



MINISTÈRE CHARGÉ  
DE L'EMPLOI

# DOSSIER PROFESSIONNEL(DP)

*Nom de naissance*

► BOUVIER

*Nom d'usage*

► Entrez votre nom d'usage ici.

*Prénom*

► Vanessa

*Adresse*

► 17 rue Frédéric Chopin  
10300 Ste-Savine

## Titre professionnel visé

Développeur Web et Web Mobile

### MODALITE D'ACCES :

- Parcours de formation
- Validation des Acquis de l'Expérience (VAE)

# DOSSIER PROFESSIONNEL(DP)

## Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel. **Ce titre est délivré par le Ministère chargé de l'emploi.**

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente **obligatoirement à chaque session d'examen.**

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

### Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. Du **Dossier Professionnel** (DP) dans lequel le candidat a consigné les preuves de sa pratique professionnelle
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

*[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]*

### Ce dossier comporte :

- ▶ pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- ▶ un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- ▶ une déclaration sur l'honneur à compléter et à signer ;
- ▶ des documents illustrant la pratique professionnelle du candidat (facultatif)
- ▶ des annexes, si nécessaire.

*Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.*



<http://travail-emploi.gouv.fr/titres-professionnels>

# DOSSIER PROFESSIONNEL(DP)

## Sommaire

### Exemples de pratique professionnelle

**Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité** p. 5

- ▶ Périodes-stage.fr ..... p. 5
- ▶ BKP ..... p. 7

**Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité** p.

- ▶ Périodes-stage.fr ..... p. 9
- ▶ BKP ..... p. 12

**Titres, diplômes, CQP, attestations de formation (*facultatif*)** p. 15

**Déclaration sur l'honneur** p. 16

**Documents illustrant la pratique professionnelle (*facultatif*)** p. 17

**Annexes (Si le RC le prévoit)** p. 18

# **EXEMPLES DE PRATIQUE PROFESSIONNELLE**

# DOSSIER PROFESSIONNEL(DP)

## Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°1 ► [Periodes-stage.fr](http://Periodes-stage.fr)

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour l'obtention des Certifications de Développeur Intégrateur Web, j'ai travaillé sur un projet de groupe.

Le but du projet était de créer le site « periodes-stage.fr », sur lequel un utilisateur (Responsable pédagogique, secrétaire...) peut enregistrer une ou plusieurs écoles (Université, ESC, IUT...) et référencer les stages existants pour celles-ci. Nous devions réaliser un site web responsive qui référence les périodes stages des écoles par date, type de formation... pour permettre à de futurs employeurs et/ ou étudiants de connaître les dates de stage des écoles.

Nous avons créé le site avec le Framework MVC Symfony. J'ai eu plusieurs tâches à effectuer sur le Front.

#### PAGE ACCUEIL (ANNEXE 1)

- J'ai créé presque toute la page d'accueil sauf le footer et la barre de navigation. J'ai effectué la mise en forme, grâce à Bootstrap, ajouté les images de fond, ainsi que le logo.
- J'ai été chargée de rendre la page responsive grâce à des Media Queries et en modifiant le footer en version mobile. Pour cela j'ai remplacé les titres du footer (« A propos », « FAQs »..) par des boutons en créant des IDs permettant de faire apparaître ou de cacher des éléments en fonction de la taille de l'écran.
- J'ai également utilisé les class Bootstrap pour modifier la taille des colonnes et le padding en fonction de la taille de l'écran. Par exemple : class= "xs-col-12 sm-col-6 px-3 sm-px-0".
- J'ai aussi créé un bouton de retour vers le haut. Je l'ai positionné visuellement au niveau du footer et je l'ai ancré dans le code au-dessus de la barre de recherche, ce qui fait que quand on clique dessus, on est ramené directement en haut du site, au lieu de scroller.

#### SCRIPT JS (ANNEXE 2)

J'ai codé un script Javascript qui affiche un texte de présentation du site (slogan demandé par le client) : « Périodes-stage.fr, trouvez facilement les périodes de stage ». Ce texte s'affiche sur la page d'accueil au-dessus de la barre de recherche et apparaît au chargement de la page. J'ai mis un délai de 0,130 secondes pour l'apparition de chaque caractère. Un fois le texte entièrement affiché, il reste fixe. Seul le recharge de la page peut relancer le script.

