



Portfolio avec VueJS

Par Luc PERARD


CEF janvier 2024 :
Développeur Web et Web
mobile

Projet bilan – juin 2024



TABLE DES MATIERES

Résumé	1
Introduction et préambule	2
Démarche de réalisation du projet.....	2
Conception du projet.....	4
Maquette de principe.....	4
Charte graphique	6
Exigences du projet	8
Développement avec VueJS	10
Vérification et livraison du projet	14
Synthèse du développement	15
Conclusion du projet	15



RESUME

Pour ce projet de portfolio personnel, l'enjeu est d'exploiter le cadre de travail VueJS ainsi que deux environnements professionnels : VSCode pour le codage et GitHub pour les versions du code.

Le « portfolio de PerLucCo », un site avec une structure de pages dynamiques, développe les fonctionnalités d'accueil, de présentation de PerLucCo, de description de travaux et de prise de contact. Le site gère les erreurs d'adressage des pages avec une « Page 404 », et affiche une « page modale » pour chaque travail sélectionné par le visiteur. Le design sobre et simple est choisi autour des couleurs du logo de VueJS.

D'autre part, ce projet met en œuvre une gestion des tâches et son calendrier en utilisant les applicatifs GanttProject (planification) et Trello (gestion des tâches). Le développement par étapes gérées avec GitHub permet d'accéder à quatre versions : trois prototypes et une version finale opérationnelle.

Ainsi, pour mener ce projet à bien, une complète formation complémentaire sur le cadre de travail VueJS a été menée avec le site W3Schools. Les différents points techniques bloquant ont fait l'objet de recherches sur les sites français et anglophones, permettant la résolution de tous les sujets identifiés. Disponible sur GitHub, la réalisation menée en quatorze semaines – dont huit de formation à VueJS et quatre de développement - présente le site opérationnel du « portfolio de PerLucCo » en VueJS.

Paramètres de synthèse du résumé :

- 229 mots
- 1199 caractères (sans espace)
- 20 lignes

INTRODUCTION ET PREAMBULE

Ce projet consiste à développer le site d'un portfolio personnel en utilisant le framework VueJS version 3. Le contenu de ce site est fictif et n'a vocation qu'à animer les fonctionnalités demandées pour un portfolio.

Ce portfolio concerne « PerLucCo », l'autoentrepreneur que je suis.

DEMARCHE DE REALISATION DU PROJET

Pour réaliser ce projet, la démarche a été menée en utilisant un outil :

- de planification, l'appliquatif « GanttProject » ;
- de gestion des tâches, la plateforme « Trello » ;
- de diagramme, l'appliquatif « draw.io » ;
- de codage, l'environnement « Visual Studio Code » ;
- de gestion et de version de code, la plateforme « GitHub » ;
- de rédaction de rapport, l'éditeur de texte « Microsoft Word » ;
- de présentation, l'éditeur de vue « Microsoft PowerPoint ».

La planification s'est basée sur un délai initial de 4 semaines de réalisation, du 25 mars au 19 avril 2024. La figure présente cette version initiale du calendrier des travaux.

