-elles indispensables ?
- Pourrait-on préremplir certaines informations ?

- Pourrait-on diminuer les aller-retours souris – clavier ?
- La navigation avec utilisant la tabulation est-elle fonctionnelle ?
- Pourrait-on diminuer les déplacements souris ?
- Pourrait-on faciliter l'accès aux boutons (taille, placement) ?
- Les boutons ont-ils une taille suffisante pour permettre un clic rapide ?

Les + :

- Toutes les actions demandées à l'utilisateur sont nécessaires
- Navigation au clavier fonctionnel
- Les aller-retours souris-clavier sont corrects

Les - :

- Pré remplissage inutile pour le title dans l'ajout d'item (mettre le texte prérempli en placeholder) :

Title:

Résolution : le champ est rempli en placeholder et il suffit de se mettre à écrire pour que cela disparaîsse



- Déplacement de la souris :

- Sur la page print list, pour revenir à la page précédente :

The screenshot shows the TaskStep 1.1 application interface. On the left, there's a sidebar with various filters and categories: 'Add item', 'Ideas (1) (0)', 'Might Want to Buy (0) (0)', 'Immediate (0) (1)', 'This week (0) (0)', 'This month (0) (0)', 'This year (0) (0)', and 'Some day maybe (0) (0)'. The main content area is titled 'Might Want to Buy' and displays the message 'No items in this section! Add some!'. At the bottom, it says 'Taskstep 1.1 - By Rob Lowcock, Elhan Romba, and Thomas Hooge'.

- Comme montré par la figure suivante, le bouton print list renvoie la page suivante :

The screenshot shows a web application interface. At the top, there are standard browser navigation buttons: back, forward, and refresh. To the right of these are links for "Importer les marque-..." and "Débu". Below the header, the main content area has a title "Might Want to Buy" underlined. The rest of the page is mostly blank white space.

- Le seul moyen de retour est d'appuyer sur la flèche de gauche qui est présente en haut à gauche de la page

Ideas

This screenshot shows a section titled "Ideas". At the top, there are buttons for "Sort items by: Date" and "Sort". To the right is a "Print list (3 x 5 Index card)" button. Below these are some notes: "Sample task - 2007-08-27 | SampleProject | SampleContext", "Notes", and a link "http://www.taskstep.com". There are also small icons for checkmark, pencil, and trash.

- Le bouton print list, valider, modifier et supprimer sont trop éloignés du reste de la tâche. Ils sont également très proches entre eux et très petits, rendant le clic sur l'un d'entre eux très précis à effectuer

Resultat :

Nous n'avons pas pu implémenter cette fonctionnalité par manque de temps de plus cette fonctionnalité ne fonctionnait pas donc il était difficile de comprendre à quoi servait cette fonctionnalité originellement.

B. 2.2 Densité Informationnelle

Note : 2 /4

Nouvelle note :4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Réduire la charge de travail du point de vue perceptif et mnésique, pour des ensembles d'éléments et non pour des items. Par exemple, limiter la densité informationnelle de l'écran, en affichant seulement les informations nécessaires.

Exemples de questions à vous poser :

- Y a-t-il des éléments graphiques indiquant la structure de la page (identification des blocs en 3 secondes, pas nécessairement des thématiques) ?
- Les identifiants graphiques (jeu de police de caractères, couleurs, ...) sont-ils en nombre limité (6 couleurs max, 5 polices maximum) ?

Les + :

- Sur l'écran d'ajout et la page principale, la structure générale est bien identifiable en quelques secondes : barre de navigation en haut, menu à gauche, contenu principal à droite.

Les - :

- Sur la page d'ajout, tous les champs sont visibles d'un coup et empilés entre eux sur un même bloc, même si certains pourraient apparaître progressivement selon le contexte (exemple : n'afficher "URL" que si c'est nécessaire). L'utilisateur pourrait se sentir un peu noyé s'il n'est pas familier avec l'outil.
- Il y a plusieurs types d'icônes (dessinées, emoji-like) et polices non uniformes, ce qui pourrait nuire à la cohérence visuelle.
- Il serait bon de limiter à 1 ou 2 polices, et garder un style uniforme d'icônes.
- Tous les items ont le même poids visuel, alors qu'on pourrait mettre en avant la catégorie "Immédiate" par exemple.
- Un code couleur ou une séparation plus nette des priorités aiderait.

Résultat la page d'ajout a été épuré avec un formulaire plus clair à remplir

Ajouter une tâche

Les champs marqués d'un (*) sont nécessaires.

Titre(*)

Entrez un titre à la tâche

Notes(*)

Ajoutez quelques notes

Section(*)

Immediate

Contexte(*)

fezezfef

Projet(*)

GZ

Date limite(*)

jj / mm /aaaa



URL(*)

<https://example.com>

Soumettre

Description :

Prise en compte par le système des actions explicites des utilisateurs et le contrôle qu'ont les utilisateurs sur le traitement de leurs actions.

A. 3.1 Actions Explicites

Note : 2 /4

Nouvelle note :4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Expliciter la relation entre le fonctionnement de l'application et les actions des utilisateurs. Par exemple, l'entrée de commandes doit se terminer par une indication de fin (« Enter », « OK ») à laquelle des possibilités d'édition doivent être préalables.

Exemples de questions à vous poser :

- Est-il bien montré ce qu'engendrera chacune des actions possibles de l'utilisateur ?
- Les boutons ont-ils un label explicite ?
- Si des pictogrammes sont utilisées, sont-ils bien identifiables. Sont-ils accompagnés d'un label textuel ? Y a-t-il un tooltip associé qui permet d'avoir une explication textuelle ?

Les + :

- Quasiment tous les boutons, liens et pictogrammes ont des labels et noms explicites

Les - :

- Les boutons valider, modifier et supprimer n'ont pas de label indiquant leur action bien que le tooltip soit présent pour chacun des boutons.
- Le lien http ne montre pas explicitement où est la redirection

All Items

Sort items by: Date Sort

Print list (3 x 5 Index card)

 cnk - -0001-11-30 SampleProject SampleContext knzn zrhrtzhzr	  
--	---

Sample task - 2007-08-27 SampleProject SampleContext Notes http://www.taskstep.com	  
---	---

Résultat :

Maintenant il est bien plus visible de voir les tâches on comprend directement que fais chaque bouton

Trier par	
Titre	<input type="button" value="▼"/>
Ordre	
Croissant	<input type="button" value="▼"/>
<input type="button" value="Filtrer"/>	
Task 21	
Context: grzgrzgrzrg Project: GZRGRZ Date: 2023-01-14 Status: <input checked="" type="checkbox"/> Fait	
<input type="button" value="View"/>	<input type="button" value="Marquer comme Faite/Non Faite"/>
<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>
Task 12	
Context: fezezfef Project: GZRGRZGZR Date: 2023-03-17 Status: <input checked="" type="checkbox"/> Fait	
<input type="button" value="View"/>	<input type="button" value="Marquer comme Faite/Non Faite"/>
<input type="button" value="Modifier"/>	<input type="button" value="Supprimer"/>
↳ Liste de contextes	
<input type="button" value="Ajouter un Contexte"/>	
<p><small> ⓘ Le "contexte" de la tâche définit la situation ou le lieu où la tâche peut être effectuée (à la maison, à la maison, au téléphone, ...)</small></p>	
fezezfef	
<input type="button" value="Chercher tâches"/>	<input type="button" value="Modifier contexte"/>
<input type="button" value="Supprimer contexte"/>	
grzgrzgrzrg	
<input type="button" value="Chercher tâches"/>	<input type="button" value="Modifier contexte"/>
<input type="button" value="Supprimer contexte"/>	

B. 3.2 Contrôle Utilisateur

Note : 2 /4

Nouvelle note :4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

L'utilisateur doit pouvoir contrôler le déroulement des traitements informatiques en cours. Par exemple, autoriser l'utilisateur à interrompre tout traitement en cours.