#### PARTIE ADMINISTRATION DES ECOLES (ANNEXE 3)

J'ai réalisé la page « Espace Ecole » avec la liste des stages et dans l'espace utilisateur la partie avec la liste des écoles enregistrées. J'ai créé des tableaux html pour afficher les stages et les écoles. Ces tableaux sont responsives grâce à des Medias Queries. J'ai supprimé des colonnes des tableaux au fur et à mesure que la taille de l'écran diminue et j'ai changé la disposition des éléments sur la page : l'un à côté de l'autre puis l'un au-dessus de l'autre. J'ai aussi créé deux types de boutons : des boutons larges qui apparaissent sur les écrans larges et des petits boutons qui apparaissent uniquement sur les plus petits écrans (version mobile). J'ai programmé ces boutons en leur mettant des IDs qui sont contrôlées par les Media Queries.

# DOSSIER PROFESSIONNEL(DP)

## CARTES DE RESULTATS (ANNEXE 4)

J'ai mis en forme les cartes affichant les résultats de recherche. J'ai rendu les cartes responsives en les faisant passer de deux par ligne à une seule sur les écrans plus petits. J'ai également ajouté un mouse hover pour mettre en évidence la carte sur laquelle pointe la souris. La carte change de couleur et s'agrandit ( « scale(1.05) » ). J'ai rajouté les propriétés de transformation « webkit-transform », « oz-transform »... pour éviter les problèmes de compatibilité avec les différents navigateurs.

Le projet entier comporte 544 lignes de CSS, dont j'ai écrit environ 85 à 90%.

## 2. Précisez les moyens utilisés :

### Technologies utilisées :

Framework MVC : **SYMFONY**

Moteur de templates PHP : **TWIG**

Framework CSS : **BOOTSTRAP**

Outil de versionning : **GIT**

### Langages de programmation :

**Javascript**

**Html**

**CSS**

## 3. Avec qui avez-vous travaillé ?

### Notre équipe était composée de 4 personnes :

Justine Jourdain, Camille Gomez, Elie Francisco, Vanessa Bouvier

## 4. Contexte

Nom de l'entreprise, organisme ou association ► **Webforce3 Troyes**

Chantier, atelier, service ► **Projet de fin de formation**

Période d'exercice ► **Du 19/08/2019 au 30/08/2019**

## 5. Informations complémentaires (*facultatif*)

**ANNEXE 1, ANNEXE 2, ANNEXE 3, ANNEXE 4**

# DOSSIER PROFESSIONNEL(DP)

## Activité-type 1

Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°2 ► BKP

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'effectue un stage de 6 mois (2/9/19 au 28/2/20) en tant que Développeuse Fullstack chez Spuro, une entreprise spécialisée dans la Blockchain. Une des missions principales de mon stage concerne la signature électronique BKP : Blockchain Key Proof. Cette application existait déjà mais je devais la modifier et l'améliorer pour proposer plus de fonctionnalités aux clients.

En effet, un utilisateur peut s'enregistrer à distance sur l'application Spuro, mais je devais réaliser un formulaire qui serait utilisé par une Autorité d'Enregistrement pour permettre la vérification de visu des utilisateurs. Les informations du formulaire doivent être insérées en BDD, écrites dans le Registre de la Blockchain existante, ou bien les deux.

J'ai réalisé 3 versions différentes du projet :

- 1- Un formulaire d'enregistrement avec insertion des données de l'utilisateur en Base de Données et enregistrement de l'empreinte numérique dans la Blockchain.
- 2- Un formulaire d'enregistrement renvoyant vers une page PHP qui effectue l'enregistrement de l'empreinte numérique dans la Blockchain et affiche le résultat sur une autre page que celle du formulaire.
- 3- Un formulaire d'enregistrement avec insertion des données de l'utilisateur en Base de Données, sans rechargement de la page du formulaire.

J'ai créé les 3 formulaires avec Boostrap. J'ai utilisé la class « container » de Boostrap pour centrer et rendre responsive les formulaires.

Pour la mise en forme de ceux-ci, j'ai utilisé la class «form-row» de Boostrap pour aligner des inputs sur la même ligne. Puis j'ai défini la largeur des colonnes : « form-group col-md-8 », « form-group col-md-6 » ou «form-group col-md-4 ». « md » correspondant au breakpoint MEDIUM de Boostrap c'est-à-dire une taille d'écran > 768px (taille tablette). La page du formulaire a nécessité peu de css. Les couleurs utilisées sont celles du logo de l'entreprise Spuro.

