Modèle de copie :
Évaluation en cours de formation

Développeur Web et Web Mobile

Prénom : Semih
Nom : BASAK

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

Nom du projet : Hypnos Groupe Hotel

Lien Github du projet : https://github.com/BASAKSemih/ecf-studi-BASAK-Semih
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) : http://hypnoshotel.basaksemih.com/
TRELLO: https://trello.com/invite/b/AtuBL2Wt/6392c31b15c14678312d015a668fb1b6/ecf-studi-basak-semih
WIREFRAMES : https://www.figma.com/file/eq01GzQjU4h4gYl3wuIxDM/ECF-Studi

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 e-commerce.

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

J'ai choisi le sujet groupe hôtelier Hypnos, un projet de développement d'une application web & mobile.
La problématique du groupe Hypnos, c'est qu'il souhaite ne pas dépendre de Booking pour la réservation des chambres ou voir les hôtels disponibles, chambres prix...
L'application se présente sous 3 formes principale, là parti administrateurs pour gérer les établissements que ce soit la création ou la mise à jour et aussi ajouter ou modifier un gérant qui possède un établissement propre-à lui seul. L'administrateur peut voir les messages que les utilisateurs ou visiteurs qui ont envoyée depuis le formulaire de contact disponible sur le site.
Le gérant doit pouvoir rajouter des chambres à son établissement dédier et pas les autres établissement. Il a le choix entre créer des suites ou les mettre à jour avec une description, prix, image, mise en avant...
Les visiteurs ou utilisateurs dois pouvoir réserver une chambre avec une date arrivée et une date de sortie et des informations complémentaires et il doit aussi pouvoir la disponibilité d'une suite. Ils peuvent aussi connecter ou non contacter le groupe hôtelier (l'administrateur et le seul qui reçoit les messages actuellement) directement depuis l'application.
Les utilisateurs, gérant ou administrateur on leur propre authentification qui respecte la sécurité à l'accès aux données par exemple un utilisateur ne pourra pas se connecter depuis l'espace gérant ou administrateur, inversement pour administrateur et le gérant.
Grâce à ça cela favorise le fonctionnement et la sécurité de l'application web.

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

US1. Gérer les établissements « Utilisateurs concernés : Administrateur »
L'administrateur est un employé du groupe Hypnos, il a la possibilité de créer des établissements avec un nom, une ville, une adresse, une description et une image principale de l'établissements, et leur attribuer un gérant avec un nom, prénom, email de connexion et un mot de passe.

US2. Gérer les suites « Utilisateurs concernés : Gérants »
Le gérant se connecte depuis l'espace de connexion gérant avec les identifiants fournis par l'administrateur. Il ne voit que son établissement attribué par l'administrateur et depuis son interface il peut créer des suites ou plusieurs avec un titre, une image mise en avant, une description, un prix et un lien de réservation vers booking.com.

US3. Découvrir le catalogue des établissements
« Utilisateurs concernés : Visiteurs»
Un visiteur du site ou un utilisateur peut voir les établissements l'adresse, la description, l'image ainsi que voir les suites et la description et l'image.

US4. Réserver une suite « Utilisateurs concernés : Visiteurs, Clients »
Les utilisateurs ou visiteurs on accès à un formulaire où ils peuvent choisir l'établissement ainsi que la suite qui appartient à l'établissement et renseigner la date d'arriver et de sortie pour connaître la disponibilité de la suite. Si le visiteurs non connecter souhaite réserver par la suite du formulaire il doit se connecter ou se créer un compte, avec un nom, prénom, une adresse email et un mot de passe de sécurité.
Tout cela en Javascript se qui fait que les réponses sont dynamiques.

US5. Voir ses réservations « Utilisateurs concernés : Clients »
Un client qui se connecte grâce à ses identifiant depuis l'interface de connexion d'utilisateur, peut à tout moment retrouver ses réservations depuis sont interface utilisateur ainsi qu'annuler la réservation si c'est fait 3 jours avant la date d'arrivé.

US6. Accélérer la réservation d'une suite« Utilisateurs concernés : Visiteurs, Clients »
Un client ou un visiteur peut à tout moment cliquer sur le bouton réservation en dessous d'une suite pour être rediriger vers un formulaire pour renseigner la date d'arriver et de sortie pour savoir s'il est disponible. Le formulaire avec les champs établissement ou suite sera automatique pré remplis pour la suite en question.

US7. Contacter le groupe hôtelier « Utilisateurs concernés : Visiteurs, Clients, Administrateurs »
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 :

Pour le front-end j'ai décidé de partie sur le twig qui est un langage de templating avec HTML et CSS accompagnée du puissant framework Bootstrap et webpack-encore qui ce charge du build Javascript (pour dynamiser les pages webs) et le CSS SCSS pour la rapidité d'affichage de page. Grâce a le regroupement de ses technologies front-end nous sommes sûres d'avoir une application rapide et fiable à l'utilisateurs.

Partie Back-end :

Pour les choix technologiques back-end de mon projet Hypnos, j'ai décidé de partir sur une structure MVC avec le framework Symfony version 5.4 qui est la LTS actuel (Long Term Support) qui assure la stabilité du framework et la sécurité. J'ai choisi php 8.0 pour mon langage de programation car, la nouvelle version de php renforce le déclaration de type dans mon projet pour assurer la correspondance du code, Symfony 5.4 et php 8.0 assemblés c'est 20% de performances améliorer. Pour ma base de données je suis partie sur l'ORM Doctrine proposée par Symfony et MySql, doctrine un puissant ORM qui assure aussi la sécurité de l'insertion en base de données.

