**Présentation du Projet**

**« PORTEFOLIO AVEC VUR .JS »**

Table des matières

[Sommaire 1](#_Toc165056564)

[Présentation du projet 2](#_Toc165056565)

[Résumé du projet 3](#_Toc165056566)

[Conception du site 4](#_Toc165056567)

[Installation de l’environnement 4](#_Toc165056568)

[Développement 5](#_Toc165056569)

[Le Header : 5](#_Toc165056570)

[Le Footer : 5](#_Toc165056571)

[Page d'Accueil : 5](#_Toc165056572)

[Modal Dynamique : 6](#_Toc165056573)

[Page 404 : 6](#_Toc165056574)

[Déploiement 6](#_Toc165056575)

[Livrables 6](#_Toc165056576)

[Le fichier README.md 6](#_Toc165056577)

[I Portfolio - Développeur Full Stack 7](#_Toc165056578)

[II Prérequis 7](#_Toc165056579)

[III Installation 7](#_Toc165056580)

[Configuration 7](#_Toc165056581)

[IV Lancement 7](#_Toc165056582)

[Contribution 7](#_Toc165056583)

[Auteurs 8](#_Toc165056584)

[Licence 8](#_Toc165056585)

[Capture d’écrans validateur W3C 8](#_Toc165056586)

[Conclusion 8](#_Toc165056587)

Ce projet consiste a réaliser un porte folio avec la technologie vue.js.

Afin de pouvoir montrer nos compétences et accroitre notre visibilité au reste des donneur d’ordre. La création d’un portfolio est l’outil idéal.

Pour partager nos compétences de façon créative nous réaliserons ce projet port folio avec vue.js en y incluant nos premières réalisations. Des choix ont été fait tant sur le fond (contenue rédactionnel) que sur la forme charte graphique. Une attention particulière a été porté a l’ergonomie (expérience utilisateur) et accessibilité pour tous les publique (Web content accessibility guidelines) a la sémantique (SEO). Le code HTML et CSS développer répond aux norme édicté par la W3C.

Vous y retrouverez trois projet déjà réalisé : éle cv écrit en HTML et CSSé, éle Pdf cahier des charges » et le « un module JavaScript pour dynamiser une page web ».

Le portfolio permet de mettre en avant vos compétences que ce soit après d’un employeur ou auprès d’un client. Il recense vos projets et créations. CV en ligne ou portfolio ? Le CV présente votre parcours. On pourrait dire qu’il est « statique c’est une photo prise a un instant de votre parcours. Alors que le porte folio pourrai être qualifier de dynamique. Il présente vos réalisations de façon concrète et vos projets.

Afin de démontrer un savoir faire thecnique les technologie utilisées sont :

-• Le framework Front-End VueJS v3.

• HTML5, CSS3 (Flexbox), JS.

• Git et GitHub pour le versioning.

L’IDE de d »veloppement utiliser est Visual Studio Code.

# **Présentation du projet**

Ce projet consiste à réaliser un portfolio avec la technologie Vue.js. Dans le but de mettre en valeur nos compétences et d'accroître notre visibilité auprès des donneurs d'ordre, la création d’un portfolio s'avère être l’outil idéal. Pour partager nos compétences de manière créative, nous réaliserons ce projet de portfolio avec Vue.js en y incluant nos premières réalisations. Des choix ont été faits tant sur le fond (contenu rédactionnel) que sur la forme (charte graphique). Une attention particulière a été portée à l’ergonomie (expérience utilisateur), à l’accessibilité pour tous les publics (Web Content Accessibility Guidelines) et à la sémantique (SEO). Le code HTML et CSS développé répond aux normes édictées par la W3C. Vous y retrouverez trois projets déjà réalisés : un CV écrit en HTML et CSS, un cahier des charges au format PDF et un module JavaScript pour dynamiser une page web.

Le portfolio permet de mettre en avant vos compétences que ce soit auprès d’un employeur ou d’un client. Il recense vos projets et créations. CV en ligne ou portfolio ? Le CV présente votre parcours. On pourrait dire qu’il est statique, c’est une photo prise à un instant de votre parcours. Alors que le portfolio pourrait être qualifié de dynamique. Il présente vos réalisations de façon concrète et vos projets. Afin de démontrer un savoir-faire technique, les technologies utilisées sont :

* Le Framework Front-End VueJS v3.
* HTML5, CSS3 (Flexbox), JS.
* Git et GitHub pour le versioning. L’IDE de développement utilisé est Visual Studio Code et ses extensions.