J'ai aussi créé une page affichant le résultat de la soumission du formulaire pour la 2ème version du projet : un message affiche la réussite de l'enregistrement dans la Blockchain, ainsi que le hash de l'utilisateur dans une carte créée grâce à la class="container" de Boostrap. Sur celle-ci j'ai effectué un mouse hover pour mettre en évidence le message lorsque la souris est au-dessus. La carte change de couleur et s'agrandit ('scale(1.05)') et le hash est entouré d'une bordure et surligné en blanc (sur fond bleu). Cette page est responsive, j'ai pour cela ajouté quelques Media Queries.

J'ai respecté le même code couleur sur tous les formulaires en m'inspirant des couleurs du logo de l'entreprise Spuro. (ANNEXE 5)

# DOSSIER PROFESSIONNEL(DP)

## 2. Précisez les moyens utilisés :

### Technologies utilisées :

Framework CSS : **BOOTSTRAP**

Serveur : **Laragon (HTTP Apache)**

### Langages de programmation :

**Html**

**CSS**

## 3. Avec qui avez-vous travaillé ?

### Nous étions deux dévelopeuses stagiaires :

Justine Jourdain, Vanessa Bouvier

## 4. Contexte

Nom de l'entreprise, organisme ou association▶ **Spuro**

Chantier, atelier, service ▶ **Développement**

Période d'exercice ▶ **Du 02/09/2019 au 29/02/2020**

## 5. Informations complémentaires (*facultatif*)

**ANNEXE 5**

# DOSSIER PROFESSIONNEL(DP)

## Activité-type 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°1 ► [Periodes-stage.fr](http://Periodes-stage.fr)

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Pour l'obtention des Certifications de Développeur Intégrateur Web, j'ai travaillé sur un projet de groupe.

Nous devions réaliser un site web responsive qui référence les périodes stages des écoles par date, type de formation... pour permettre à de futurs étudiants et/ ou employeurs de connaître les dates de stage.

Le but du projet était donc de créer le site [periodes-stage.fr](http://periodes-stage.fr), sur lequel un utilisateur (Responsable pédagogique, secrétaire...) peut enregistrer une ou plusieurs écoles (Universités, ESC...) et référencer les stages existants pour celles-ci.

Nous avons créé le site avec le Framework MVC Symfony. J'ai eu plusieurs tâches à effectuer sur le Back.

#### BUNDLE LIIP IMAGINE (ANNEXE 6)

J'ai installé le Bundle LIIP IMAGINE pour pouvoir afficher les miniatures des logos des écoles. Ce bundle permet de prédéfinir la même taille pour toutes les images.

Pour l'installer j'ai utilisé la commande suivante dans la console de commande :

```
$ composer require liip/imagine-bundle
```

J'ai donc utilisé ce bundle pour afficher les logos des écoles sur la « Liste des écoles » dans l'espace Administration des utilisateurs et sur les cartes affichant les résultats des recherches. J'ai paramétré les deux affichages de façon différente, les miniatures sur la « Liste des écoles » étant plus grandes que celles des cartes de résultats. J'ai effectué le paramétrage dans le fichier « `liip_imagine.yalm` » .

Pour appliquer ce bundle dans Symfony, il suffit d'écrire la balise `<img>` comme ceci :

```
<img src=“{{ asset(‘path/image.jpg’) | imagine_filter (‘my_thumb’) }}” />
```

'my\_thumb' étant l'ID du filtre paramétré.

#### PAGE DETAIL ECOLE, FORMULAIRES ET AFFICHAGE DES STAGES – PARTIE ADMIN DE L’UTILISATEUR (ANNEXE 7)

- Les utilisateurs qui s'inscrivent, enregistrent ensuite une ou plusieurs écoles puis créent des stages en lien avec une école. Plusieurs informations sont demandées : type de formation, dates du stage, type de contrat, diplôme visé... J'ai créé les formulaires concernant les stages : j'ai donc tapé `php bin/make :form` dans la console, ce qui a généré un fichier « `StageType.php` ». Dans ce fichier j'ai paramétré les champs. Puis dans « `UserControlleur.php` » j'ai créé les fonctions « `editStage()` », « `ajoutStage()` » et « `supprimerStage()` ». Symfony sécurise les formulaires grâce aux contrôleurs qui gèrent les données envoyées en BDD.
- J'ai créé la page `detailEcole.html.twig`. Cette page affiche tous les stages d'une école. Pour cela j'ai créé une boucle et affiché le résultat dans un tableau html.
- J'ai amélioré les fonctions « `ajoutEcole()` » et « `editEcole()` » en ajoutant l'upload et la modification du logo de l'école.

