

Modèle de copie : Évaluation en cours de formation



Développeur Web et Web Mobile

Ceci est un modèle de copie. N'oubliez pas de renseigner vos prénom/nom, ainsi que le nom et le lien vers le projet.

Vous pouvez bien sûr agrandir les cadres pour répondre aux questions sur la description du projet si nécessaire.

Prénom : Pierre

Nom : CHAMINADE

ATTENTION ! PENSEZ À RENSEIGNER VOS NOM ET PRÉNOM DANS LE TITRE DE VOS FICHIERS / PROJETS !

Nom du projet : ECOIT

Lien Github du projet : <https://github.com/Pierrechami/Evaluation-ECF>

Lien Drive du projet (si nécessaire) :

Attention ! Merci de bien classer vos documents dans votre Github ou votre drive.

URL du site (si vous avez mis votre projet en ligne) : <https://ecoit.online/>

TRELLO: <https://trello.com/b/AzvcTytz/ecf>

WIREFRAMES : <https://www.figma.com/file/yNnEiAPajFN94ICjYyfa4b/Untitled?node-id=0%3A1>

Description du projet

1. Liste des compétences du référentiel qui sont couvertes par le projet

Activité – Type 1 : Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

1. Maquetter une application
2. Réaliser une interface utilisateur web statique et adaptable
3. Développer une interface utilisateur web dynamique
4. Réaliser une interface utilisateur avec une solution de gestion de contenu ou ecommerce.

Activité – Type 2 : Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

1. Créer une base de données
2. Développer les composants d'accès aux données
3. Développer la partie back-end d'une application web ou web mobile
4. Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce

2. Résumé du projet en français d'une longueur d'environ 20 lignes soit 200 à 250 mots, ou environ 1200 caractères espaces non compris

Dans le cadre de ma formation STUDI, j'ai comme projet le développement d'un site web comportant le front-office et le back-office, pour un organisme de formation nommé « EcoIT ». C'est un fait, en matière d'émissions de CO2, une page web pollue 115 fois plus qu'il y a 20 ans. Au vu de la crise écologique, l'entreprise EcoIT désire devenir la référence française pour les développeurs soucieux de leur impact digital. Le site web EcoIT est une plateforme d'éducation permettant à tout instructeur validé par l'équipe d'EcoIT de présenter des modules de cours. 2 Types d'utilisateurs peuvent se connecter au site, les instructeurs et les apprenants. Les instructeurs doivent être experts en accessibilité et en éco-conception web, ils ont pour objectif de créer des formations pour les apprenants, en respectant la stratégie écologique que veut mettre en œuvre l'organisme de formation. Les apprenants sont des internautes disposant d'un compte « Apprenant », souhaitant développer leurs compétences dans le développement web orienté éco responsable. Les apprenants, peuvent suivre l'intégralité des cours proposés sur la plateforme. Afin d'organiser au mieux leur apprentissage, ils disposent d'une barre de progression pour chaque formation, ainsi qu'un quiz pour chaque section et peuvent aussi poser des commentaires sur chaque cours, afin d'être aidé par la communauté ou par les instructeurs.

3. Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet

US1. Gérer les instructeurs « Utilisateurs concernés : administrateur »

L'administrateur est un employé de l'organisme de formation, il aura la possibilité de créer les comptes instructeurs. Une page du site sera dédiée aux internautes souhaitant postuler en tant qu'instructeur, pour cela, ils devront remplir un formulaire avec leur nom, prénom, e-mail, photo de profil, petite description sur leurs spécialités ainsi qu'un mot de passe sécurisé. Après leur inscription, les postulants auront une réunion avec l'administrateur, si l'échange est positif alors, l'administrateur validera leur compte.

US2. Créer un compte d'apprenant « Utilisateurs concernés : apprenants »

Un internaute peut se créer un compte « apprenant » sur une page dédiée, il suffira de remplir le formulaire d'inscription avec son adresse e-mail, un pseudo et un mot de passe sécurisé. Une fois son inscription terminée, il pourra s'identifier sur le site et aura la possibilité de suivre les formations.