Exemples de questions à vous poser :

- Si une action nécessite la réalisation de plusieurs étapes successives, y a-t-il un fil d'ariane permettant de savoir à quelle étape l'on se situe ?
- Si une action nécessite la réalisation de plusieurs étapes successives, peut-on facilement interrompre la procédure, revenir à une étape antérieure pour modifier un élément, puis poursuivre le traitement là où il avait été interrompu sans devoir rentrer les informations ?
- Peut-on facilement et constamment revenir à l'écran d'accueil ?
- A-t-on constamment accès au menu général ?
- Chaque page est-elle graphiquement facilement identifiable ?
- Est-il clairement indiqué où l'on se situe dans l'arborescence des écrans (ex : indication sur le menu général et dans le fil d'ariane de la page actuellement visualisée) ?

Les + :

- La barre de navigation se trouve sur toutes les pages et on a toujours accès au menu général

Les - :

- Si on ne connaît pas la page il n'y'a aucune indication de l'endroit où l'on se trouve dans le site. Le fait qu'aucune page ne soit titrée n'aide pas non plus à reconnaître où l'on est.

The screenshot shows the TaskStep 1.1 application interface. At the top, there is a navigation bar with links for 'Home', 'All Items', 'By context', 'By project', 'Settings', 'Help', and 'Logout'. On the left, there is a sidebar with icons for 'Add item', 'Ideas (0) (1)', 'Might Want to Buy (0) (0)', 'Immediate (0) (0)', and 'This week (0) (0)'. The main area contains a form for creating a task or step. The 'Title' field is empty. The 'Notes' field is also empty. Below these fields are dropdown menus for 'Section', 'Context', and 'Project'. The 'Section' dropdown shows 'Ideas', 'Might Want to Buy', 'Immediate', 'This week', and 'This year'. The 'Context' dropdown shows 'SampleContext' and 'test contexte'. The 'Project' dropdown shows 'SampleProject'. At the bottom of the form, there are buttons for 'Edit contexts' and 'Edit projects'.

Résultat :

Maintenant dans chaque page le bandeau du haut possède le nom de la page

The screenshot shows a dark blue header bar with the text "Page d'accueil" on the left. On the right, there are four buttons: "Paramètres", "Aide", and "Déconnexion", followed by a small arrow icon. Below the header, the date "15/05/2025" is displayed. To the right of the date are four navigation links: "Accueil", "Tous les éléments", "Par Contexte", and "Par Projet". A yellow horizontal bar highlights the text "Formulaire de tâche" in the center of the header area.

Description :

Capacité à réagir selon le contexte et selon les besoins et les préférences des utilisateurs.

A. 4.1 Flexibilité

Note : 2 /4

Nouvelle note : 2/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Mettre à la disposition des utilisateurs des moyens pour personnaliser l'interface afin de rendre compte de leurs stratégies ou habitudes de travail et des exigences de la tâche. Par exemple, les utilisateurs doivent pouvoir désactiver des affichages inutiles.

Exemples de questions à vous poser :

- L'utilisateur peut-il remodeler l'interface en fonction de ses souhaits ?

Les + :

- Possibilité de changer de style, d'afficher ou non les tips sur la page

Display

Display tips on the front page:

Stylesheet

default.css 

Update settings

Les - :

- Le site manque d'ajustements utilisateur en dehors du choix du CSS

Résultat :

Par manque de temps nous n'avons pas pu améliorer ce point

B. 4.2 Prise en Compte de l'Expérience de l'Utilisateur

Note : NA /4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Le système doit respecter le niveau d'expérience de l'utilisateur. Par exemple, prévoir des choix d'entrées pas-à-pas ou multiples selon l'expérience des utilisateurs.

Exemples de questions à vous poser :

- Y a-t-il un plusieurs modes proposés si plusieurs profils utilisateurs sont identifiés (ex : mode expert = interface plus dense mais plus rapide) ?

Non applicable dans le cas du site car seuls des changements de l'interface homme-machine sont possibles par des changements de feuilles CSS changeant l'apparence globale du site.

Aucune fonctionnalité n'est disponible à un type d'utilisateurs en particulier, un seul type d'utilisateur existe sur ce site web.

The screenshot shows the TaskStep 1.1 application interface. At the top, there is a dark header bar with the title "TaskStep 1.1". Below the header, the main content area has a light blue background. On the left side, there is a sidebar with various navigation links: "Add item", "Ideas (0) (1)", "Might Want to Buy (0) (0)", "Immediate (0) (0)", "This week (0) (0)", "This month (0) (0)", "This year (0) (0)", and "Some day maybe (0) (0)". In the center, there is a yellow box titled "Immediate (0)" containing the message "No immediate items left to-do! [Add some!](#)". Below this, there is a purple box with a tip: "Tip: You can list all the items by context or project.". On the right side, there is a green box titled "Welcome to TaskStep" containing the message "Good afternoon, and welcome to TaskStep. TaskStep is designed to help you with day-to-day tasks, long-term aims and general lists, vaguely organising them GTD-style with contexts and projects." Below this, another green box contains the message "For those of you unfamiliar with the idea, anything which requires more than one action/step is a project. A context is where you will be doing the action, for example, at your computer." At the bottom right of the central area, there is a message "There is currently 1 task left to do."

TaskStep 1.1

Today: 16th April 2025 Home All Items By context By project Settings Help Logout

- Add item
- Ideas (0) (1)
- Might Want to Buy (0) (0)
- Immediate (0) (0)
- This week (0) (0)
- This month (0) (0)
- This year (0) (0)
- Some day maybe (0) (0)

⚡ Immediate (0)
No immediate items left to-do! [Add some!](#)

Tip: You can now select the date from a calendar. Just click in the date box as if you were typing.

Welcome to TaskStep

Good afternoon, and welcome to TaskStep. TaskStep is designed to help you with day-to-day tasks, long-term aims and general lists, vaguely organising them GTD-style with contexts and projects.

