

Développement d'un outil de validation pour HTML, CSS, JavaScript, Bootstrap

Bechir NASR, Maxence DELFANTI

Mai 2020

Résumé : Aujourd'hui, la plupart des étudiants qui entrent en première année de licence n'ont jamais développé de sites internet. Dans le cadre de l'ue POM, nous avons développé un outil pour les aider dans la correction des différentes erreurs de syntaxe qu'ils peuvent produire. En effet, ce projet regroupe différents validateurs syntaxiques selon les normes fixées par le World Wide Web Consortium. Ce site développé à l'intention des étudiants en informatique suivant l'un des cours dispensés dans la licence informatique de Lyon 1, doit leur permettre de visualiser leurs erreurs en HTML, CSS, JavaScript, Bootstrap ou sur l'évaluation de l'accessibilité de leur production.

Mots clés : Validateur, Syntaxe, Accessibilité, programmation web

1 Introduction

Ce projet s'inscrit dans le cadre du second semestre de la première année de Master en Informatique à Lyon 1. Notre sujet POM est : « Développement d'un outil de validation pour HTML, CSS, JavaScript, Bootstrap ». Ce sujet a été proposé et encadré par Stéphanie Jean-Daubias, professeure à l'Université Claude Bernard Lyon 1, chercheuse en ingénierie des connaissances et EIAH (Environnements Informatique pour l'Apprentissage Humain) dans l'équipe TWEAK du LIRIS et Fabien Duchateau, enseignant-chercheur, maître de conférences à l'Université Claude Bernard Lyon 1 et chercheur dans l'équipe Base de Données au LIRIS.

Les étudiants qui commencent leurs études dans le développement web ne sont pas encore habitués aux différentes normes syntaxiques. De ce fait différentes erreurs peuvent donc se glisser dans leur production. Aujourd'hui, s'ils veulent avoir une aide automatique à la correction, ils doivent utiliser différents outils pour chacun des langages, rendant ce travail plus fastidieux. L'objectif de ce projet est de créer un outil pédagogique pour les étudiants, réunissant plusieurs vérificateurs déjà existants en une seule application. Ceci afin de mettre en lumière les différentes erreurs qu'ils auraient pu introduire dans le site web produit dans leur TP. Et ainsi leur permettre de se corriger avant le rendu final.

Notre application, que nous avons appelée *Valide*, permet de vérifier les productions étudiantes qui auraient été produites dans les langages suivants : HTML, CSS, Javascript. Bien qu'il soit théoriquement possible d'utiliser Valide pour vérifier l'accessibilité de certains sites réalisés en PHP ou d'autres langages serveur, cela n'a pas été testé. Valide doit également être simple d'utilisation pour permettre à ses utilisateurs d'accéder rapidement à la liste de leurs erreurs.

2 Des outils existants

Bien que notre application doive vérifier et apporter des éléments de corrections syntaxiques, il n'était pas souhaitable de recréer des validateurs. En effet, il en existe déjà un certain nombre, qui sont maintenus par des organisations officielles ou une communauté de développeurs.

Tout d'abord, l'organisme de standardisation World Wide Web Consortium, ou W3C, qui a standardisé la syntaxe HTML et CSS, met à disposition du public un vérificateur nommé [Unicorn](#). Ce vérificateur possède une api nous permettant d'avoir un retour sur les fautes commises par l'apprenant.

Pour la vérification de l'écriture du code en JavaScript, nous utilisons le validateur [JSHint](#), créé en 2010, et qui est régulièrement mis à jour. Contrairement au W3C, JSHint ne possède pas de serveur dédié à la vérification, mais le code de son vérificateur est open-source, permettant ainsi sa réutilisation. Aujourd'hui, le vérificateur JSHint est utilisé par de grandes entreprises telles que Mozilla, Facebook, Twitter.

Enfin, pour l'évaluation de l'accessibilité, nous utilisons l'application [Wave](#), qui permet de visualiser une page d'un site (en ligne) et de l'annoter en indiquant les erreurs d'accessibilité présentes sur cette page. Parmi les erreurs que l'on peut trouver avec Wave, il y a des indications sur les oublis de texte alternatif pour les images (indispensable pour les personnes avec une déficience visuelle), des liens redondants, ou encore des problèmes de contraste trop faible entre la couleur d'un texte et de son fond. Tout comme le W3C, Wave possède un serveur ouvert au public. Mais contrairement à ce dernier, Wave ne permet pas de récupérer automatiquement et gratuitement la liste des erreurs. Notre application renverra donc l'utilisateur vers le site du vérificateur Wave, si ce dernier souhaite contrôler l'accessibilité de son site.