Partie Hébergement :

L'hérbergement est assurée par o2switch un hébergeur très reconnu en France pour la qualité de ses serveurs mis a dispositions et les options disponibles. Apache et le moteur PHP LiteSpeed assure des performances incroyables aux applications php mais aussi de nombreuse fonctionnalités qui peuvent être paramétrés pour avoir des retours utilisateurs très positif. J'utilise MariaDB pour la base de données.

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

Mesures de sécurité :

Grâce au composant Security de Symfony, mon site web dispose d'un ou plusieurs pare-feu disposant d'une panoplie de configuration pour chaque pare-feu.

Chaque Entity Manager (Gérant), Admin (Admin), Utilisateur (User) à son propre pare-feu restreint et implémente UserInterface, PasswordAuthenticatedUserInterface. Car j'ai décidé de créer 3 Authenticator propre aux Users (Gérant, Admin, User) chaque Users à son propre provider. Mais le vrai atout de faire cela c'est avoir moins de bug au long terme, j'aurai pu en effet créer une seule Entity User et gérer toutes les parties à partir de ROLE qui est puissant dans symfony, mais le risque de sécurité est présente en fessant cela, mais avec 3 Users différent permet aussi d'augmenter la couche de sécurité. Un utilisateur qui essaye de se connecter à l'espace administrateur n'y arriverai jamais, vice versa aussi.

Injection SQL des formulaires : Injection SQL n'est qu'un détail quand on utilise le composant Form et l'ORM Doctrine car le composant Form ajoute une couche de sécurité concernant les formulaires et couplées à Doctrine et au Assert pour contrôler si on reçoit bien une string par exemple nous sommes garanties que notre base de données est en sécurité.

Il y a aussi Twig qui protège mon front-end des failles de sécurité pour attaquer mon serveur web, les attaques XSS, mais pour être sûre de la sécurité XSS il faut aussi choisir le bon hébergeur web.

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

Les recherches que j'ai pu effectuer sont sur les tests fonctionnelles avec Symfony, car mon but dans l'ECF c'était aussi d'avoir une couverture de code à 100% j'ai atteint ce but et je vais vous expliquer ma recherche de documentation. Avant de commencer chaque type de test à sa restriction du code, d'atteinte, en utilisant des tests unitaires nous testons un objet, un service voir comment il réagit. Un test intégration de mon avis, mes recherches et un test presque unitaire sauf avec une utilisation de base de données, en autre on peut tester nos requêtes dans nos Repository dans les tests intégrations. Le meilleur pour la fin, les tests fonctionnelles ici nous testons toute une fonctionnalité entière par exemple une inscription utilisateur, nous testons l'insertion de données dans le formulaire, s'il la réponse du serveur est 200, nous pouvons aussi tester la sécurité avec les tests fonctionnelles par exemple un utilisateur qui essaye de se connecter à l'espace administrateur avec son compte va-t-il être renvoyé vers la page de connexion utilisateur ect..

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

Functional Test Setup :
https://symfonycasts.com/screencast/phpunit/functional-tests

Functional tests look like unit tests at first: they use PHPUnit in the exact way we've been seeing. But instead of writing one test class per PHP class, you'll usually create one test class per controller class.

It doesn't have much yet, but we're going to functionally test our homepage. Since the code behind this lives in DefaultController, let's create a Controller directory in tests and add a new DefaultControllerTest class.

But now, instead of extending TestCase or KernelTestCase, extend WebTestCase. But wait! There are two! The normal base class is the one from FrameworkBundle. It actually extends KernelTestCase, which means we have all the same tools as integration tests. But, it adds a few methods to help create a client object: a special object we'll use to make requests into our app.

Today we'll choose WebTestCase from LiipFunctionalTestBundle. No surprise, this class itself extends the normal WebTestCase. Then, it adds a bunch of optional magic.

Configuration d’un test fonctionnelle :

Les tests fonctionnels ressemblent beaucoup à des tests unitaires, ils utilisent PHPUnit de la même manière, mais au lieu d'écrire une class de test par class php, on crée généralement une class de test par controlleur.

Il n'y a pas grand-chose, mais nous allons tester fonctionnellement notre page d'accueil. Puisque le code est dans le DefaultController, c'est parti pour créer un Controller dans le répertoire test et appelons le DefaultControllerTest.

Mais maintenant au lieu d'étendre TestCase ou KernelTestCase, étendez WebTestCase la class normal est l'une des class de FrameworkBundle, actuellement il étend KernelTestCast, ce qui veut dire que nous avons les mêmes outils que les tests intégrations. Mais cela rajoute peu de méthode supplémentaire pour créer un client objet : spécial car on va s'en servir pour faire des requêtes dans notre application.

Aujourd'hui on choisis WebTestCase depuis LiipFunctionalTestBundle. Pas de surprise, cette class étend elle même WebTestCase. Mais elle ajoute des fonctionnalités magiques.

8. Autres ressources

Portfolio : basaksemih.com

9. Informations complémentaires

Les différents espaces de connexions :

http://hypnoshotel.basaksemih.com/

https://github.com/BASAKSemih/ecf-studi-BASAK-Semih/tree/develop/doc
(Le mode emploie sera dans le dossier doc)

Partie Administrateur :
http://hypnoshotel.basaksemih.com/espace-administrateur/connexion
Email : john@doe.com
Mot de passe : password

Partie Manageur
http://hypnoshotel.basaksemih.com/espace-manager/connexion
Email : mathilde.marois@email.com
Mot de passe : password

Partie Utilisateur
http://hypnoshotel.basaksemih.com/espace-utilisateur/inscription

Github :

Intégration continu : https://github.com/BASAKSemih/ecf-studi-BASAK-Semih/actions
Doc supplémentaire : https://github.com/BASAKSemih/ecf-studi-BASAK-Semih/tree/develop/doc