# DOSSIER PROFESSIONNEL(DP)

## **LIENS DE RETOUR (ANNEXE 8)**

J'ai créé les liens de retour suivants :

- Formulaires ajouter, modifier & supprimer stage vers la page de l'école (detailEcoles.html.twig)
- Formulaires ajouter, modifier & supprimer école vers la page de l'utilisateur (index.html.twig)
- Formulaire modifier les informations utilisateur vers la page de l'utilisateur (index.html.twig)

Avec Symfony pour créer un lien qui renvoie vers une autre page il faut écrire :

```
<a href="{{ path('user_index') }}"> TEXT <a/>
```

## **BASE DE DONNEES (ANNEXE 8)**

Nous avons créé la Base de Données en groupe.

Avec Symfony pour créer une Base De Données il faut taper : `php bin/console doctrine :database :create` dans la console. Ensuite nous avons créé les Entités : `php bin/console make:entity` .

Les Entités sont des fichiers php (ex : « Ecole.php ») représentant la structure d'une table SQL en BDD, composée des propriétés, getters et setters.

Pour finir nous avons créé les relations entre les tables (clés étrangères) : OneToMany, ManyToOne.

## **2. Précisez les moyens utilisés :**

### **Technologies utilisées :**

Framework MVC : **SYMFONY**

Moteur de templates PHP : **TWIG**

Outil de versionning : **GIT**

Outil de gestion de BDD : **PhpMyAdmin**

### **Langages de programmation :**

**PHP**

## **3. Avec qui avez-vous travaillé ?**

Notre équipe était composée de 4 personnes :

Justine Jourdain, Camille Gomez, Elie Francisco, Vanessa Bouvier

## **4. Contexte**

**Nom de l'entreprise, organisme ou association►    *Webforce 3 Troyes***

# DOSSIER PROFESSIONNEL(DP)

Chantier, atelier, service ► Projet de fin de formation

Période d'exercice ► Du 19/08/2019 au 30/08/2019

## 5. Informations complémentaires (*facultatif*)

ANNEXE 6, ANNEXE 7, ANNEXE 8

# DOSSIER PROFESSIONNEL(DP)

## Activité-type 2

Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Exemple n°2 ► BKP

### 1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

J'effectue un stage de 6 mois (2/9/19 au 28/2/20) en tant que Développeuse Fullstack chez Spuro, une entreprise spécialisée dans la Blockchain. Une des missions principales de mon stage concerne la signature électronique BKP : Blockchain Key Proof. Cette application existait déjà mais je devais la modifier et l'améliorer pour proposer plus de fonctionnalités aux clients.

En effet, un utilisateur peut s'enregistrer à distance sur l'application Spuro, mais je devais réaliser un formulaire qui serait utilisé par une Autorité d'Enregistrement pour permettre la vérification de visu des utilisateurs. Les informations du formulaire doivent être insérées en BDD, écrites dans le Registre de la Blockchain existante, ou bien les deux.

#### J'ai réalisé 3 versions différentes du projet :

##### **1- Un formulaire d'enregistrement avec insertion des données de l'utilisateur en Base de Données et enregistrement de l'empreinte numérique dans la Blockchain. (ANNEXE 9)**

- J'ai créé une Base de Données que j'ai appelée « bkp » et une table « utilisateurs ».

Un ID se rajoute pour chaque nouvel utilisateur, en auto-incrémentation. Concernant les champs « secu », « code\_postal » et « telephone » j'ai choisi le type VARCHAR et non INT, pour pouvoir contrôler et vérifier le nombre de caractères entrés dans les inputs et ainsi éviter les erreurs. Pour le champ « date\_de\_naissance », j'ai choisi le type DATE car l'enregistrement dans la BDD est fait au format JJ/MM/AAAA. J'ai appelé ce champ « date\_de\_naissance » et non « date » car ce mot est utilisé dans le langage SQL (type de données que j'ai d'ailleurs utilisé), ce qui peut provoquer des erreurs avec d'autres systèmes de BDD comme Oracle. Tous les autres champs sont en type VARCHAR.

