**Développeur Web et Web Mobile**

**Prénom : Hugo**

**Nom : DESSAUW**

Nom du projet : **Dantabase**, ECF Hugo Dessauw

Lien Github du projet :

<https://github.com/HugoD66/SDS-App>

URL du site (si vous avez mis votre projet en ligne) : ……….A METTRE

Planning du projet :



* Liste des compétences du référentiel qui sont couvertes par le projet

Implantation et decryptage du projet :

* Analyse du projet
* Analyse des besoins et maquettage
* Mise en place d'un planning suivant les différents besoin de l'application
* Savoir rédiger en Francais
* Creation/Compléter les différents diagrammes UML
* Savoir créer un environnement de développement stable
* Connaitre les différents paradigmes de programmation afin de choisir celui qui sera optimum pour la création de mon application
* Choisir les différentes aides qui nous seront utiles par la suite (sites sur lesquels s'appuyer, methodologie à adopter, Frameworks, Editeur de code .... )

Développer la partie front-end d’une application web ou web

mobile:

* Intégration une interface utilisateur web statique et adaptable
* Développer une interface utilisateur web **dynamique**
* Réaliser une interface utilisateur

Développer la partie back-end d’une application web ou web

mobile:

* Créer une base de données
* Développer les composants d’accès aux données
* Développer la partie back-end d’une application Web ou Web
* Mobile
* Élaborer et mettre en œuvre des composants dans une
* application de gestion d'entreprise.

Compétences complémentaires:

* Savoir mettre en ligne un projet web, ainsi que sa base de donnée (via Heroku)
* Connaitre les menaces les plus répandues sur la sécurité des application web afin les contrer
* Creation d'un document spécifique sur les mesures de sécurité et les bonnes pratiques
* Creation d'un readme indiquant comment installer localement un projet, et le tester.
* Résumé du projet en français d’une longueur d’environ 20 lignes soit 200 à 250 mots, ou environ 1200 caractères espaces non compris

**Dantabase** est un programme de gestion en ligne d'entreprise simple d'utilisation. C'est plus précisément une application qui régle les droits d'une franchise, possédant plusieurs structures pour chaque ville où elle est implantée, et où il est possible d'y trouver plusieurs bâtiments possédés par la franchise.

Cette application est utilisée par l'équipe de développement de la marque pour y créer et actualiser les droits d'accés de chaque bâtiments, puis est consultée par leur différents collabotateurs.

On y retrouve toutes les structures en activité, ainsi que celles qui sont indisponibles pour raison diverse. Dans le dernier cas, le propriaitaire de la marque ne pourrat pas accéder à l'application.

Afin de garder une transparance totale et un suivi des différentes oppérations, un système de notification par mail est utilisé pour tenir informé les collaborateurs concernés toute modification de leurs structures.

Dans le cas présent, Dantabase a été mis a disposition pour une franchise de sport, du nom de Garde la patate. Implantée dans une grande partie des villes, son équipe technique se sert de cette application pour optimiser leur gestion au niveau nationnal.

En plus d'un gain de temps, les sites personnels des différentes structures pourront tirer leurs informations de la base de donnée de Dantabase, ce qui assurera une unification des informations et des actualisations.

* Cahier des charges, expression des besoins, ou spécifications fonctionnelles du projet

-Contexte du projet :

Plateforme de gestion en ligne d'entreprise, 3 Rôles différents ( seulement page

d'acceuil libre d'accés) ;

-Délais:

Du 3 Aout au -------------------;

-Définition de l'architecture:

Partie libre d'accés : Tous utilisateurs même non inscrit, pourront aller sur la page d'acceuil, verront seulement en lecture seule la franchise.

Partie réservée collaborateur : Quand connecté , possibilité pour tous utilisateurs de parcourir en **LECTURE SEULE** les différentes pages / listes de structure.

Partie Administrateur : Accés à la création/ modification/ suppression des structures/ bâtiments/ franchises.

* Opérations possibles selon rôle, moteur de template **TWIG** privilégié ({% if app.user.roles == ['ROLE\_ADMIN'] %}).
* Telechargement,traitement et stockage des fichiers (**Image**) pour enregistrement des franchises et des structures.
* Possibilité d'enregistrements seulement faite par l'administrateur.
* Système de mailing à mettre en place pour les collaborateurs.

-Logiciel de maquettage utilisé :

* Adobe XD;
* Spécifications techniques du projet, élaborées par le candidat, y compris pour la sécurité et le web mobile

**Spécifications techniques :** ( A retrouver également dans la Doc technique).

Compilation:

* NPM

Frontend:

* HTML5
* CSS 3 / SASS 1.54
* Boostrap 5.2
* Webpack ENCORE 1.15

Backend:

* PHP 8.1
* Symfony 6.1
* Symfony Stimulus 3.2
* Composer 2.3.5
* Xampp ( dev )
* Maria-DB 10.4.11
* Heroku ( prod )

Bundle utilisés:

* Annotation
* Twig
* Form
* Doctrine
* Validator
* Maker
* Mailer
* Google-mailer
* Requirements-checker
* Verify-Email
* String
* String-extra
* Axios
* Profiler
* Security
* Security-csrf
* Rate-limiter

L'application est développée sur un ordinateur sous windows,

le logiciel de utilisé au developpement de l'application est **PHPStorm**.