Notons qu'avec cette méthode Wave peut évaluer seulement les sites se trouvant sur le web. Pour pallier à ce problème, les productions des étudiants déposés sur Valide seront accessibles depuis l'application avec une url contenant un hash. Cela n'est pas sans conséquence, car les sites qui utilisent une base de données par exemple ne fonctionnent plus pour le test sur Wave. Cela n'est pas un souci car les sites réalisés dans le cadre de LIFIHM sont relativement simples et sans base de données.

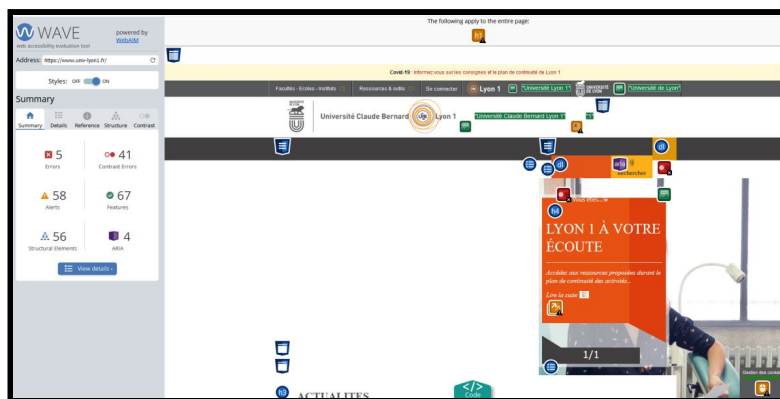


Figure 1 : Exemple d'utilisation de l'interface Wave sur le site Web de l'Université de Lyon 1

L'objectif de notre travail à travers ce projet est de regrouper différents outils en un seul endroit, tout en offrant la possibilité d'ajouter ou d'enlever des vérificateurs.

3 Travail réalisé

3.1 Des échanges réguliers

Ce travail est le fruit de différentes conversations de vive voix ou par mail avec nos encadrants. Différentes attentes de leur part ont été proposées, différentes maquettes ou réalisations de notre part ont émergé de ces discussions. Mais ces échanges auront permis de rythmer notre travail, et d'améliorer la qualité de notre production.

3.2 La liste des fonctionnalités

3.3.1 Les fonctionnalités en tant qu'étudiants

Pour les étudiants, l'application propose les fonctionnalités suivantes :

- Possibilité de rejoindre un projet existant qui n'a pas été terminé
- Visualiser la liste des projets auquel l'étudiant est inscrit
- Déposer sa production pour un projet spécifique
- Visualiser la liste de ses erreurs

3.3.2 Les fonctionnalités en tant qu'enseignants

Pour les enseignants, l'application propose les fonctionnalités suivantes :

- Visualiser la liste des projets créés par l'enseignant
- Création de projets avec les éléments suivants : titre, date de début, date butoir, liste des vérificateurs utilisés, UE concernée et description du projet
- Modification et suppression de projets existants
- Ajout/Suppression d'une liste d'étudiant à un projet
- Possibilité de télécharger la production d'un étudiant inscrit au projet
- Possibilité de visualiser les résultats des différents vérificateur sur la production d'un étudiant inscrit au projet.*

3.3 L'architecture du projet

3.3.1 Les framework utilisés

Pour produire le projet Valide, il n'y avait pas de contrainte de langage de la part de nos encadrants. Nous avons opté pour le framework PHP [Laravel](https://laravel.com/) pour le backend car Laravel est l'un des frameworks PHP les plus utilisés, et possède une documentation très importante, ce qui nous a été d'une grande aide lors du développement. De plus, ce choix a été guidé car Bechir Nasr a déjà eu l'occasion d'utiliser ce framework dans de nombreux projets.

Pour le frontend nous avons décidé d'utiliser le framework Bootstrap 4, développé par Twitter, qui met à disposition une grande liste d'éléments HTML stylisés et permet aux sites de mieux s'adapter aux différents dispositifs (tailles d'écran notamment).

3.3.2 Les échanges avec les différents vérificateurs

Après avoir déposé sa production, notre application Valide appellera les différents vérificateurs liés au projet dans lequel la production s'inscrit.