- J'ai créé une page « connexion.php » avec tous les paramètres nécessaires pour effectuer la connexion à la BDD ci-dessus.

- Dans le formulaire j'ai défini la méthode POST, qui transmet les informations du formulaire de manière masquée mais non cryptée. Ensuite j'ai requis les paramètres du Framework pour pouvoir utiliser les fonctions déjà existantes, dont j'ai besoin pour l'écriture dans le Registre.

Je préfère utiliser « require\_once » plutôt que « include » ... pour 2 raisons : PHP vérifie si le fichier a déjà été inclus, et si c'est le cas, ne l'inclut pas une deuxième fois et « require\_once », à l'inverse de « include\_once », va générer une erreur et stopper l'exécution du script PHP

- J'ai fait une vérification des champs nécessaires avant de préparer la requête et de l'exécuter, pour insérer les données en BDD.

- La Blockchain Spuro ayant été créée par un Développeur extérieur, j'ai dû prendre en compte le code déjà créé pour effectuer l'écriture dans le Registre. Pour pouvoir réaliser ce projet j'ai donc dû utiliser des fonctions déjà existantes pour la création et la sauvegarde de l'empreinte, la création du Block dans le Registre et la mise à jour du Registre.

# DOSSIER PROFESSIONNEL(DP)

## 2- Un formulaire d'enregistrement renvoyant vers une page PHP qui effectue l'enregistrement de l'empreinte numérique dans la Blockchain et affiche le résultat sur une autre page que celle du formulaire.

- Cette version est basée sur la première, sauf qu'il n'y a pas d'insertion dans une BDD.
- Le formulaire d'enregistrement renvoie vers une page PHP qui effectue l'écriture de l'empreinte numérique de l'utilisateur dans la Blockchain et affiche le résultat sur une autre page que celle du formulaire. J'ai donc mis « action= "interface.php" » sur la balise <form> pour renvoyer sur la page interface.php, qui effectue la vérification des champs, le hachage des données , l'écriture dans le Registre puis affiche le message de réussite avec le hash de l'utilisateur.
- La vérification des champs est identique à la Version 1.

## 3- Un formulaire d'enregistrement avec insertion des données de l'utilisateur en Base de Données, sans rechargement de la page du formulaire. (ANNEXE 10)

- J'ai créé une Base de Données que j'ai appelée « formulaire » et une table « enregistrer ».
- Les types des champs sont identiques à la BDD de la VERSION 1.
- J'ai créé une page « connexion.php » avec tous les paramètres nécessaires pour effectuer la connexion à la BDD ci-dessus.
- L'insertion et l'affichage des messages devant se faire sans recharger la page, j'ai donc décidé d'utiliser de l'Ajax. L'Ajax (Asynchronous Javascript and XML) est une combinaison de plusieurs technologies qui permet de récupérer des données sans à avoir à recharger la page.

J'ai décidé d'utiliser la bibliothèque JS Jquery car celle-ci simplifie l'utilisation de l'ajax.

N'ayant pas appris l'AJAX durant ma formation, j'ai effectué de nombreuses recherches.

Suite à mes recherches j'ai créé une page index.php où j'ai codé le formulaire et ajouté le script ajax. J'ai ajouté l'id #form sur le formulaire. L'ajax récupère l'id et empêche la page de se recharger. Ensuite j'ai utilisé la méthode \$.post() cette requête permettant d'envoyer des données vers le serveur avec une requête HTTP POST. L'Ajax récupère la valeur des inputs et les envoie à la page insertion.php qui effectue la vérification des champs, prépare la requête et effectue l'insertion en BDD.

Une fois la méthode terminée j'ai créé une fonction qui permet d'afficher soit un message d'erreur soit un message de réussite. J'ai utilisé setTimeout() pour afficher le message d'erreur puis qu'il disparaisse au bout de 10 secondes.