L’application est une Single Page Application (SPA) qui utilise un header avec un logo cliquable renvoyant à la page d’accueil et une barre de navigation utilisant un système de routage vers les différentes parties du site. Un footer présentant des icônes des réseaux ouvrant les liens dans une nouvelle page, ainsi que la date de la dernière mise à jour. Sur la page d’accueil, vous retrouverez différentes sections comprenant notamment une présentation de l’entreprise fictive que je viens de rejoindre, ainsi qu'une présentation sous forme de « card » de mes réalisations. Celles-ci, au clic, ouvriront vers une vue détaillée du projet sélectionné.

# **Résumé du projet**

Ce projet vise à concevoir un site web dynamique et fonctionnel à l'aide de VueJs pour promouvoir mes compétences et créations. Voici un aperçu des principales fonctionnalités du site :

**Le Header :**

* Présent sur toutes les pages, le header comporte un logo cliquable ramenant en haut de la page d'accueil pour une navigation intuitive.
* Il intègre un menu avec des liens d'ancrage vers différentes sections telles que la présentation, les créations et le formulaire de contact. L'élément de menu actif est mis en évidence pour guider les utilisateurs.

**Le Footer :**

* Présent sur toutes les pages, le footer propose des icônes menant vers mes réseaux sociaux (GitHub, LinkedIn), Ces liens s'ouvrent dans un nouvel onglet pour une expérience utilisateur optimale.
* La date de la dernière mise à jour du site est affichée pour maintenir les visiteurs informés de la pertinence des informations présentées.

**Page d'Accueil :**

* La page d'accueil est ma vitrine personnelle, comprenant mes nom, prénom et une brève présentation pour établir une première connexion avec les visiteurs.
* Elle met en avant mes créations à travers des images et des titres (card) cliquables, permettant ainsi d'accéder à une modal détaillée. Des effets visuels subtils, tels qu'une ombre au survol, sont intégrés pour améliorer l’expérience des visiteurs.
* Un formulaire de contact est inclus, permettant aux visiteurs de vous contacter en toute simplicité en envoyant les données à une adresse e-mail définie en variable d'environnement.

**Modal Dynamique :**

* La modal dynamique présente en détail les informations de chaque création, telles que le titre, la date de création, les technologies utilisées, ainsi qu’un lien vers le document ou le repository GitHub.
* Elle offre une expérience utilisateur fluide en se fermant par un simple clic sur le bouton de fermeture ou en dehors, tout en restant fixe sur la page même en scrollant.

**Page 404 :**

* La page 404 est une solution de secours en cas d'erreur de navigation, affichant une image et le texte "404" pour signaler à l'utilisateur qu'une ressource n'a pas été trouvée.
* Elle s'affiche automatiquement lorsque l'utilisateur tente d'accéder à autre chose que la page d'accueil, garantissant ainsi une expérience de navigation cohérente.

# **Conception du site**

## Installation de l’environnement

Pour démarrer le développement de ce projet j’ai du installer different outils et composant :

Installation de Node.js pour pouvoir utiliser le gestionnaire de paquet npm (Node Package Manager

Installation de Vue CLI (Interface de ligne de commande Vue). Vue CLI est un outil de développement pour construire des applications Vue.js, fournissant des fonctionnalités telles que la création de projets, le développement en mode développement, la compilation et la construction en mode production.

Création d'un nouveau projet Vue avec Vue CLI, Ce qui génére une structure de répertoire de base pour mon projet Vue.js, avec des fichiers et des dossiers préconfigurés. Ce qui fut une gageure pour moi car en suppriment les modules près installer « cassais » mon application ».

Configuration l’IDE VS Code : J'ai utilisé mon éditeur de texte préféré, et installé des extensions recommandées pour faciliter le développement Vue.js. Ces extensions fournissent des fonctionnalités telles que la coloration syntaxique, l'autocomplétions, le formatage de code et le débogage.

Une fois ces étapes d'installation terminées, j'ai pu commencer à développer mon site web en utilisant Vue.js et à mettre en œuvre les fonctionnalités requises pour mon projet.

## Développement

### Le Header :

1. **Création du composant Header.vue :**
   * Créez un fichier Header.vue dans votre répertoire de composants.
   * Définissez le contenu du header, y compris le logo cliquable et le menu avec des liens d'ancrage vers différentes sections.
2. **Gestion de la navigation :**
   * Utilisez Vue Router pour gérer la navigation entre les différentes pages de votre application.
   * Appliquez des classes CSS pour mettre en évidence l'élément de menu actif.

### Le Footer :

1. **Création du composant Footer.vue :**
   * Créez un fichier Footer.vue dans votre répertoire de composants.
   * Ajoutez les icônes menant vers vos réseaux sociaux avec des liens qui s'ouvrent dans un nouvel onglet.
   * Affichez la date de la dernière mise à jour en utilisant JavaScript pour récupérer la date actuelle et la formatter.