For those of you unfamiliar with the idea, anything which requires more than one action/step is a project. A context is where you will be doing the action, for example, at your computer.

There is currently 1 task left to do.

Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hooge

TaskStep

Today: 16th April 2025 Home All Items By context By project Settings Help Logout

- Add item
- Ideas (0) (1)
- Might Want to Buy (0) (0)
- Immediate (0) (0)
- This week (0) (0)
- This month (0) (0)
- This year (0) (0)
- Some day maybe (0) (0)

⚡ Immediate (0)
No immediate items left to-do! [Add some!](#)

Tip: The calendar script is JACS by Anthony Garrett.

Welcome to TaskStep

Good afternoon, and welcome to TaskStep. TaskStep is designed to help you with day-to-day tasks, long-term aims and general lists, vaguely organising them GTD-style with contexts and projects.

For those of you unfamiliar with the idea, anything which requires more than one action/step is a project. A context is where you will be doing the action, for example, at your computer.

There is currently 1 task left to do.

Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hooge

XXXVI. 5. Gestion des Erreurs

Description :

Moyens permettant d'une part d'éviter ou de réduire les erreurs, d'autre part de les corriger lorsqu'elles surviennent.

A. 5.1 Protection Contre les Erreurs

Note : 0 /4

Nouvelle note :4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Mettre en place des moyens pour détecter et prévenir les erreurs. Par exemple, toutes les actions possibles sur une interface doivent être envisagées et plus particulièrement les appuis accidentels des touches du clavier afin que les entrées non-attendues soient détectées.

Exemples de questions à vous poser :

- Tous les champs de saisie sont-ils bien adaptés à l'information qui doit être renseignée ?

Les + :

- La plupart des champs textes comme utilisé pour les mots de passes attendent bien un champ texte.



Les - :

- Il y a la possibilité de saisir du texte dans des champs (de date par exemple) et aucune erreur n'est levée, ce qui montre que le site ne l'a pas prévu. Il existe donc non seulement un manque de gestion d'erreur en HTML

mais aussi le placement d'une date différente dans le cas où une mauvaise aurait été entrée.

Sample task - 2007-08-27

Due date:

Url:

Taskstep 1

M	T	W	T	F	S	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

Today: 16/04/2025

and Thomas Hooge

- Toujours lors de la saisie, le site attend que l'utilisateur saisisse un lien url dans un champ de saisie de type texte, aucune vérification n'est possible quant à s'il s'agit d'un URL ou non :

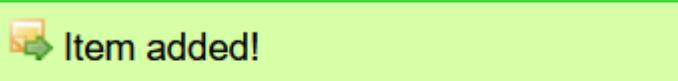
Url:

- L'utilisateur peut cliquer sur "Add item" et donc la classe est créé automatiquement, mais lors des appuis accidentels, il n'y a pas de pop-up pour confirmer la saisie afin de confirmer la création. Il n'y aussi pas la possibilité d'annuler la saisie quand la confirmation a été faite de manière malencontreuse.
- Le titre d'une task peut être laissée vide

Title:

Notes:

Section:	Context:	Project:
<input type="button" value="Ideas"/>	<input type="button" value="SampleContext"/>	<input type="button" value="SampleProject"/>



Title:

Notes:

Might Want to Buy

Sort items by:

 - 2025-04-03 | SampleProject | SampleContext

rereerer

eztez

Résultat :

Tous ces soucis sont gérés en HTML : les champs sont limités selon des contraintes posées (champs URL devant être un URL, tous les champs doivent être remplis, la date est imposée comme une date, ...), de même en PHP côté serveur où chaque champs se doit d'être inséré et doit être typé de la bonne façon.

Les champs marqués d'un (*) sont nécessaires.

Titre(*)

Entrez un titre à la tâche

Veuillez compléter ce champ.

Date limite(*)

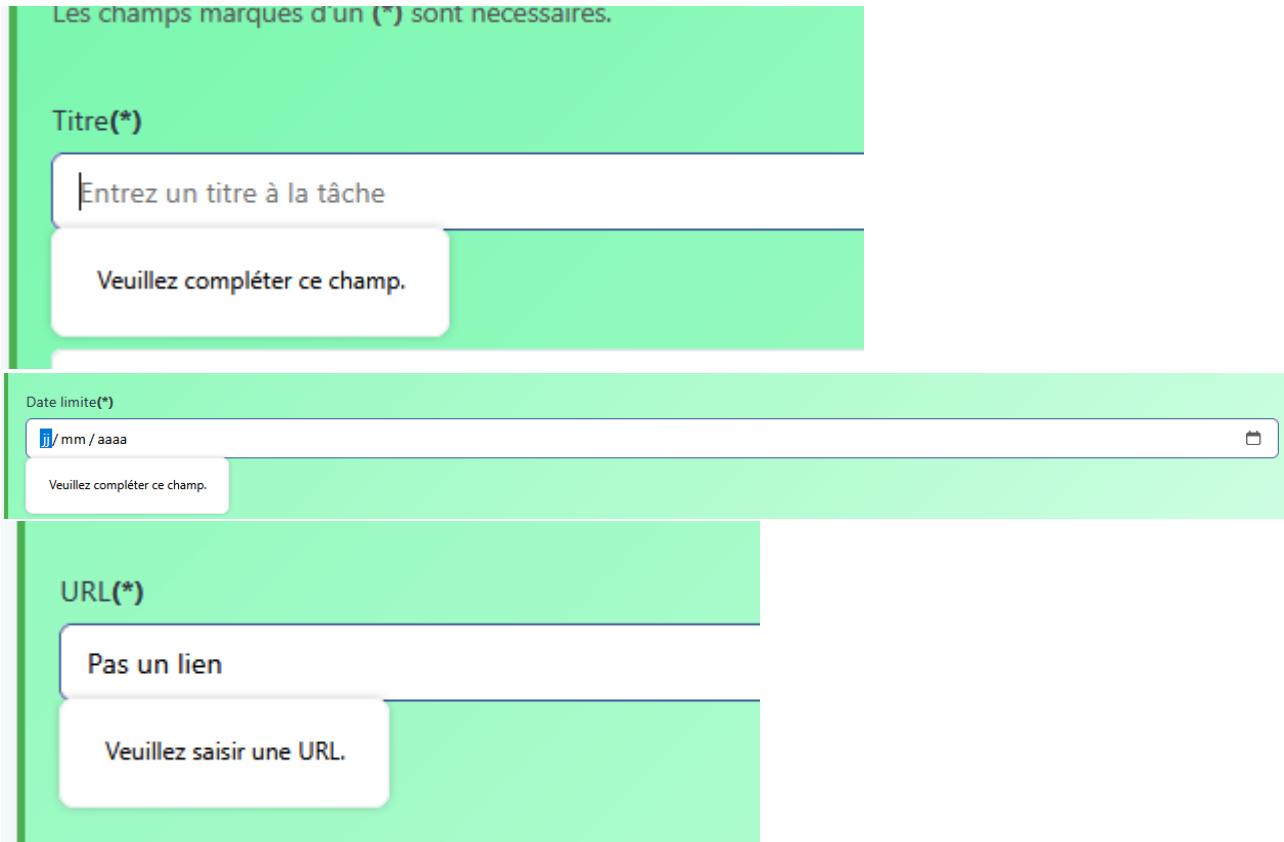
dd/mm/aaaa

Veuillez compléter ce champ.