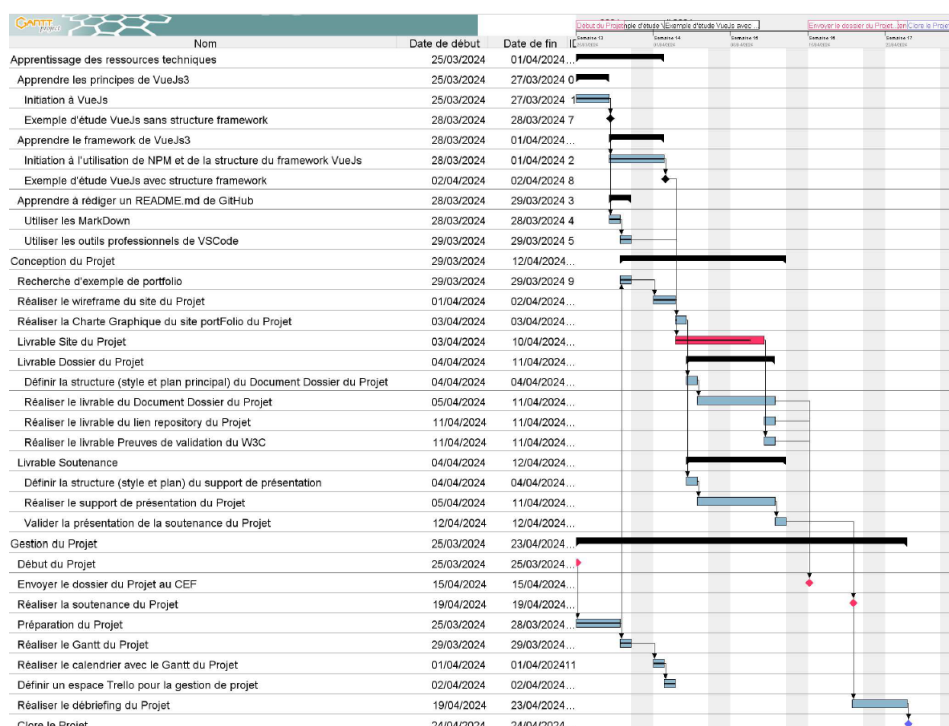


Figure 1 : planning initial du projet de portfolio

Toutefois, il est apparu une difficulté majeure : la formation pour exploiter le framework VueJS était insuffisante pour débiter le codage du projet dans l'environnement de développement VSCode. Le délai de 4 jours prévus dans le planning au titre de « apprentissage des ressources techniques » devait être totalement revu avec une phase préalable de *primo* formation à VueJS.

Dans les faits, une neutralisation de neuf (9) semaines a été faite pour reprendre complètement en huit (8) semaines la formation de VueJS version 3 (soient 80 heures) sur le site anglophone www.w3schools.com, auxquelles s'ajoute une (1) semaine de congés. La date effective de début du projet de portfolio en VueJS était donc le 27 mai 2024. La date de présentation devenait le 24 juin 2024.

Pour le suivi des tâches, la plateforme Trello permet de visualiser les différentes tâches ouvertes, à faire, en cours, bloquée, terminée comme le présente la figure suivante.

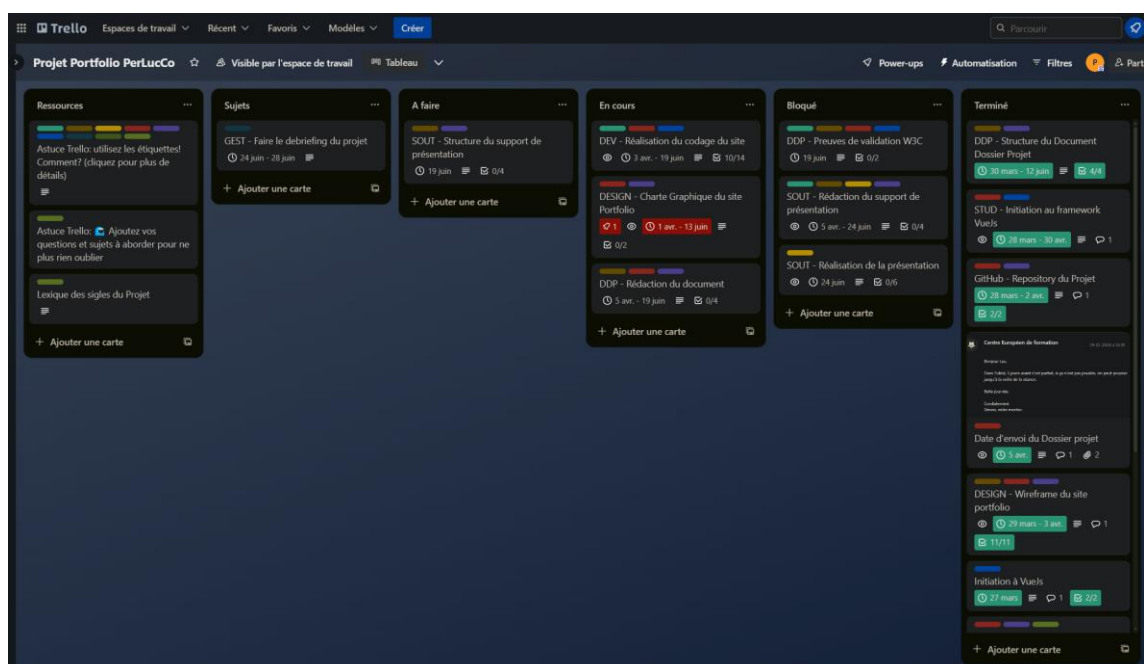


Figure 2 : gestion des tâches sur la plateforme Trello (www.trello.com)

L'archivage des différentes versions du code du projet est réalisé sur la plateforme GitHub (www.github.com) dans l'espace public de MonLucCo sous le nom de dépôt « CEF_Portfolio-with-VueJs_Test-version-01 ». La figure suivante présente un avancement intermédiaire du projet à l'issue d'une troisième version du prototype du site archivée dans le dossier « portfolio ».

Pour chaque étape de développement du code dans l'environnement VSCode, une demande (issue sur GitHub) a été établie tout en étant associée à une branche de développement spécifique. La synchronisation et la coordination des travaux entre GitHub et Git de VSCode se trouvent alors facilitée lors des opérations de fusion des travaux dans le référentiel de GitHub.