L'application possédera une vue destinée à la navigations des utilisateurs sur l'application comprenant l'interface graphique composées de pages HTML ( avec bundle **TWIG**) .

Possédera une série de controlleurs répondant aux différentes requêtes des visiteurs.

Le développement de l'application ( majoritairement en **PHP** ) devra respecteur le modéle de conception **MVC** (Modéle Vue Controleur) , mettant l'accent sur la séparation entre la logique métier, et l'affichage du logiciel.

Création de l'application en mobile first, responsive.

Ajout de mediaQuery pour adapter l'application à l'écran utilisateur.

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

- Twig: permettant de protéger l'applications de toutes failles XSS;

- Gestion des rôles et de leurs autorisationsfait dans security.yaml;

- { path: ^/permissions, roles: ROLE\_PARTENAIRE }  
- { path: ^/structure, roles: ROLE\_PARTENAIRE }

- Chiffrage du mot de passe: Hachage du mot de passe de l'utilisateur (PasswordAuthenticatedUserInterface)servant à protéger vos données en cas d'attaque;

- Csrf\_token: rendant votre connexion et vos identifiants sécurisée;

value="{{ csrf\_token('authenticate') }}"

- Des protection permettant d'éviter les attaques de force brute;

- Une série de validation lors de nos formulaire limitant les manipulations malveillantes,

login\_throttling:  
 max\_attempts: 20  
 interval: '15 minutes'

- Protection contre les injections SQL;

! Never trust User !

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

MAIIIIILINGGGGGG ?

* 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).

**Description de la recherche :**

Traitement d'un formulaire via un controlleur.

**Url du site :**

<https://symfony.com/doc/current/forms.html>

Ici ne sera traduit copié que les paragraphes d'explication.

* **ANGLAIS**

This controller follows a common pattern for handling forms and has three possible paths:

1.When initially loading the page in a browser, the form hasn't been submitted yet and **$form->isSubmitted()** returns **false**. So, the form is created and rendered;

2.When the user submits the form, **handleRequest(**) recognizes this and immediately writes the submitted data back into the **task** and **dueDate** properties of the **$task** object. Then this object is validated (validation is explained in the next section). If it is invalid, **isValid()** returns **false** and the form is rendered again, but now with validation errors;

3.When the user submits the form with valid data, the submitted data is again written into the form, but this time **isValid()** returns **true**. Now you have the opportunity to perform some actions using the **$task** object (e.g. persisting it to the database) before redirecting the user to some other page (e.g. a "thank you" or "success" page);

* **FRANCAIS**

Ce controlleur suit un pattern commun pour gérer le formulaire et a trois solutions possible:

1.Quand initialement on charge la page dans un navigateur, le formulaire n'est pas encore soumis et **$form->isSubmitted()** retourne **faux.**Donc, le fomulaire est créé et rendu;

2.Quand l'utilisateur soumet le fomulaire, **handleRequest()** le reconnait et écrit immédiatement les données dans les propriétées de **task** et **duelDate** de l'objet **$task.** Cet objet est donc validé(la validation est exipliquée dans la partie suivant). Si il est invalide, **isValid()** retournera **faux** et le formulaire est rentourné, mais maintenant avec les erreurs que contient la validation;

3.Quand l'utilisateur soumet le formulaire avec des données valides, les données soumises sont encore rééccrite dans le formulaire, mais cette fois **isValid()** retourne **vrai**. Vous avez maintenant l'opportunitée de réaliser certaines actions en utilisant l'objet **$task** (exemple le persister dans la base de donnée) avant la redirection de l'utilisateur vers une autre page (exemple une page "merci" ou "succés");

* Autres ressources
* Aide principales :

STUDI : <https://app.studi.fr/>

SYMFONY : <https://symfony.com/doc/current/index.html>

YOUTUBE : <https://www.youtube.com/c/ThomasBoileau>

MDN : <https://developer.mozilla.org/>

* Aides selection photo :

UNPLASH : <https://unsplash.com/>

* Aide frameworks Bootstrap :

BOOTSTRAP : <https://getbootstrap.com/docs/4.4/getting-started/introduction/>

* Aide organisation du planning :

TRELLO : <https://trello.com/hugodessauw/boards>

* Aide palette de couleur :

HTMLCOLORS: [https://htmlcolorcodes.com](https://htmlcolorcodes.com/)

* Aide commandes GIT :

HOSTINGER : <https://www.hostinger.fr/tutoriels/commandes-git>

* Aide icones :

ICONE8 : <https://icones8.fr/icons/set/cookie>

BOOTSTRAP : <https://icons.getbootstrap.com/>

* Aides complémentaires :

Ancien projet ECF : <https://github.com/HugoD66/Ecoit-Ecf>

Guide ECF

Lives Jean Baptiste LAVISSE

* Informations complémentaires