Explication de l'écriture dans le Registre (ANNEXE 11)

# DOSSIER PROFESSIONNEL(DP)

## 2. Précisez les moyens utilisés :

### Technologies utilisées :

Framework CSS : **BOOTSTRAP**

Serveur : **Laragon (HTTP Apache)**

Bibliothèque JS : **JQuery**

**Spuro Blockchain Platform**

**Hash / Cryptographie**

**Ajax**

### Langages de programmation :

**PHP (POO)**

**Javascript**

## 3. Avec qui avez-vous travaillé ?

Nous étions deux développeuses stagiaires :

Justine Jourdain, Vanessa Bouvier

## 4. Contexte

**Nom de l'entreprise, organisme ou association▶ SPUR**

**Chantier, atelier, service ▶ Développement**

**Période d'exercice ▶ Du 02/09/2019 au 29/02/2020**

## 5. Informations complémentaires (*facultatif*)

**ANNEXE 9, ANNEXE 10, ANNEXE 11**

# DOSSIER PROFESSIONNEL(DP)

## Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
CNCP Techniques d'Intégration Web	WEBFORCE 3 Troyes	30/08/2019
CNCP Techniques de Développement Web	WEBFORCE 3 Troyes	30/08/2019
DAEU, option A Mention TRES BIEN	CNED & Université Paris Sud	2011
CAE Anglais niveau C1 (Cambridge English Exam)	Cambridge Assessment English (Cambridge) University	2011

# DOSSIER PROFESSIONNEL(DP)

## Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] ..**Vanessa Bouvier**.....,

déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis l'auteur(e) des réalisations jointes.

Fait à ..*Troyes*..... le ..30/10/2019.....

pour faire valoir ce que de droit.

Signature :

# DOSSIER PROFESSIONNEL(DP)

## Documents illustrant la pratique professionnelle

*(facultatif)*

### Intitulé

ANNEXE 1 : Page accueil visuel et responsive - ACTIVITE TYPE 1 Exemple 1

ANNEXE 2 : Script Javascript - ACTIVITE TYPE 1 Exemple 1

ANNEXE 3 : Pages « Détail Ecole » et « Stages » visuel et responsive - ACTIVITE TYPE 1 Exemple 1

ANNEXE 4 : Cartes des résultats de recherche - ACTIVITE TYPE 1 Exemple 1

ANNEXE 5 : Formulaire - ACTIVITE TYPE 1 Exemple 2

ANNEXE 6 : Bundle Liip Imagine - ACTIVITE TYPE 2 Exemple 1

ANNEXE 7 : Pages « Détail Ecole » et « Stages » - ACTIVITE TYPE 2 Exemple 1

ANNEXE 8 : Liens de retour + BDD - ACTIVITE TYPE 2 Exemple 1

ANNEXE 9 : BKP Version 1 - ACTIVITE TYPE 2 Exemple 2

ANNEXE 10 : BKP Version 3 avec AJAX - ACTIVITE TYPE 2 Exemple 2

ANNEXE 11 : BKP Ecriture dans le Registre - ACTIVITE TYPE 2 Exemple 2

# DOSSIER PROFESSIONNEL(DP)

## ANNEXES

*(Si le RC le prévoit)*

# ANNEXE 1

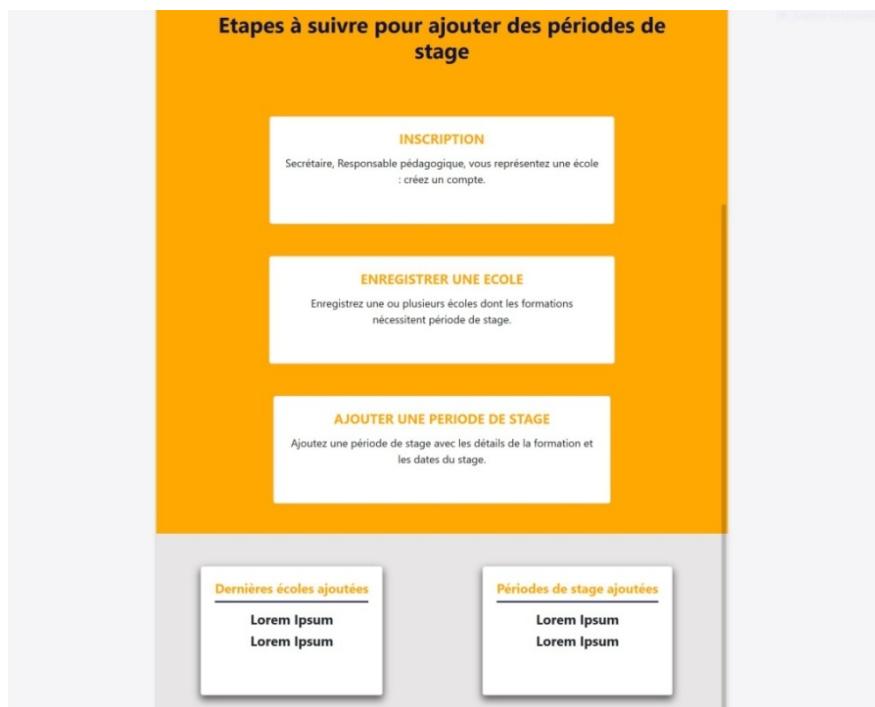
## Page accueil visuel et responsive - ACTIVITE TYPE 1 Exemple 1