US3. Se connecter « Utilisateurs concernés : administrateurs, instructeurs, apprenants »

Quelque soit le type de compte, tout visiteur pourra se connecter grâce au même formulaire de connexion. Les identifiants à entrer seront l'adresse e-mail et le mot de passe

US4. Créer une formation « Utilisateurs concernés : instructeurs »

Les instructeurs avec un compte valide pourront créer leurs propres formations. Une formation possède un titre, elle est structurée grâce à des sections, et ces dernières regrouperont des leçons. Chaque leçon aura une vidéo ainsi qu'une explication textuelle. Des ressources (images, pdf, etc.) peuvent être jointes à une leçon si besoin.

US5. Suivre une formation « Utilisateurs concernés : apprenants »

Pour suivre une formation, il est obligatoire d'avoir un compte apprenants. Lorsque l'apprenant clique sur le bouton « leçon terminée », la leçon est validée et la progression de la formation augmente. Si toutes les leçons ont été suivies, alors la formation est considérée comme terminée pour l'apprenant.

US6. Découvrir le répertoire des formations « Utilisateurs concernés : apprenants, visiteurs »

Dans ce catalogue « accéder aux formations », les formations seront réparties dans une grille. Chaque formation verra affiché son titre, une image, un petit texte d'accroche et un bouton pour accéder à la formation. Deux fonctions dynamiques doivent être disponibles pour filtrer les résultats, et ce, sans recharger la page. Une barre de recherche, une option "Formations en cours" / "Formations terminées" si l'utilisateur est un apprenant seront donc présentes.

Sur la page d'accueil, seules les 3 formations sorties les plus récemment seront visibles.

US7. Lier un quiz à une section « Utilisateurs concernés : apprenants »

Afin de valider les concepts enseignés dans les leçons, un quiz sera rajouté aux sections. Sur la page d'un quiz, les questions sont affichées les unes à la suite des autres. Chaque question est un formulaire dont les réponses sont des champs de type radio. Au clic du bouton "corriger", le quiz révèle si les réponses choisies sont correctes ou incorrectes. Si la réponse sélectionnée est incorrecte, alors la bonne réponse est montrée.

4. Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile

Pour la réalisation de ce projet, j'ai fait certains choix :

Partie Front-end :

Une base HTML 5 et CSS 3, qui sera épaulée par le Framework Bootstrap. Bootstrap dispose d'un système de grille qui permet de positionner les éléments HTML sur les pages Web. Possibilité de rendre un site web responsive de manière assez simple par rapport aux méthodes CSS pures. Compatibilité entre tous les navigateurs. J'ai aussi utilisé le langage de programmation JavaScript afin d'avoir des pages web interactives.

Partie Back-end :

Le choix du framework Symfony (PHP 7.3) m'a permis de gagner en productivité sur ce projet. En effet, l'ajout de bundles sous Symfony garanti une gestion simplifiée et performante pour les fonctionnalités requises par le projet. Plus particulièrement certains bundles proposent de nombreux outils pour gérer et renforcer la sécurité d'une application web.

Partie Hébergement :

J'ai choisi Infomaniak, pour le déploiement de l'application en ligne. Infomaniak est une entreprise spécialisée dans l'hébergement de sites web, il dispose d'un protocole HTTPS qui rajoute une sécurité supplémentaire en garantissant la sécurisation des échanges.

De plus, Infomaniak est un hébergeur vert et responsable, ce qui est en parfait accord avec le projet.

5. Description de la veille, effectuée par le candidat durant le projet, sur les vulnérabilités de sécurité

Mesures de sécurité :

Le bundle « SecurityBundle » de Symfony fournit une grande panoplie d'outils pour gérer et renforcer la sécurité de l'application.

L'authentification :

- Renforcement de la politique de sécurité lors de la création de mot de passe par l'utilisateur.

- Les mots de passes sont chiffrés en Base de données

- Protection contre les attaques grâce au token CSRF

Les autorisations :

- Gestion des droits utilisateurs, et restriction des accès à certaines pages de l'application

Protection contre les injections :

- Validation obligatoire des données entrées par l'utilisateur, grâce au système de « contrainte » de Symfony

- ORM doctrine permet de gérer les interactions avec la base de données, protéger l'application des injections SQL

- Le moteur de template TWIG protège l'application contre les attaques de type « Cross Site Scripting » (XSS), grâce au système de « output escaping » qui permet l'échappement des données.

Le protocole Https :

- Le service HTTPS activé vient renforcer la sécurité des échanges, grâce à un certificat SSL, permettant l'authentification, le chiffrement et le déchiffrement des données envoyées.