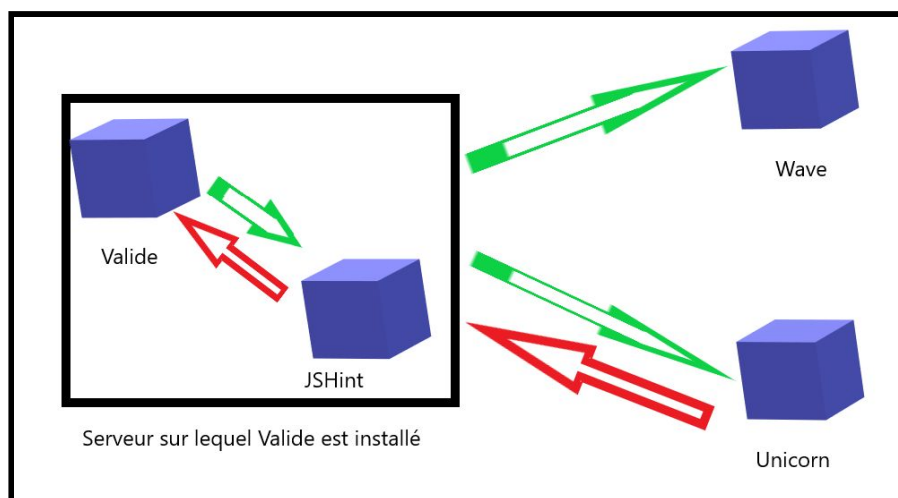


Figure 2 :Schéma montrant les différents échange entre Valide et les vérificateurs

3.3.2.1 L'échange avec le W3C

Si le vérificateur HTML et/ou CSS a été sélectionné pour vérifier le projet, comme il est montré dans la figure 2, des échanges avec le serveur du W3C (Unicorn) ont lieu pour évaluer la qualité du code html et/ou css. Pour ce faire, notre application envoie une requête [curl](#) lancée directement depuis la console du serveur vers le serveur du W3C précisant quel type de vérificateur doit être utilisé (HTML ou CSS). En réponse, Unicorn renvoie un fichier JSON contenant la liste des erreurs présentes dans le fichier envoyé précédemment. Le contenu de ce fichier json sera transformé (i.e., erreurs formatées selon notre modèle de représentation des erreurs) et ajouté à un fichier JSON plus global, contenant toutes les erreurs du projet.

3.3.2.2 L'échange avec JSHint

Comme on peut voir dans la figure 2, JSHint est installé directement sur le même serveur que notre application Valide. Si le vérificateur Javascript a été choisi pour le projet, alors chacun des fichiers JS de la production sera analysé par JSHint. Pour chaque fichier analysé JSHint renverra un fichier texte contenant la liste des erreurs. L'application Valide récupère alors toutes les informations utiles depuis ce fichier, les formate selon notre modèle d'erreur, puis les ajoute dans un fichier JSON plus global contenant la liste des erreurs du projet.

3.3.2.3 L'échange avec Wave

Contrairement aux échanges précédents, celui avec Wave ne se produit pas au moment du dépôt. Lors de la visualisation des erreurs, si l'enseignant a activé pour le projet le validateur d'accessibilité alors un champ est proposé à l'utilisateur pour y saisir le lien relatif du fichier qu'il souhaite évaluer depuis la racine de sa production. Comme montré dans la figure 2, l'application Valide redirige l'utilisateur vers le site de Wave en fournissant à Wave le lien vers la page à évaluer et qui est contenue dans l'application Valide. Wave se démarque aussi des autres vérificateurs car aucun retour vers notre application Valide n'a lieu.

3.4 Une interface pour les étudiants

Pour accéder une première fois à l'application, les étudiants doivent s'inscrire, en renseignant leur numéro étudiant. Cette contrainte est obligatoire pour pouvoir faire un rapprochement entre un profil sur notre application et un étudiant. On aurait pu outrepasser cette contrainte en utilisant le service

d'authentification (CAS) de l'Université Claude Bernard. Mais ce dernier n'est pas ouvert pour les projets POM.

3.4.1 Déposer un projet

Une fois connecté, l'étudiant arrive sur une interface lui montrant les différents projets auxquels il est inscrit.




Title	Createur	Ue	Date de creation.	Date Butoir
projetProf1	3	ue1	2020-06-01 00:00:00	2020-07-01 00:00:00
projetProf1_second	3	ue1	2020-06-01 00:00:00	2020-07-01 00:00:00

Showing 1 to 2 of 2 entries

Figure 3 : Interface étudiants : Liste des projets auxquels l'étudiant est inscrit

Un clic sur une de ces lignes permet à l'étudiant d'accéder à l'interface de dépôt de projet où il pourra déposer sa production, compressée en ZIP. Enfin, ceci lui permettra d'atteindre la page des résultats des différents vérificateurs.