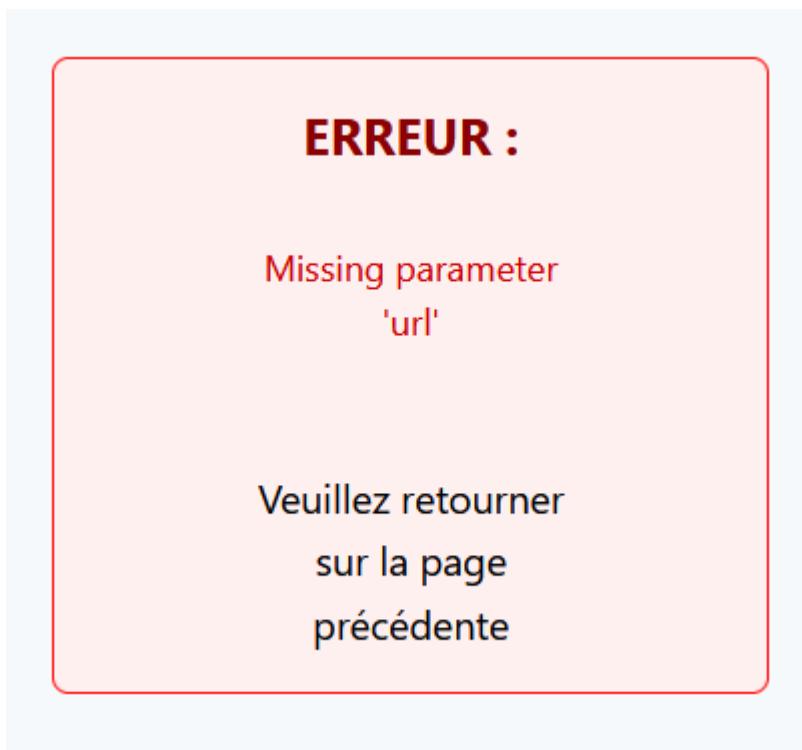
URL(*)

Pas un lien

Veuillez saisir une URL.



De même, en cas d'exception, la tâche n'est pas ajoutée en base de données et renverra simplement une exception :



B. 5.2 Qualité des Messages d'Erreurs

Note : 3/4

Nouvelle note : 4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

S'assurer que l'information donnée aux utilisateurs sur la nature des erreurs commises (syntaxe, format, etc.) et sur les actions à entreprendre pour les corriger, soit pertinente, facile à lire et exacte. Par exemple, utiliser un vocabulaire neutre, non-personnalisé, non réprobateur dans les messages d'erreurs ; éviter l'humour.

Exemples de questions à vous poser :

- Tous les signalements d'erreurs sont-ils bien compréhensibles ?

Les + :

- Le message d'erreur est explicite lorsqu'on ne sélectionne pas de project et de section :

 Sorry, you need to specify a context, project and section.

Title:	Task or step title		
Notes:	grrhrhe		
Section:	Ideas Might Want to Buy Immediate This week This month This year Some day maybe	Context:	SampleContext
Project:	SampleProject		
	 Edit contexts		 Edit projects
Due date:	heheteh		
Url:	htheeht		
	<input type="button" value="Add item"/>		

Les - :

- Lorsqu'on sélectionne seulement un ou deux des éléments parmi les 3 listes, le même message d'erreur

apparaît alors que l'on a sélectionné certains des éléments mentionnés

Résultat :

Lorsqu'on ne remplit pas un champ celui-ci est directement marqué si on essaie de valider

Ajouter une tâche

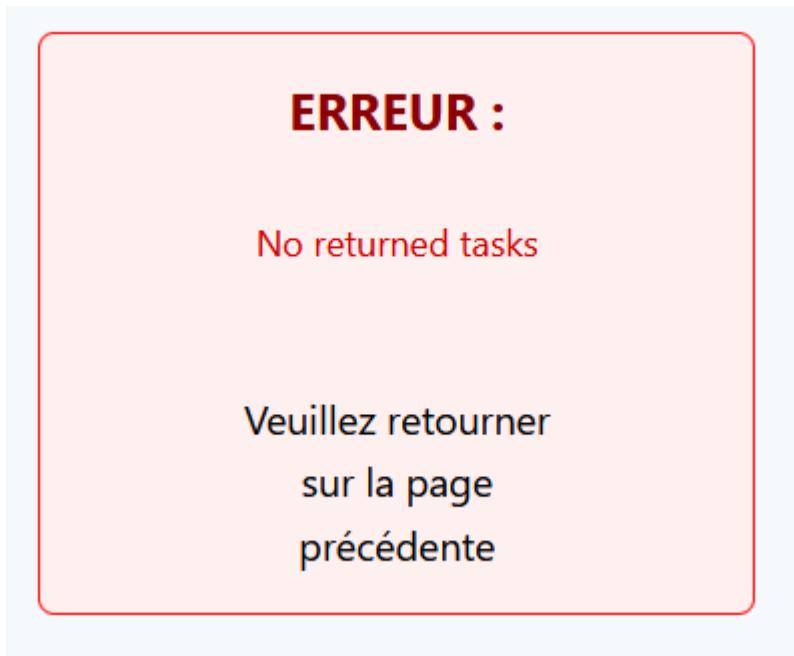
Les champs marqués d'un (*) sont nécessaires.

Titre(*)

Notes(*)

! Veuillez renseigner ce champ.

De même, des messages d'exception assez clairs sont indiqués à l'utilisateur en cas d'erreurs côté serveur :



C. 5.3 Correction des Erreurs

Note : 2/4

Nouvelle note : 4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Mettre à la disposition des utilisateurs des moyens pour corriger leurs erreurs. Par exemple, fournir la possibilité de modifier les commandes lors de leur saisie.

Exemples de questions à vous poser :

- Si une erreur est signalée, l'utilisateur peut-il facilement la corriger ?

Les + :

- Sur la page de connexion au site, une erreur de mot de passe entré nous indique qu'il faut insérer le bon mot de passe

The screenshot shows a login interface for 'TaskStep'. At the top left is a small icon of a notepad and pen. The title 'TaskStep' is centered above a form area. Inside the form, there is a yellow shield icon followed by the text 'Please enter your password to login'. Below this is a password input field and a 'Login' button. Underneath the input field, a red 'X' icon is followed by the text 'Incorrect password.'