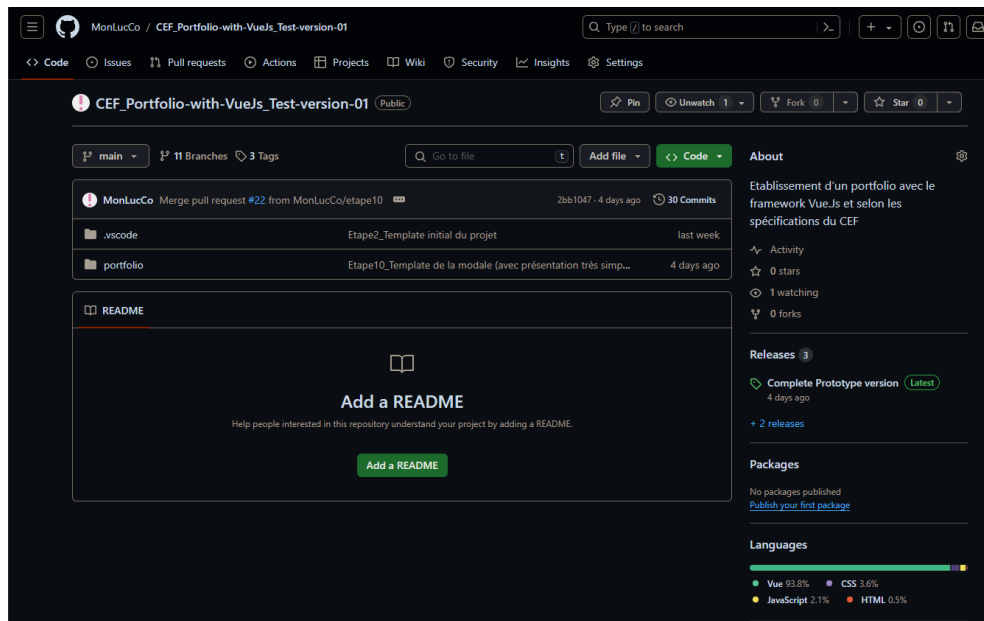


Figure 3 : dépôt (version intermédiaire) du projet dans GitHub
(www.github.com)

Avant de débiter le codage du site, une maquette du projet a été établie avec l'applcatif « draw.io ». Ce sujet étant une étape de la conception, il est présenté dans le chapitre suivant.

CONCEPTION DU PROJET

La conception du site du portfolio de PerLucCo a été menée en cinq phases :

- 1) l'élaboration d'une maquette de principe ;
- 2) l'identification d'une charte graphique basé sur le logo de VueJS ;
- 3) l'élaboration de « composantes » dans le code de VueJS ;
- 4) l'élaboration de contenus fictifs pour animer le site ;
- 5) la validation du code par fonctionnalité.

MAQUETTE DE PRINCIPE

La maquette de principe a été élaboré avec l'éditeur de diagramme « draw.io ». Elle a pour but d'apporter une vue rapide du principe de fonctionnement sans pour autant représenter le graphique de la solution finale. Elle a été réalisée à partir de la description fonctionnelle exprimée dans le cahier des charges (le sujet du devoir du projet Bilan).

Cette maquette a été réalisée avant d’engager la formation complémentaire de VueJS, soient 9 semaines avant le début du codage du site. Elle se compose de six graphiques qui représentent dans le tableau ci-dessous les différentes pages du site : accueil, présentation, travaux, contact, modale, erreur 404.

Pour préparer l’analyse de décomposition en composantes, les différentes pages de la maquette identifient des modules graphiques visualisés par des cadres en pointillés et nommés pour servir de guide dans la structure du code développé.