Html
<ul style="list-style-type: none"> End tag "h2" seen, but there were open elements. Nom du fichier : testPOM/test1.html Ligne : 13 Start tag seen without seeing a doctype first. Expected "<!DOCTYPE html>". Nom du fichier : testPOM/test2.html Ligne : 1 Element "head" is missing a required instance of child element "title". Nom du fichier : testPOM/test2.html Ligne : 4 End tag "h2" seen, but there were open elements. Nom du fichier : testPOM/test2.html Ligne : 9 Consider adding a "lang" attribute to the "html" start tag to declare the language of this document. Nom du fichier : testPOM/test2.html Ligne : 1

Accessibilité

Figure 4 : Interface étudiants : Visualisation des résultats des vérificateurs

Sur cette page, l'utilisateur pourra voir les différentes erreurs qu'il a commises. Chacune des erreurs sera affichée avec les informations suivantes : le détail de l'erreur dans la langue du vérificateur utilisé, le nom du fichier contenant l'erreur et la ligne où elle a été produite.

Comme il a été dit précédemment l'application ne peut pas récupérer directement la liste des erreurs d'accessibilité. Si l'étudiant souhaite évaluer l'accessibilité de son site, il est invité à remplir un champ pour

indiquer le chemin relatif depuis la racine de l'archive son application vers le fichier qu'il souhaite examiner. Il sera ainsi amené sur l'interface Wave comme illustré sur la figure 1.

3.4.2 Intégrer un projet

En plus de pouvoir vérifier son travail, l'étudiant pourra rejoindre un projet. Pour ce faire, il devra cliquer sur le bouton « Chercher un projet » dans la partie « Navigation ». Il sera amené vers une page semblable à celle de la figure 2. Il pourra ainsi sélectionner un projet dans la liste proposée afin de le rejoindre.

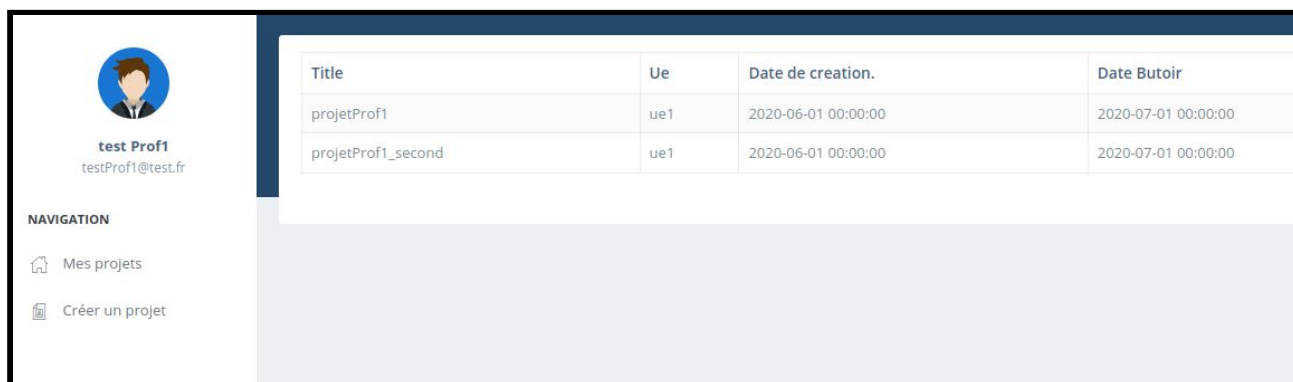
3.5 Une interface pour les enseignants

Pour accéder une première fois à l'application, les enseignants doivent s'inscrire. Ils obtiennent ainsi un profil étudiant. Il peut paraître étrange de donner un profil étudiant à un professeur, mais il nous est impossible de vérifier automatiquement l'authenticité du rôle de l'utilisateur qui souhaite s'inscrire sur l'application.

Un administrateur du site devra par la suite changer leurs autorisations dans la base de données pour obtenir le statut d'enseignant.

3.5.1 Visualiser un projet

Une fois connecté avec les bonnes autorisations, l'enseignant arrive sur l'interface "Mes projets". Cette interface a pour objectif de montrer la liste des différents projets créés par l'enseignant. Cette liste est vide au moment de la création du compte.