Les - :

- Le message d'erreur mentionné plus tôt ne précise pas lequel des 3 champs est à remplir précisément si on en a déjà sélectionné un ou deux des 3

 Sorry, you need to specify a context, project and section.

Title:

Notes:

Section:

Context:

Project:

 [Edit contexts](#)  [Edit projects](#)

Due date:

Url:

Résultat d'optimisation :

En HTML, le champ qui doit être modifié est indiqué directement à l'utilisateur :

The screenshot shows a web form with two fields highlighted in green. The first field, labeled "URL(*)", contains the text "Pas un lien" and has a tooltip "Veuillez saisir une URL.". The second field, labeled "Date limite(*)", contains the placeholder text "jj/mm/aaaa" and has a tooltip "Veuillez compléter ce champ.". Both fields have a red asterisk indicating they are required.

URL(*)	Pas un lien Veuillez saisir une URL.
Date limite(*)	jj/mm/aaaa Veuillez compléter ce champ.

6. Homogénéité/Cohérence

Note : 3/4

Nouvelle note : 4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Les choix de conception d'interface doivent être conservés pour des contextes identiques, et doivent être différents pour des contextes différents. Par exemple, toujours afficher au même endroit l'incitation pour la saisie des données ou des commandes.

Exemples de questions à vous poser :

- Les différentes pages ont-elles la même structure ?
- La charte graphique (couleurs, formes géométriques, police...) est-elle conservée d'un écran à l'autre ?

Les + :

- Les couleurs sont plutôt unis des teintes de gris, de noir et de blanc, il n'y a pas de grand contraste entre les différents éléments des pages

Les - :

- La page d'accueil est trop colorée

TaskStep 1.1

Today: 16th April 2025 Home All Items By context By project Settings Help Logout

- Add item
- Ideas (0) (2)
- Might Want to Buy (0) (0)
- Immediate (0) (0)
- This week (0) (0)
- This month (0) (0)
- This year (0) (0)
- Some day maybe (0) (0)

Immediate (0)
No immediate items left to-do! [Add some!](#)

Tip: The calendar script is JACS by Anthony Garrett.

Welcome to TaskStep
Good afternoon, and welcome to TaskStep. TaskStep is designed to help you with day-to-day tasks, long-term aims and general lists, vaguely organising them GTD-style with contexts and projects.

For those of you unfamiliar with the idea, anything which requires more than one action/step is a project. A context is where you will be doing the action, for example, at your computer.

There are currently 2 tasks left to do.

Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hoope

Résultat : la page a été harmonisé

Page d'accueil

POUR QUITTER LE MODE PLEIN ÉCRAN, APPUYEZ SUR **F11**

15/05/2025 Paramètres Aide Déconnexion

Accueil Tous les éléments Par Contexte Par Projet

- +Ajouter une tâche
- ↳ [Immediat](#)
- ↳ [Idées](#)
- ↳ [Achats potentiels](#)
- ↳ [Cette semaine](#)
- ↳ [Ce mois-ci](#)
- ↳ [Cette année](#)
- ? [Un jour peut-être](#)

Formulaire de tâche
Pas de tâches immédiates en attente!

[Ajouter une tâche](#)

Astuce: Ajouter une tâche avec le bouton juste ici

Bienvenue sur TaskStep
Bonjour, et bienvenue sur TaskStep. TaskStep est conçu pour vous aider avec vos tâches quotidiennes, vos objectifs à long terme et les listes générales, en les organisant vaguement selon le style avec des contextes et des projets.

Pour ceux d'entre vous qui ne connaissent pas ce concept, tout ce qui nécessite plus d'une étape/action est un projet. Un contexte est l'environnement où vous effectuerez l'action, par exemple sur votre ordinateur.

[Aide](#)

Pour plus d'informations cliquez sur ce bouton

XXXVII. 7. Significance des Codes et Dénotations

Note : 3 /4

Nouvelle note :4/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Il doit y avoir adéquation entre l'objet ou l'information affichée ou entrée, et son référent. Par exemple, rendre les règles d'abréviation explicites.

Exemples de questions à vous poser :

- Le vocabulaire ou abréviations non communes sont-ils bien explicités (soit avec un glossaire, soit avec un tooltip) ?

Les + :

- L'abréviation GTD ci-dessous est expliquée à travers un lien

The screenshot shows a window titled "Welcome to TaskStep". The main text reads: "Good afternoon, and welcome to TaskStep. TaskStep is designed to help you with day-to-day tasks, long-term aims and general lists, vaguely organising them GTD-style with contexts and projects." Below this, another paragraph explains: "For those of you unfamiliar with the idea, anything which requires more than one action/step is a project. A context is where you will be doing the action, for example, at your computer." At the bottom, there is a status bar with a small icon and the text "There are currently 2 tasks left to do."

Les - :

- On pourrait également avoir entre parenthèses ce à quoi correspondent les initiales GTD sur la page.

Résultat :

Ajout de la signification dans les parenthèses

Présentation de TaskStep

TaskStep est une application de gestion de tâches open-source inspirée de la méthode GTD(Getting Things Done) pour organiser efficacement vos actions.

XXXVIII. 8. Compatibilité

Note : 0 /4

Nouvelle note :3.5/4

(0 = très mauvais, 1 = mauvais, 2 = moyen, 3 = bon, 4 = très bon)

Description :

Il faut qu'il y ait accord entre les caractéristiques des utilisateurs et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part. Par exemple, les termes employés doivent être familiers aux utilisateurs, et relatifs à la tâche à réaliser.

Exemples de questions à vous poser :

- L'outil s'adapte-t-il bien sur tous les supports possibles ?

Les + :

- Le site s'adapte à tous les navigateurs

Les - :

- Non accessible selon la taille de l'écran :



Today: 16th April 2025

Home

All Items

By context

By project

Settings

Help

[+ Add item](#)[💡 Ideas \(1\) \(0\)](#)[🛒 Might Want to Buy \(0\) \(0\)](#)[⚡ Immediate \(0\) \(1\)](#)[📅 This week \(0\) \(0\)](#)[📅 This month \(0\) \(2\)](#)[🌐 This year \(0\) \(0\)](#)[👤 Some day maybe \(0\) \(0\)](#)