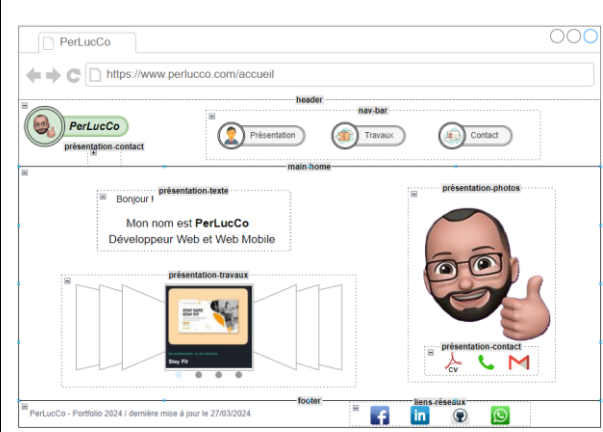
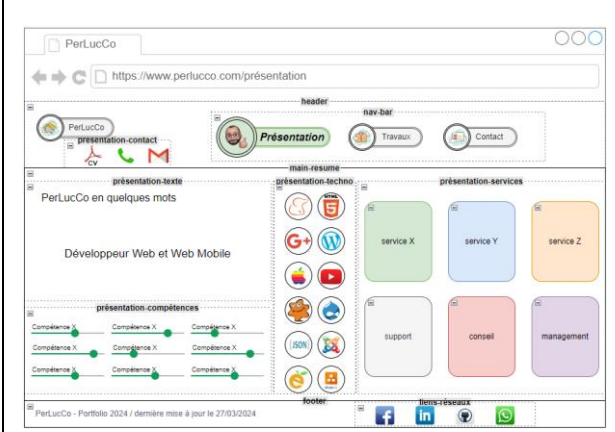
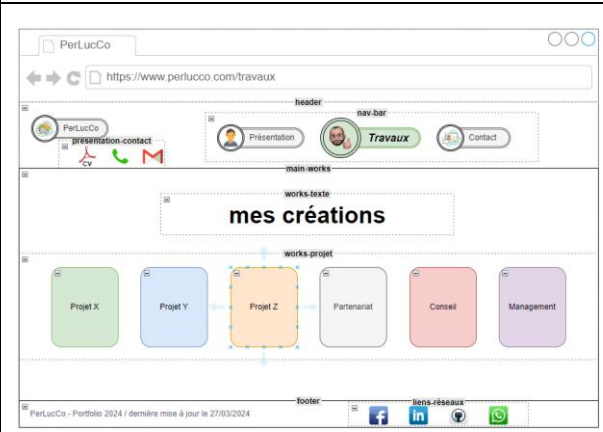
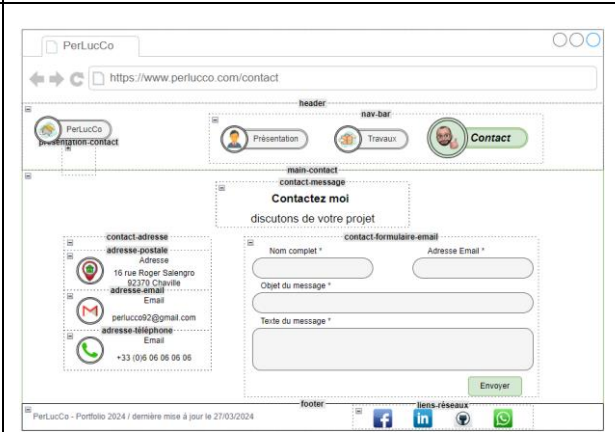
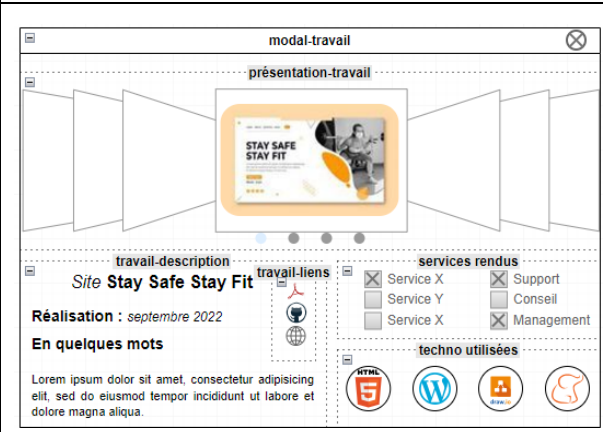

	
Page Accueil	Page Présentation
	
Page Travaux	Page Contact
	
Page modale	Page Erreur 404

Tableau 1 : maquette de principe des différentes pages du site

CHARTRE GRAPHIQUE

La charte graphique choisie pour le site du portfolio de PerLucCo se veut simple et sobre.

La palette de couleur se base sur les couleurs du logo de VueJS, A la fois, car ces couleurs représentent bien la personnalité de PerLucCo. Mais aussi pour faire référence à l'origine du code, le cadre de travail VueJS.

Les polices de caractères exploitent les polices du thème de VueJS qui sont sobres et correspondent à l'esprit de ce portfolio.

Les icônes choisies sont regroupées dans le tableau ci-dessous. Elles cherchent à être représentatives, simples ou décalées, en lien avec VueJS.






















 (32x32) favicon.ico	 (128x128) Icône_pdf_01.png	 (128x128) Icône_pdf_02.png
 (128x128) Icône_Email_08.png	 (128x128) Icône_tel_03.png	 (128x128) Icône_tel_05.png
 (128x128) Social_GitHub.png	 (128x128) Social_LinkedIn.png	 (svg) logo.svg
 (32x32) Icône_Fermeture_01b.ico	 (128x128) Icône_Email_02.png	 (220x220) Social_GitHub_03.png
 (169x169) Icône_Drupal_01.png	 (481x481) Icône_RPA_03.png	 (169x169) Icône_VBA_01.png
 (200x200) Icône_html5.png	 (720x720) Icône_css3_03.png	 (171x171) Icône_JS_03.png
 (60x80) Icône_Position_01.png	 (5000x5000) Icône_Localisation_02c.png	 (128x128) Icône_tel_01.png

Tableau 2 : icônes du site

Les images choisies sont des illustrations humoristiques ou symboliques qui sont regroupées dans le tableau ci-dessous pour la description des services et des travaux.












