|  |  |  |
| --- | --- | --- |
|  | **BTS SIO**  **Services Informatiques aux Organisations** | |
| **Option** | SISR |
| **Session** | **2021/22** |

|  |  |  |
| --- | --- | --- |
| Nom et prénom : | **Activité professionnelle N°** | 05 |

|  |  |
| --- | --- |
| **NATURE DE L'ACTIVITE** | Mise en place d'un site web portofolio avec GitHub et Jekyll |
| **Contexte** | Préparation de l’épreuve E4 du BTS SIO |
| **Objectifs** | Réaliser un site web portofolio permettant de résumer mes travaux en BTS |
| **Lieu de réalisation** | H3 Campus/Domicile |

|  |
| --- |
| **SOLUTIONS ENVISAGEABLES** |
| * Utiliser un outil de création de site web type Winx/Google Site * Utiliser un CMS type Wordpress/Joomla et trouver un hébergeur * Utiliser la fonction pages de GitHub afin d’héberger gratuitement un site web |

|  |  |
| --- | --- |
| **DESCRIPTION DE LA SOLUTION RETENUE** | |
| **Conditions initiales** | Compte GitHub créé avec un repo « fabienchevalier.github.io » |
| **Conditions finales** | Site web accessible à l’adresse https://fchevalier.net/bts |
| **Outils utilisés** | Documentation de Jekyll, GitHub et Markdown. HTML/Markdown. Console de gestion de domaine OVH |

|  |  |
| --- | --- |
| **CONDITIONS DE REALISATION** | |
| **Matériels** | Macbook Pro 13’ |
| **Logiciels** | Visual Studio Code, rubyEnv, Homebrew, Jekyll, GitHub Page, OVH Cloud |
| **Durée** | 1 mois |
| **Contraintes** | / |

|  |  |
| --- | --- |
| **COMPETENCES MISES EN OEUVRE POUR CETTE ACTIVITE PROFESSIONNELLE** | |
| Code | Intitulé |
| Organiser son développement professionnel | ▸Mettre en place son environnement d’apprentissage personnel  ▸Gérer son identité professionnelle  ▸Développer son projet professionnel |
| Mettre à disposition des utilisateurs un service informatique | ▸Réaliser les tests d’intégration et d’acceptation d’un service |
| **DEROULEMENT DE L'ACTIVITE** | |
| **I/ Mise en place de l’environnement de développement**  GitHub Pages utilise le CMS Jekyll pour effectuer le rendu des pages web. Afin de pouvoir tester mon site web en local et de voir les modifications apparaître au fur et à mesure, j’ai décidé d’installer Jekyll sur mon Mac. Le logiciel permet en effet à travers un petit serveur web de voir ses modifications apparaître en temps réel. Aussi, afin de pouvoir publier et apporter des changements à mon site web, j’ai configuré Visual Studio pour qu’il puisse interagir avec mon repo GitHub à travers la mise en place d’un token. Pour réaliser cette activité, j’ai suivi les documentations officielles des outils proposés, ainsi qu’un article de blog pour la partie DNS.  *Sources de documentation :*  <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/markdown/>  <https://jekyllrb.com/docs/>  <https://da-sha1.me/configuration/2019/03/03/redirect-custom-domain-to-github-pages.html>  <https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site>  <https://mmistakes.github.io/minimal-mistakes/docs/quick-start-guide/>  **A) Paramétrage de Visual Code pour se connecter à mon repo GitHub**  Dans mes documents, j’ai créé un dossier fabienchevalier.github.io :    Ce dossier contient l’ensemble de mon site web. Visual Studio me permet de naviguer à ma guise dans le contenu du dossier :    Dans mon compte GitHub, j’ai créé un token de connexion :    Puis simplement configuré GIT pour se connecter à mon repo avec les identifiants suivants via cette commande :    Une fois cela fait, mes changements sont visibles dans cet onglet :    Il me suffit de les appliquer pour mettre à jour mon site.  **B) Installation de Jekyll pour permettre de voir mon site en temps réel en local.**  L’installation de Jekyll est simple. Jekyll nécessite une version récente de Ruby pour fonctionner. La version par défaut installée avec MacOS ne convenant pas pour le faire fonctionner, je suis passé par rbenv pour installer une version plus récente à côté de celle installée par défaut. J’utilise homebrew comme gestionnaire de paquet sous MacOS :  brew install rbenv  Puis :  rbenv install 3.0.0  rbenv global 3.0.0  Une fois la bonne version de ruby installée, j’ai pu installer Jekyll :  gem install --user-install bundler jekyll  Ensuite, il me suffit de naviguer dans mon dossier GitHub, et de lancer la commande jekyll serve pour que mon site soit accessible depuis l’adresse localhost :4000 :      **II/ Réalisation du site web**  Je me suis basé sur un template pré-existant appelé Minimal Mistake que j’ai modifié pour l’adapter à mon portofolio. J’ai dû apprendre le language « Markdown » afin de mettre en forme mon site web, ainsi que le mode de fonctionnement. Concrètement, mes pages sont stockées dans le dossier « \_pages » :    Voici par un exemple ma page « ppe » vu via le navigateur :    Sous Visual Studio, cela ressemble à ceci :    J’ai ensuite rédigé chaque élément en suivant les modifications sur mon second écran grâce à Jekyll avant de valider les changements.  **III/ Rendre mon site accessible via mon nom de domaine (en https)**  Github permet de configurer un nom de domaine personnalisé à son site web. Je possède le nom de domaine « fchevalier.net ». La configuration est simple et bien documentée. Voici comme j’ai procédé :  Sur les paramètres de mon repo, dans la section « page », j’ai tapé l’adresse souhaitée :    Cela a pour effet de créer un fichier « CNAME » à la base de mon repo, contenant l’adresse tapée. Il m’a suffi ensuite de rajouter 4 champs A et AAAA dans les paramètres de mon domaine pointant vers les IP des serveurs de GITHUB pour que le lien se fasse :    Il a fallu ensuite attendre que la propagation se fasse, et que le certificat soit généré pour la connexion HTTPS. Pour suivre l’avancement de la propagation, des outils en ligne existent tel que dnschecker.org. Pour vérifier que l’opération ai réussi, on peut aussi taper la commande suivante dans un terminal sous Unix :    Comme indiqué sur la capture, mon nom de domaine pointe bien vers les serveurs de GitHub. Mon site est désormais accessible via l’adresse <https://fchevalier.net/bts>. | |

|  |
| --- |
| **CONCLUSION** |
| La création de site web via GitHub paraît compliquée au premier abord, mais une fois une période d’adaptation passée, elle se révèle extrêmement simple et rapide. La connaissance des langages web tell que CSS/HTML et JavaScript reste cependant nécessaire pour bien comprendre la structure et permettre une personnalisation plus poussée. |