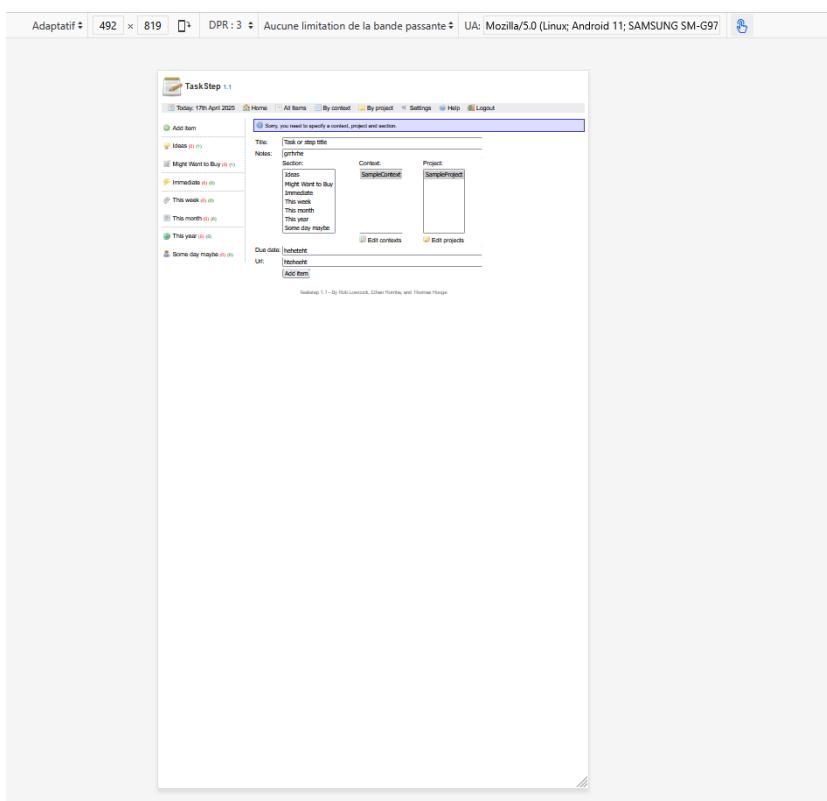
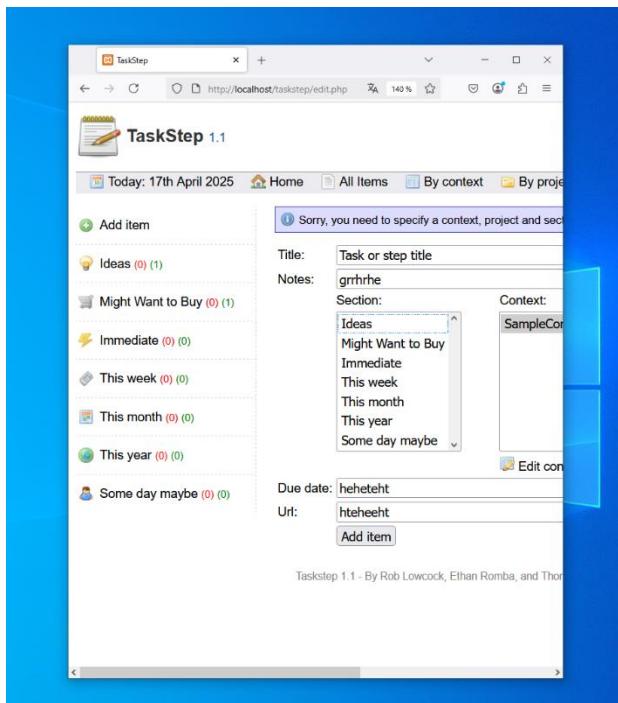
Today: 16th April 2025

Sort items by: [Done](#) [Sort](#)[🖨 Print list \(3\)](#)

ⓘ No items today! Either there is nothing to do, or you should [Add some!](#)

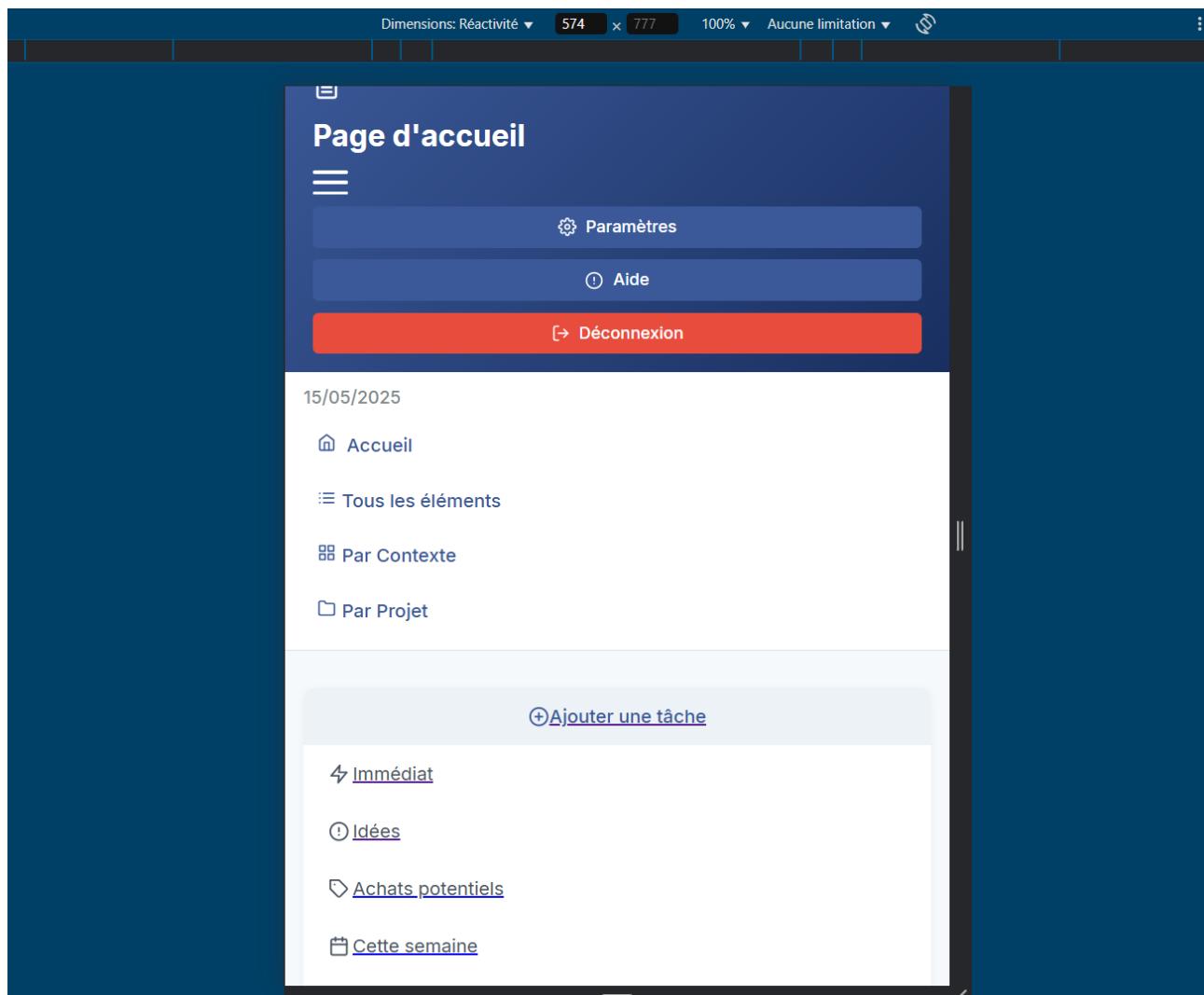
Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hooge

- Le site n'est pas responsive et donc pas adapté aux appareils mobiles



Résultat :

Le site est responsive et s'adapte à tous les écrans :



XXXIX. Synthèse d'optimisation

Ici est présenté un bilan général du site ainsi que les remarques les plus importantes faites précédemment dans le document en faisant référence précisément aux sections du document concernées

Synthèse écrite :

La saillance visuelle de l'Interface Homme-Machine de Taskstep a été presque intégralement corrigée.