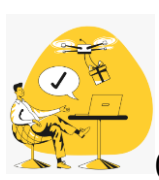
 (217x233) Avatar8perLucCo.png	 (1292x1535) CV-de-Luc-Pérard_capture.jpeg	 (382x367) Avatar_About_24.png
 (392x339) Avatar_About_31.png	 478x375) Avatar_About_32.png	 (426x425) Avatar_About_37.png
 (367x431) Avatar_About_16.png	 (430x417) Avatar_About_17.png	 (406x396) Avatar_About_22.png
 (342x391) Avatar_About_25.png	 (372x408) Avatar_About_26.png	 (388x416) Avatar_About_27.png

Tableau 3 : illustrations et avatars du site

Pour la représentation chiffrée (valeur textuelle affichée ou non) d'un sujet, deux possibilités graphiques de visualisation sont possibles, soit sous la forme :

- d'une jauge linéaire horizontale ;
- d'une jauge circulaire associée à une icône.

Le tableau ci-dessous présente des exemples de ces représentations de jauges.


 Jauge : linéaire Valeur : 30% affichée	 Jauge : circulaire Valeur : 60% affichée	 Jauge : circulaire Valeur : 70% non affichée
--	--	--

Tableau 4 : exemples de jauges utilisées pour le site

EXIGENCES DU PROJET

Le tableau ci-dessous exprime avec une classification par thème les exigences techniques (ET) et fonctionnelles (EF) du projet qui sont issues du document transmis par le CEF pour le projet « créer un portfolio avec Vue.JS ».

Exigence	Thème	Libellé
ET-01-01	Dossier de projet	Contient une page de présentation
ET-01-02	Dossier de projet	Contient un sommaire
ET-01-03	Dossier de projet	Contient un résumé en français de : <ul style="list-style-type: none">- environ 20 lignes- entre 200 à 250 mots- environ 1200 caractères espaces non compris
ET-01-04	Dossier de projet	Conception du site via un lien vers le repository GitHub
ET-01-05	Dossier de projet	Un fichier README.md à la racine du repository avec les prérequis du projet ainsi que les instructions d'installation et de lancement
ET-01-06	Dossier de projet	Fournir les captures d'écrans des résultats des validateurs W3C pour le code HTML et CSS
ET-02-01	Mission	Créer un site avec Vue.JS
ET-02-02	Mission	Le site doit être ergonomique
ET-02-03	Mission	Le site doit être accessible
ET-02-04	Mission	La sémantique du site doit être correct (hors des erreurs de code inhérentes au code métier de VueJS)
ET-03-01	Technologies	Utiliser le framework VueJS v3, HTML5, CSS3 (flexbox) et JavaScript
ET-03-02	Technologies	Coder avec Visual Studio Code (code propre, commenté et indenté)
ET-03-03	Technologies	Utiliser Git et GitHub pour le versioning
ET-04-01	Identité graphique	Choisir une palette de couleurs, une police et un logo

Exigence	Thème	Libellé
EF-01-01	Contenu des pages	Le footer est identique sur toutes les pages
EF-01-02	Contenu des pages	Le header est présent sur toutes les pages
EF-01-03	Contenu des pages	Mettre un titre et une description à la page à destination des moteurs de recherche
EF-01-04	Contenu des pages	La page 404 devra apparaître dès lors qu'on essaie d'accéder à autre chose que les pages du site
EF-02-01	Page 404	La page 404 contient une image, un texte « 404 »
EF-02-02	Page 404	La page 404 contient un retour vers la page d'accueil
EF-03-01	Footer de page	Le footer contient des icônes avec des liens vers les réseaux sociaux. Les liens devront s'ouvrir dans un nouvel onglet
EF-03-02	Footer de page	Le footer contient la date de dernière mise à jour du site
EF-04-01	Header de page	Le header contient un logo ou une photo qui ramène en haut de la page d'accueil lorsqu'on clique dessus
EF-04-02	Header de page	Le header contient un menu avec des liens d'ancrage vers la présentation, les créations et le formulaire de contact
EF-05-01	Page d'accueil	La page d'accueil contient nom, prénom et une courte présentation
EF-05-02	Page d'accueil	La page d'accueil contient les créations avec chaque fois une image et un titre. La disposition est libre.
EF-05-03	Page d'accueil	La page d'accueil contient un formulaire de contact (nom/prénom, objet et message) qui s'envoie à l'adresse email définie en variable d'environnement
EF-06-01	Page modale	La page modale s'ouvrira avec plus de photos et détails au clic sur la création à montrer