Title	Ue	Date de creation.	Date Butoir
projetProf1	ue1	2020-06-01 00:00:00	2020-07-01 00:00:00
projetProf1_second	ue1	2020-06-01 00:00:00	2020-07-01 00:00:00

test Prof1
testProf1@test.fr

NAVIGATION

- Mes projets
- Créer un projet

Figure 5 : Interface enseignant: Mes projets

Un clic sur une de ces lignes permet à l'enseignant d'accéder à l'interface de description de son projet. De cette page il peut :

- Ajouter une liste d'étudiants à son projet, via le numéro étudiant de ces derniers
- Avoir accès à la liste des étudiants inscrites à son projet
- Supprimer un étudiant de son projet
- Supprimer la totalité des personnes de son projet
- Avoir un visuel sur les différentes informations du projet (Titre, description, liste des vérificateurs utilisés, date de début, date butoir)
- Pouvoir télécharger la production d'un étudiant
- Pouvoir visualiser le résultat des vérificateurs d'une production déposé par l'étudiant (l'interface est la même que celle présentée en figure 3, à l'exception du menu « Navigation »)
- Modifier le projet

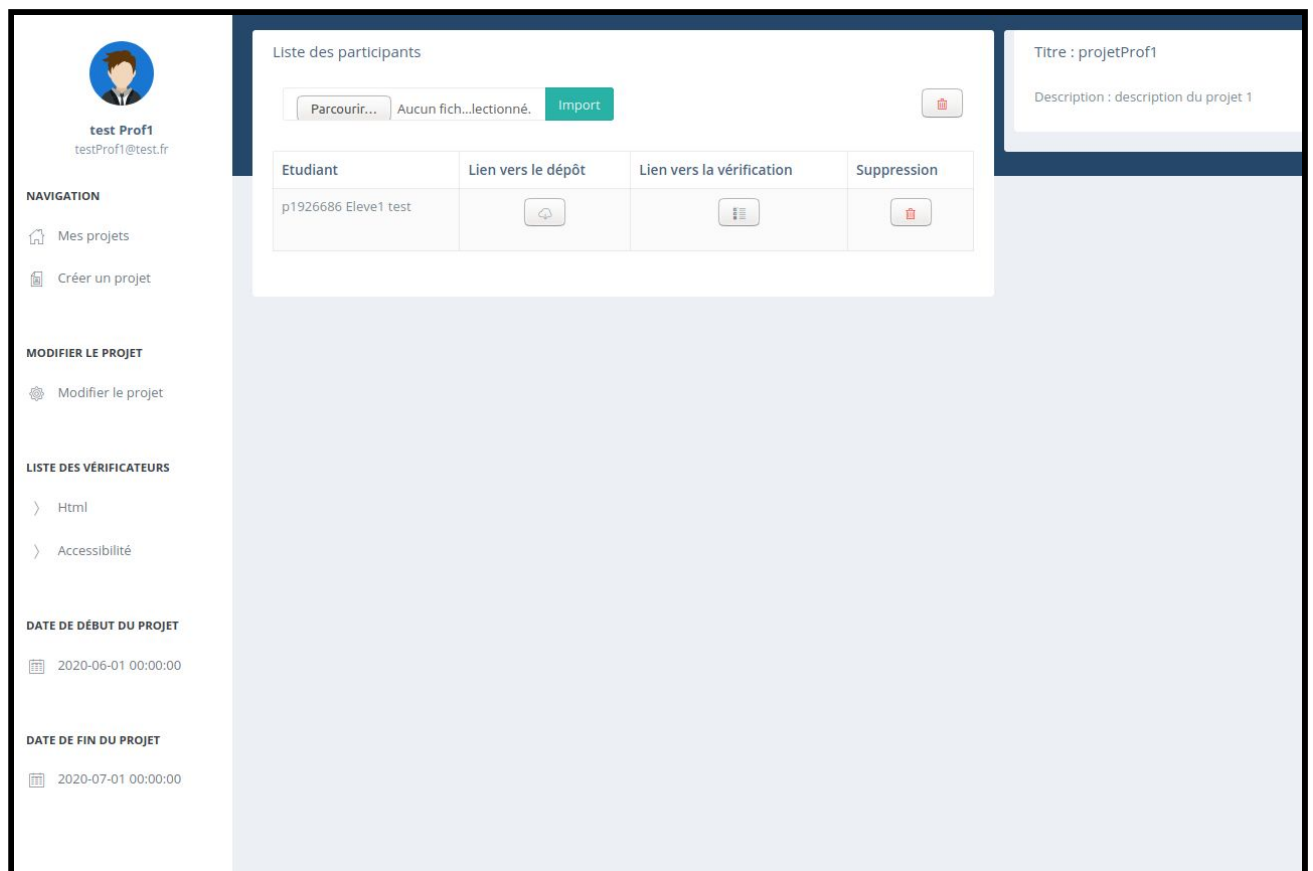


Figure 6 : Interface enseignant: Description du projet

3.5.2 Créer un projet

L'enseignant a également la possibilité de créer de nouveaux projets. Pour cela, il lui suffit de cliquer sur le bouton « Créer un projet » dans le menu de « Navigation ». Il pourra ainsi renseigner les différents détails du projet.