Le site Taskstep a révélé de nombreuses failles critiquées, liées principalement à un manque de mesures d'ergonomie. Il n'était pas responsive, l'affichage était obsolète (format 4:3), la navigation était trop peu intuitive et le feedback était peu précis, si ce n'est inexistant. De même, les formulaires étaient mal conçus et nécessitaient une sérieuse remise à niveau.

Ainsi, nous avons pu mettre à jour l'affichage du site pour le rendre responsive et en format 16:9, adapté aux écrans de nos jours. La navigation a été rendue beaucoup plus claire à l'aide de titres de page et de boutons plus clairs sur là où ils redirigent. Les différents messages d'erreurs étaient également plus précis après correction, nous indiquant beaucoup plus clairement ce qu'il faut réparer en cas d'erreur.

Les 3 problèmes prioritaires et actions correctives :

Responsivité du site : Le site ne s'adaptait pas correctement à toutes les tailles d'écran, ce qui impliquait nécessairement une mauvaise expérience du site. Durant la refonte du site web, nous avons donc pu ajuster le CSS afin de le rendre responsive et accessible sur les écrans mobiles

Format 4:3 vers 16:9 : Le format même du site devait être refait afin de le rendre accessible à un écran 16:9 plutôt que le format 4:3 actuel. Nous l'avons donc en grande partie géré avec la responsivité du site mais plusieurs autres ajustements auront été faits dans le code HTML/CSS du site web afin de le rendre beaucoup plus satisfaisant à utiliser

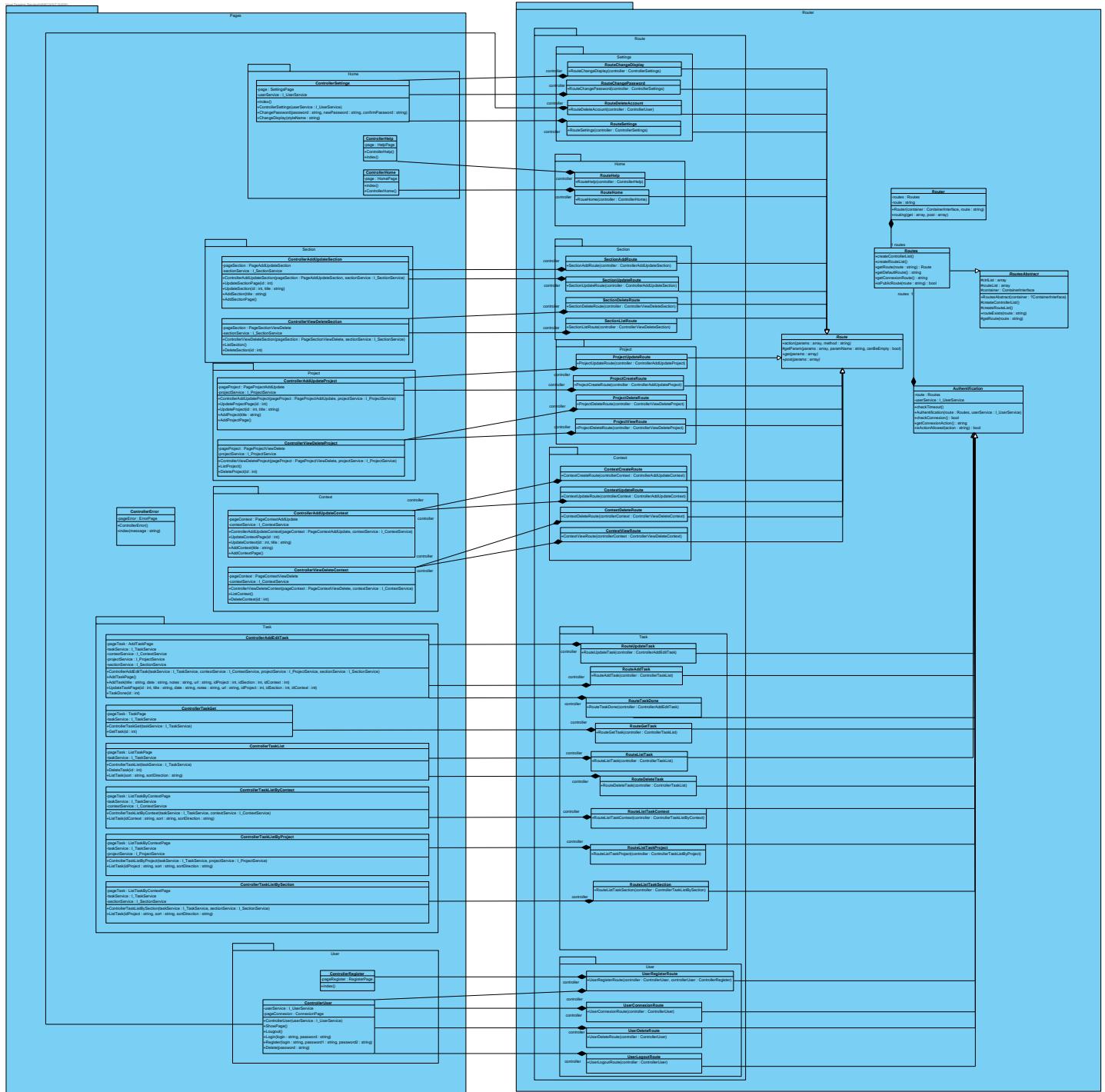
Saisie de données : Plusieurs labels et placeholders étaient à mettre en place ou réajuster, des instructions n'étaient pas assez pertinentes et pas assez de limites posées sur les données que l'on peut entrer. Tous les formulaires ont donc été refaits et sont désormais beaucoup plus agréables à remplir, indiquant clairement ce qu'il faut remplir et où.