Exigence	Thème	Libellé
EF-06-02	Page modale	La page modale se remplit avec les données de la création à montrer : titre, date de création, technologies utilisées, liens au besoin (pour visiter le site ou le fichier PDF, vers le repository GitHub)
EF-06-03	Page modale	La page modale se fermera lorsqu'on clique sur le bouton de fermeture ou en dehors de la fenêtre modale
EF-06-04	Page modale	La page modale devra rester fixe sur la page même si l'on scrolle
EF-07-01	Effets et animations	Au survol d'une création, une ombre apparaît en bas et à droite
EF-07-02	Effets et animations	L'élément de menu actif devra être souligné

Tableau 5 : exigences techniques (ET) et fonctionnelles (EF)

Ces 33 exigences (dont 30 liées au développement et 3 liées au rapport) sont exploitées dans les chapitres suivants pour les travaux de développement et sont ensuite évaluées lors de la vérification pour livraison qui conduit à la dernière version du site, dite « version finale opérationnelle ».

DEVELOPPEMENT AVEC VUEJS

Pour développer le projet dans VSCode, il est nécessaire de mener deux actions techniques qui visent à avoir un environnement adapté au codage avec VueJS.

D'une part, il s'agit d'installer (ou de vérifier l'installation) les (des) extensions de VSCode concernant notamment :

- le versioning du code (GitHub Codespaces, Git Graph) ;
- la syntaxe du code (Vitest, Vetur, ESLint, es6-string-html, Error Lens) ;
- la syntaxe Markdown (Markdown All in One, Markdown preview GitHub Styling, Markdown Preview Enhanced, markdownlint);
- la visualisation de fichier PDF (vscode-pdf).

D'autre part, il s'agit d'installer le « framework VueJS » avec le terminal de VSCode pour obtenir un cadre de travail permettant de débiter le codage. Plusieurs commandes sont possibles. Dans le cadre du projet cette commande est « **create npm vue@latest** » pour :

- créer un projet « portfolio » (nom du dossier du code VueJS) ;
- choisir « router » et « DevTool » (options retenues pour faciliter le codage).

L'environnement de travail dispose alors des codes du « framework VueJS » dans la version initiale proposée dans le « repository GitHub ». Les différentes étapes du développement commencent par supprimer tous les fichiers inutiles que l'installation a mis en place pour permettre une visualisation d'un site informatif au look VueJS.

Le développement par composantes VueJS s'est déroulée en étapes majeures complétées d'étapes intermédiaires pour stabiliser et vérifier progressivement l'avancement du projet. Les étapes majeures ont conduit à la réalisation de 3 versions opérationnelle de prototype du site, suivies d'une étape majeure et finale pour la réalisation de la version finale et opérationnelle du site.

Le tableau ci-dessous récapitule les étapes majeures du développement pour les 30 exigences relatives au développement.

Etape	Version	Description du résultat
A	(v0.0.0)	<p>Modèle initial du projet en VueJS : une page dynamique avec une zone « header », une zone « footer » et une zone « main ».</p> <p>Architecture du code pour permettre la satisfaction des thèmes d'exigences ET-01 à ET-04 et EF-01 à EF-05.</p>
B	V0.1.0	<p>Premier prototype opérationnel : une page dynamique avec une navigation opérationnelle incluant la gestion des erreurs d'adresse (page 404).</p> <p>Utilisation de la palette de couleurs et des icônes.</p> <p>Pas de contenu opérationnel, seulement des textes de validation fonctionnelle pour la navigation.</p> <p>Le contenu de la page 404 est dans la version opérationnelle.</p> <p>Architecte du code et codage pour satisfaire les exigences ET-01-04, ET-02-01, ET-02-02, ET-02-03, ET-02-04, ET-03-01, ET-03-02, ET-03-03, ET-04-01, EF-01-01, EF-01-02, EF-01-04, EF-02-01, EF-02-02, EF-03-01, EF-03-02, EF-04-01 et EF-04-02.</p> <p>Couverture des exigences du développement : 18 sur 30, soit 60%.</p>
C	V0.2.0	<p>Second prototype opérationnel : une page dynamique du premier prototype avec l'ajout du contenu des pages accueil, présentation, travaux et contact.</p> <p>La page modale est accessible dans une version préliminaire de validation de l'ouverture et de la fermeture.</p> <p>Architecture du code et codage pour satisfaire les exigences du prototype v0.1.0 (pas de régression) et les exigences EF-05-01, EF-05-02, EF-05-03, EF-06-04 et EF-07-01.</p> <p>Couverture des exigences du développement : 23 sur 30, soit 77%.</p>

Etape	Version	Description du résultat
D	V0.3.0	<p>Troisième prototype opérationnel : une page dynamique du second prototype avec l'ajout de la page modale pour tous les travaux présentés.</p> <p>Architecture du code pour satisfaire les exigences du prototype v0.2.0 (pas de régression) et les exigences EF-06-01, EF-06-02, EF-06-03.</p> <p>Couverture des exigences du développement : 26 sur 30, soit 87%.</p>
E	V1.0.0	<p>Version opérationnelle : une page dynamique du troisième prototype avec l'ajout de la description du site à destination des robots, du design d'animation du menu, le README.md et les corrections des erreurs de la validation W3C.</p> <p>Architecture du code pour satisfaire les exigences du prototype v0.3.0 (pas de régression) et les exigences ET-01-05, ET-01-06, EF-01-03 et EF-07-01.</p> <p>Couverture des exigences du développement : 30 sur 30, soit 100%.</p>