### Page d'Accueil :

1. **Création du composant Home.vue :**
   * Créez un fichier Home.vue dans votre répertoire de composants.
   * Incluez vos nom, prénom et une brève présentation dans ce composant.
   * Affichez vos créations sous forme de cartes cliquables qui ouvrent une modal détaillée.
2. **Gestion du formulaire de contact :**
   * Créez un formulaire de contact dans le composant Home.vue.
   * Utilisez les variables d'environnement pour définir l'adresse e-mail de réception des messages.
   * Utilisez les directives de Vue.js pour lier les données du formulaire au composant.

### Modal Dynamique :

1. **Création du composant Modal.vue :**
   * Créez un fichier Modal.vue dans votre répertoire de composants.
   * Définissez la structure de la modal avec les informations détaillées de chaque création.
   * Ajoutez la fonctionnalité pour fermer la modal en cliquant sur un bouton ou en dehors de celle-ci.

### Page 404 :

1. **Création du composant NotFound.vue :**
   * Créez un fichier NotFound.vue dans votre répertoire de composants.
   * Affichez l'image et le texte "404" pour signaler une erreur de navigation.
2. **Gestion de l'affichage automatique :**
   * Utilisez Vue Router pour rediriger automatiquement les utilisateurs vers la page 404 lorsqu'une ressource n'est pas trouvée.

## Déploiement

Pour utiliser l’application en dehors de l’environnement de développement j’ai du la « construire ». la construction fournis les fichiers .html . css et javascript optimisé et mignifié .

Merge vers la branch main.

# Livrables

## Le fichier README.md

### I Portfolio - Développeur Full Stack

Ce projet est un portfolio développé avec Vue.js, conçu pour présenter les réalisations d'un développeur full stack. Il offre une vitrine professionnelle pour mettre en avant ses compétences, ses projets et ses expériences.

### II Prérequis

Avant de commencer, assurez-vous d'avoir les éléments suivants installés :

- Node.js et npm

III Installation

1. Clonez ce dépôt sur votre machine :

git clone < https://github.com/giblimdev/portfolio>

2. Accédez au répertoire du projet :

cd portfolio

3. Installez les dépendances nécessaires :

npm install

### Configuration

Avant de lancer l'application, vous devez configurer certaines variables d'environnement :

Créez un fichier .env à la racine du projet.

Définissez la variables d'environnement nécessaires, l'adresse e-mail de réception pour le formulaire de contact.

### IV Lancement

Une fois les dépendances installées et la configuration terminée, vous pouvez lancer l'application localement :

npm run serve

L’application sera accessible à l’adresse indiqué dans votre terminal. Exemple : http://localhost:5173/

### Contribution

Les contributions sont les bienvenues ! Si vous souhaitez contribuer à ce projet, veuillez suivre ces étapes :

Fork ce dépôt

Créez une branche pour votre fonctionnalité (git checkout -b feature/NouvelleFonctionnalité)

Committez vos modifications (git commit -am 'Ajout d'une nouvelle fonctionnalité')

Pushez votre branche (git push origin feature/NouvelleFonctionnalité)

Soumettez une demande d'extraction

### Auteurs

Projet initié et développer par Giblimdev

### Licence

Etat néant

## Capture d’écrans validateur W3C

Grâce au constructeur de vite nous n’avons aucune erreur ou warning. Le fichier index.html et le dossier SRC sont prêt pour être mis en ligne.

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

# Conclusion

Grace ce projet, j’ai plongé dans l’univers des frameworks et découvert l’utilisation de Vue.js 3. En tant que débutant, les défis ont été nombreux. L’installation de services, la découverte de NPM, l’utilisation de Vite : autant de nouvelles notions qui m’ont confronté à des problèmes inattendus.

L’un des choix cruciaux a été de me concentrer sur la partie fonctionnelle. Plutôt que de créer un portfolio attractif et complet pour les recruteurs car à ce stade de formation cela serai utopique, j’ai préféré me focaliser sur l’implémentation des fonctionnalités essentielles : le routeur, l’affichage dynamique des articles, etc. Cette approche pragmatique m’a permis de mieux comprendre le fonctionnement de Vue.js et d’appréhender les subtilités de la composition API. De nombreuses zones d’ombres persistent.

Les difficultés rencontrées ont été nombreuses. Les erreurs de configuration, les bugs d’affichage : autant d’obstacles qui ont testé ma patience et ma persévérance. Cependant, chaque problème résolu était une victoire, et j’ai appris à apprécier ces moments d’apprentissage intensif.

Malgré ces défis, les points positifs sont nombreux. Vue.js 3 m’a séduit par sa simplicité et sa réactivité. La composition API offre une approche plus intuitive pour structurer le code, et le système de composants facilite la réutilisation.

En conclusion, ce projet m’a ouvert les portes d’un nouvel univers, celui des frameworks JavaScript. Ce qui me permet de mieux organiser mon travail grâce aux composant.