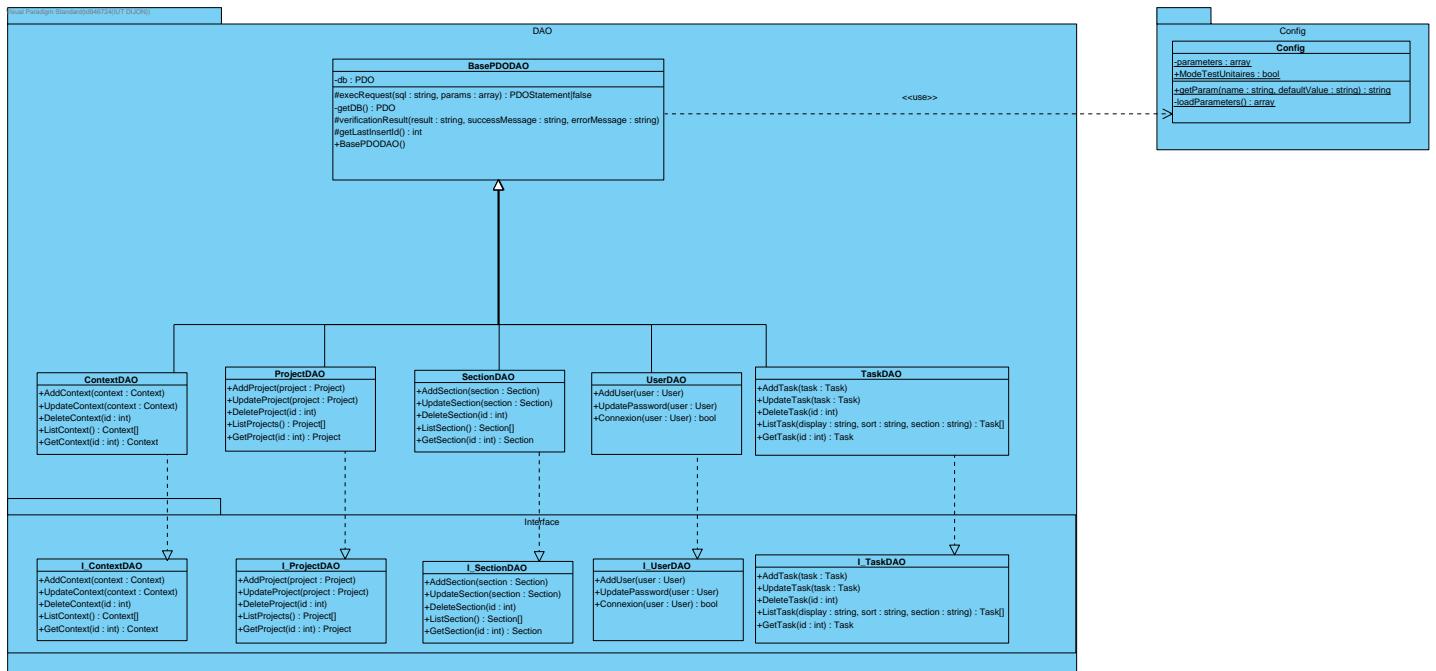
Résumé des optimisations :

1. Guidage		Résultats
1.1 Lisibilité	Site au format 4:3 et très petit à un zoom 100% (1.5/4)	Site au format 16:9 et zoom ajusté (4/4)
1.2 Incitation	Plusieurs labels mal identifiés, soulignements de liens inexistantes, liens peu professionnels (1/4)	Labels mieux identifiés, liens bien plus corrects (4/4)
1.3 Groupement/Distinction entre Items	Disposition des éléments illogiques et trop rapprochés, noms de liens illogiques (1/4)	Disposition des éléments plus rapprochés et logiques (4/4)
1.4 Feedback immédiat	On ne sait pas quand un click est pris en compte (3/4)	Boutons changeant de couleur quand on survole ou clique dessus (4/4)
2. Charge de travail		
2.1 Brièveté	Pré-remplissages inutiles, boutons très mal disposés (trop rapprochés entre eux et éloignés de ce qu'ils concernent) (2.5/4)	Pré-remplissages mieux implémentés, boutons mieux disposés (3.5/4)
2.2 Densité Informationnelle	Incôhérence visuelle (icônes, polices non uniformes) et trop d'informations empilées entre elles (2/4)	Les champs du formulaire ont moins d'informations compactées (4/4)
3. Contrôle explicite		
3.1 Actions Explicites	Pas de labels aux boutons (2/4)	Des labels ont été attribués aux boutons (4/4)
3.2 Contrôle Utilisateur	Aucune indication de la page où on se trouve actuellement (pas de titre h1, ou de titre de page) (2/4)	Chaque page contient un titre h1 permettant de connaître la page actuelle et la page contient un titre en head(4/4)
4. Adaptabilité		
4.1 Flexibilité	Pas assez modulaire (2/4)	Manque de temps pour ajouter de la modularité (2/4)
4.2 Prise en Compte de l'Expérience de l'Utilisateur	Non applicable : pas assez de modes différents d'utilisation	/
5. Gestion des Erreurs		
5.1 Protection Contre les Erreurs	Saisie de mauvais types de données dans certains champs (texte dans une date par exemple), champs nécessaires pouvant être vides (0/4)	Tous les champs de données se doivent d'être remplis et dans de bons types de données (4/4)
5.2 Qualité des Messages d'Erreurs	Même message d'erreur pour plusieurs erreurs (3/4)	Messages d'erreur beaucoup plus clairs et précis (4/4)
5.3 Correction des Erreurs	Même message d'erreur pour plusieurs erreurs (on ne sait pas laquelle corriger) (2/4)	Champs HTML indiquant où modifier ce qu'il faut pour éviter des erreurs (4/4)
6. Homogénéité/Cohérence		
7. Significance des Codes et Dénominations	On pourrait avoir les initiales correspondant à une abréviation lorsqu'il y en a (3/4)	Signification d'initiales ajoutées en paramètres (4/4)
8. Compatibilité		
	Non accessible selon la taille de l'écran, pas responsive (0/4)	Le site est maintenant responsive (3.5/4)

XL.Tableau des tâches

Recommandations	Priorité(X)	Complexité(Y)	Effort (taille)	gravité	Optimisation
Bouton retour à la page précédente	1	1	1	1	
Modules de site (plus d'adaptations possibles)	1	4	3	1	
Initiales à des abréviations	1	1	1	1	
Page d'accueil trop colorée	2	1	1	2	
Soulignement de liens	3	1	1	2	
Boutons mal disposés	3	2	1	1	
Incohérence visuelle (icônes, police, informations empilées)	3	2	2	1	
Message d'erreur commun à plusieurs erreurs	3	2	1	1	
Labels mal identifiés	4	1	1	3	
Disposition des éléments illogiques et trop rapprochés	4	2	2	2	
Noms de liens illogiques	4	1	1	2	
Pré-remplissages de formulaire inutile	4	1	1	2	
Labels aux boutons	4	1	2	2	
Titre de page	4	1	1	2	
Noms peu professionnels	4	1	1	2	
Retour visuel	5	2	2	2	
Format 16:9 et lisible à 100% de zoom	5	4	2	3	
Mauvais type de données pouvant être entrés	5	2	1	3	
Responsivité du site	5	4	3	3	
Intuitivité/navigation	6	2	2	3	





ContextDAO

ProjectDAO

SectionDAO

UserDAO

TaskDAO

I_ContextDAO

I_ProjectDAO

I_SectionDAO

I_UserDAO

I_TaskDAO