Tableau 6 : différentes versions du développement en VueJS

Pour la réalisation des développements, des règles de codage ont été définies pour :

- le positionnement des fichiers dans l'arborescence du dossier du projet ;
- le nommage des fichiers de codes (les composantes de VuesJS)
- l'emploi de la méthode « Options API » pour la rédaction des composantes ;
- l'utilisation minimale de bibliothèques d'extension de VueJS ;
- la suppression avant livraison des composantes développées et non utilisées.

Le tableau suivant présente une description succincte des règles utilisées pour le codage du projet.

Thème	Description de la règle
Position des fichiers dans l'arborescences des dossiers du projet (dossier « portfolio »)	<ul style="list-style-type: none"> - fichiers images dans « /src/assets/icones » ou « /src/assets/images » ou « /public ». - fichiers « .vue » dans « /src/components » ou « /src/views »

Thème	Description de la règle
Nommage des fichiers de codes (composantes VueJS)	<ul style="list-style-type: none"> - « ...View.vue » pour un composant décrivant une vue de premier niveau de la page (position dans « /src/views »). - « Site....vue » pour un composant décrivant un conteneur de premier niveau de la page. - « Comp....vue » pour un composant décrivant un conteneur de second niveau et plus de la page. - « CompModal....vue » pour un composant d'un conteneur de premier niveau de page « modale ». - « Elem....vue » pour un composant élémentaire.
Emploi de la méthode « API Options »	- Selon les conseils indiqués dans la formation www.w3schools.com .
Utilisation minimale de bibliothèques d'extension de VueJS	<ul style="list-style-type: none"> - Utilisation du « framework VueJS3 » - Utilisation de l'extension « Vite3 » - Utilisation de l'extension « router » - Utilisation de l'extension « DevTool7 » <p>Les extensions suivantes ont été analysées et non retenues, car réalisation possible des exigences du projet sans leur utilisation :</p> <ul style="list-style-type: none"> - Extension « Vuetify » - Extension « tailwindcss » - Extension « VueMeta ».
Suppression avant livraison des composantes développées et non utilisées	<ul style="list-style-type: none"> - Chaque composante développée et archivée dans GitHub ne sera pas supprimée pour permettre une visualisation du code. - La suppression d'un code non utilisé peut être faite avant la livraison dans une archive spécifique.

Tableau 7 : règles pour le développement et codage du projet

Lors du développement, trois sujets ont fait l'objet d'une analyse et d'une recherche sur les sites de référence avant le codage en VueJS :

- L'utilisation de l'extension « router » pour :
 - o la gestion des ancrages du menu dans la page dynamique ;
 - o la gestion d'une page d'erreur d'adressage (page 404) ;
- La gestion de l'événement « clic » pour une « page modale ».
- La description des « meta » données à destination des moteurs de recherche.

VERIFICATION ET LIVRAISON DU PROJET

Avant de livrer le projet, aucune suppression des composantes (et code) inutiles n'a été réalisée afin de pouvoir visualiser le résultat de l'ensemble des travaux de codage menés pour le projet de Portfolio de perLucCo en VueJS.

La vérification consiste à produire une version du code et à évaluer la qualité du code avec le validateur du W3C. Toutefois, l'obtention de ce code nécessite des manipulations techniques.

En effet, le serveur « Vite » associé à VueJS produisant un code d'exécution différent de celui codé dans VSCode, l'analyse du code HTML est faite à partir du code obtenu dans le DOM de l'inspecteur du navigateur (Edge pour le projet).

Le résultat obtenu fait apparaître 29 avertissements (28 liés au module « Vite » et 1 associé à « DevTool »). La figure ci-dessous présente ce résultat, ainsi que le filtrage sur les avertissements activés pour constater qu'il n'y a pas d'autre erreur.