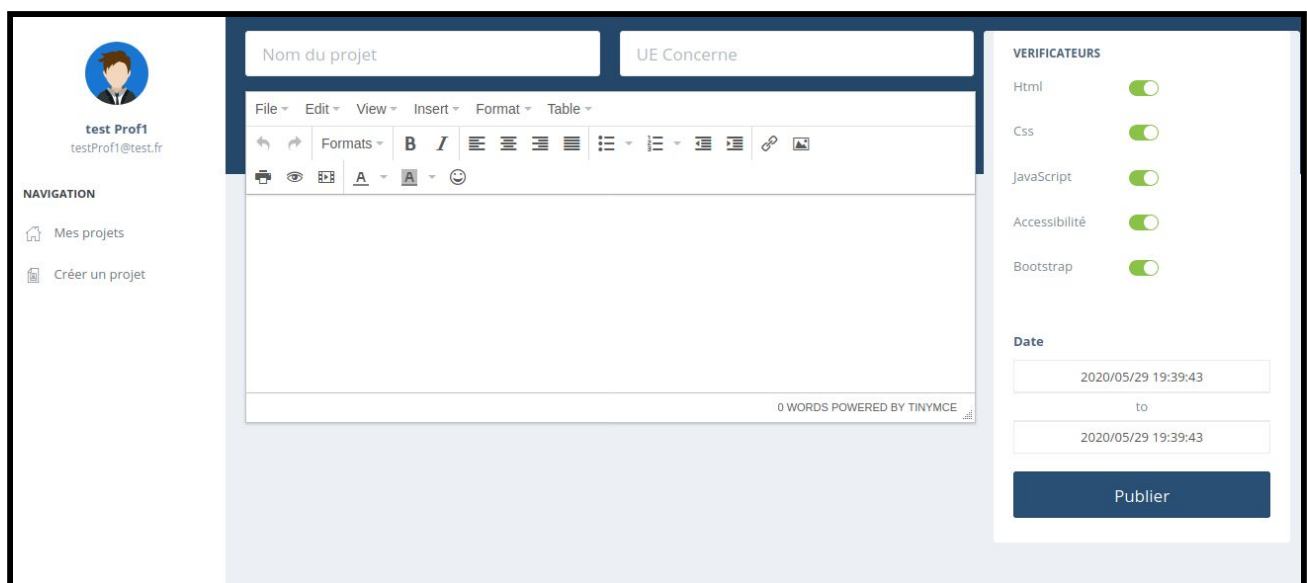


Figure 7 : Interface enseignant: Création du projet

4 Conclusion

Notre application remplit la fonction qui lui était demandée à savoir obtenir en un seul endroit les résultats des différents vérificateurs HTML, CSS, Javascript et Accessibilité.

On pourrait imaginer de futures mises à jour permettant d'ajouter de nouveaux vérificateurs syntaxiques pour d'autres langages, permettant ainsi à l'application d'être utilisable sur un plus grand nombre d'UE. Malheureusement, bien que proposé par nos enseignants, nous n'avons pas eu le temps de mettre cette application en service cette année. Nous privant ainsi d'un retour utilisateur intéressant qui aurait pu aiguiller notre développement sur l'amélioration des différentes interfaces.

Ce projet est notre premier projet universitaire qui peut avoir une portée utilitaire, autre qu'avoir une simple note. La difficulté de ce projet ne reposait pas sur les technologies utilisées, car nous avons eu la possibilité de choisir ces dernières. Ni sur la prise en main des différents vérificateurs. Mais plutôt sur un souci du détail, nous poussant à vouloir produire une application où on serait fier qu'elle puisse être utilisée dans le futur par d'autres étudiants.