6. Description d'une situation de travail ayant nécessité une recherche, effectuée par le candidat durant le projet, à partir de site anglophone

Une fois la réalisation de mon projet finalisé, je souhaitai le déployer en production grâce à l'hébergeur web Infomaniak.

7. Extrait du site anglophone, utilisé dans le cadre de la recherche décrite précédemment, accompagné de la traduction en français effectuée par le candidat sans traducteur automatique (environ 750 signes).

Anglais

A) Check Requirements

There are some technical requirements for running Symfony applications. In your development machine, the recommended way to check these requirements is to use Symfony CLI. However, in your production server you might prefer to not install the Symfony CLI tool. In those cases, install this other package in your application:

- composer require symfony/requirements-checker

B) Configure your Environment Variables

Most Symfony applications read their configuration from environment variables. While developing locally, you'll usually store these in `.env` and `.env.local` (for local overrides). On production, you have two options:

Create "real" environment variables. How you set environment variables, depends on your setup: they can be set at the command line, in your Nginx configuration, or via other methods provided by your hosting service

Or, create a `.env.local` file like your local development. There is no significant advantage to either of the two options: use whatever is most natural in your hosting environment.

You might not want your application to process the `.env.*` files on every request. You can generate an optimized `.env.local.php` which overrides all other configuration files:

- composer dump-env prod

C) Install/Update your Vendors

Your vendors can be updated before transferring your source code (i.e. update the vendor/ directory, then transfer that with your source code) or afterwards on the server. Either way, update your vendors as you normally do:

- composer install --no-dev --optimize-autoloader

D) Clear your Symfony Cache Make sure you clear and warm-up your Symfony cache:

- APP_ENV=prod APP_DEBUG=0 php bin/console cache:clear

Français

A) Vérifier les exigences

Il existe certaines exigences techniques pour exécuter les applications Symfony. Dans votre machine de développement, la méthode recommandée pour vérifier ces exigences est d'utiliser Symfony CLI. Cependant, dans votre serveur de production, vous préférerez peut-être ne pas installer l'outil Symfony CLI. Dans ces cas, installez cet autre package dans votre application :

- composer require symfony/requirements-checker

B) Configurez vos variables d'environnement

La plupart des applications Symfony lisent leur configuration à partir des variables d'environnement. Lorsque vous développez en local, vous les stockez généralement dans in.env et. env.local (pour le fonctionnement en local). En production, vous avez deux options :

Créez des variables d'environnement "réelles". La façon dont vous définissez les variables d'environnement dépend de votre configuration : elles peuvent être définies en ligne de commande, dans votre configuration Nginx ou via d'autres méthodes fournies par votre service d'hébergement
Ou, créer un fichier .env.local comme votre développement local. Il n'y a aucun avantage significatif à l'une ou l'autre des deux options : utilisez ce qui est le plus naturel dans votre environnement d'hébergement.

Vous ne souhaitez peut-être pas que votre application traite les .env.*fichiers à chaque demande. Vous pouvez générer un fichier optimisé .env.local.php qui remplace tous les autres fichiers de configuration :

- composer dump-env prod

C) Installer/Mettre à jour vos fournisseurs

Vos fournisseurs peuvent être mis à jour avant de transférer votre code source (c'est-à-dire mettre à jour le vendor/répertoire, puis transférer cela avec votre code source) ou après coup sur le serveur. Dans tous les cas, mettez à jour vos fournisseurs comme vous le faites habituellement :

- composer install --no-dev --optimize-autoloader

D) Videz votre cache Symfony

Assurez-vous de vider et de réchauffer votre cache Symfony :

- APP_ENV=prod APP_DEBUG=0 php bin/console cache:clear

8. Autres ressources

9. Informations complémentaires

Les logins de l'application :*Un administrateur*Email = chaminadepierre.24@gmail.com

Mdp = superAdmin

*Un instructeur non validé*Email = instructeurnonvalide@gmail.com

Mdp = Instructeur

*Un instructeur*Email = instructeur@gmail.com

Mdp = Instructeur

*Un apprenant*Email = apprenant@gmail.com

Mdp = Apprenant

Github :

Vous trouverez l'ensemble des livrables attendus dans mon projet Github tels qu'un fichier readme.md, un manuel d'utilisation au format PDF expliquant comment tester le fruit de mon travail et une documentation technique au format PDF avec mes réflexions initiales sur le projet (diagramme de classe UML, diagrammes de cas d'utilisation et diagramme de séquence)