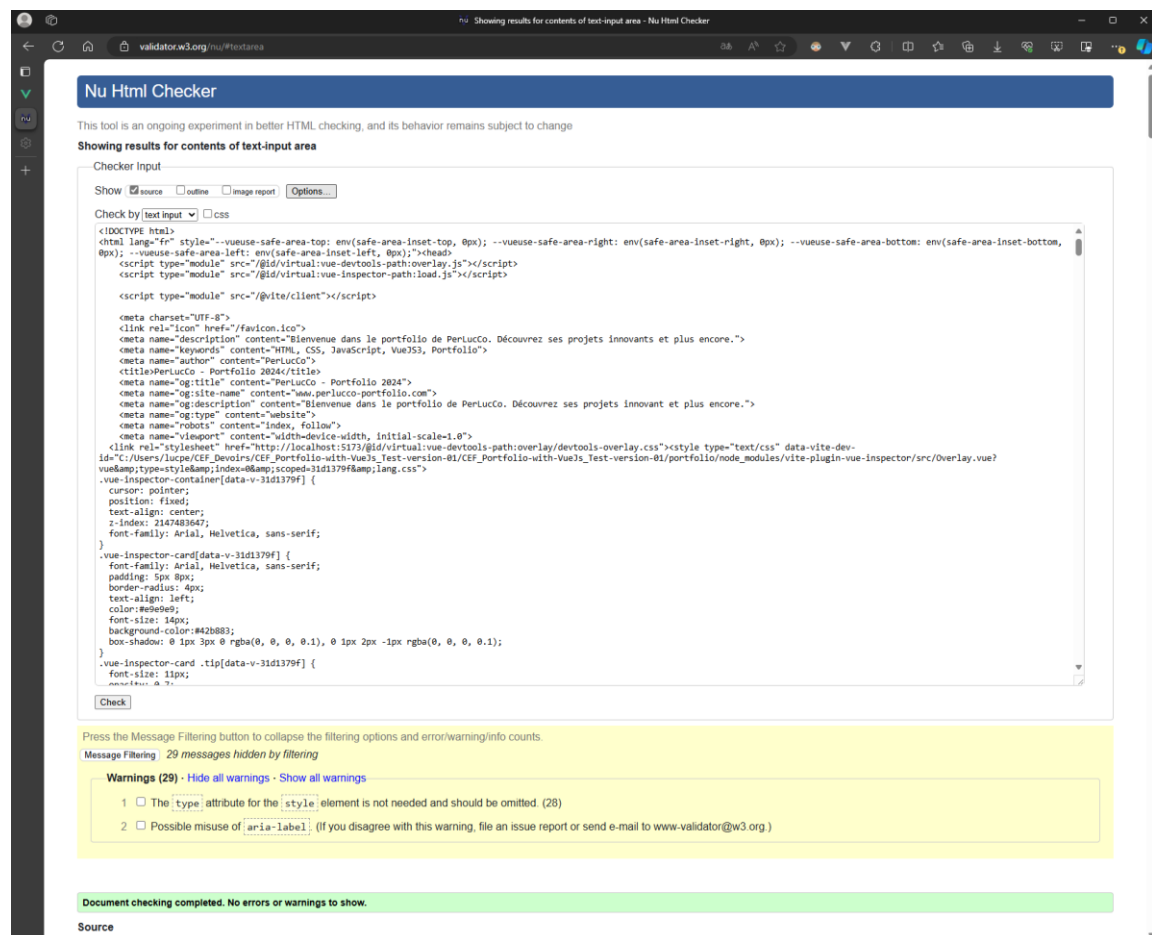


Figure 4 : résultat de la validation W3C du code HTML

Le projet se trouve disponible à l'adresse :

https://github.com/MonLucCo/CEF_Portfolio-with-VueJs_Test-version-01

SYNTHESE DU DEVELOPPEMENT

Le développement a permis de coder le projet de portfolio de PerLucCo en VueJS3 et avec un minimum d'extension (VueRouter et DevTool).

La formation sur « Vue » à partir du site www.w3schools.com a permis de traiter tous les sujets fonctionnels demandés. Quelques recherches complémentaires, notamment sur l'emploi de l'extension VueRouter, ont été nécessaires avant d'engager le codage.

La réalisation livrée d'une version opérationnelle en VueJS répond à l'ensemble des exigences techniques et fonctionnelles exprimées. L'emploi de GitHub a permis de tracer l'ensemble des étapes de codage.

CONCLUSION DU PROJET

Ce projet a permis d'apprendre les principes de fonctionnement et de codage d'une page dynamique avec le « framework » Vue.JS version 3.

La mise en place d'une démarche structurée avec une gestion des tâches et un planning a permis d'identifier rapidement le déroulement du projet. Toutefois, l'aléa dû à un manque de formation technique avec VueJS a nécessité une replanification et un report de la date de livraison. Ceci a mis en évidence la nécessité de maîtriser les outils et le langage de codage avant d'engager un tel projet.

D'autre part, la réalisation d'une maquette du site en préalable à tous travaux de développement s'est avérée très importante. Avant la concrétisation par codage en VueJS, cette maquette a servi de guide pour :

- structurer les étapes de la démarche de codage ;
- illustrer un possible fonctionnement du site.

Ce projet m'a permis de consolider mes compétences dans le développement d'un site composé d'une page web dynamique, d'utiliser l'outil de gestion de version de code source « GitHub » et de mettre en œuvre toutes les étapes du codage jusqu'à la livraison d'un produit spécifié.