## AFFICHAGE DE LA PAGE D'ACCUEIL – VERSION DESKTOP

A screenshot of the Periodes-stage.fr website homepage for desktop. The top section has a yellow background with three white boxes containing icons and text: 'INSCRIPTION' (secretary icon), 'ENREGISTRER UNE ÉCOLE' (building icon), and 'AJOUTER UNE PÉRIODE DE STAGE' (document icon). Below this is a grey section with two white boxes: 'Dernières écoles ajoutées' (with 'Lorem Ipsum' placeholder text) and 'Périodes de stage ajoutées' (with 'Lorem Ipsum' placeholder text). The footer is dark blue with white text for 'Qui sommes nous?', 'Mentions légales', 'FAQ', 'Contact', and a yellow upward arrow icon. A page number 'Page 19' is at the bottom right.

## AFFICHAGE DE LA PAGE D'ACCUEIL – VERSION TABLETTE



## AFFICHAGE DE LA PAGE D'ACCUEIL – VERSION MOBILE



## ANNEXE 2

### Script Javascript - ACTIVITE TYPE 1 Exemple 1

#### SCRIPT JS : APPARITION DU SLOGAN

```
/****** TEXTE DE PRESENTATION *****/
var content = 'Périodes-stage.fr,'+' trouvez facilement les périodes de stage';

var element = '<span>' + content.split(' ').join('</span><span>') + '</span>';

$(element).hide().appendTo('#texte-js').each(function (i) {
    $(this).delay(130 * i).css({
        display: 'inline',
        opacity: 0
    }).animate({
        opacity: 1
    }, 130);
});
```

# ANNEXE 3

Pages « Détail Ecole » et « Stages » visuel et responsive - ACTIVITE TYPE 1 Exemple 1

## AFFICHAGE DE LA LISTE DES ECOLES

Bienvenue sur votre espace personnel

Liste des écoles

Nom	Logo	Rue	Code Postal	Ville	Type d'établissement	Actions
WebForce3		17-19 rue du Colonel Driant	10000	Troyes	Autre	<a href="#">Voir plus</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
WebForce3		2 place Saint-Rémy	54300	Lunéville	Autre	<a href="#">Voir plus</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
WebForce3		Immeuble Le Mathis	67000	Strasbourg	Autre	<a href="#">Voir plus</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>
WebForce3		6 rue Gino Raymond	54590	Piennes	Autre	<a href="#">Voir plus</a> <a href="#">Modifier</a> <a href="#">Supprimer</a>

[Ajouter une école](#)

## AFFICHAGE DES STAGES, PAGE detailEcole.html.twig – VERSION DESKTOP

Espace WebForce3

INFORMATIONS DE L'ECOLE

Nom : WebForce3
Logo :
Rue : 17-19 rue du Colonel Driant
Code Postal : 10000
Ville : Troyes
Type d'établissement : Autre
<a href="#">Modifier les informations</a>

Liste des stages disponibles

Domaine d'étude	Diplôme visé	Date de début	Date de fin	Type de Contrat	Actions
Informatique, Génie logiciel	Bac +2	02.09.2019	15.10.2019	Stage conventionné	<a href="#">Modifier</a> <a href="#">Supprimer</a>
Informatique, Génie logiciel	Bac +2	15.05.2020	30.06.2020	Stage conventionné	<a href="#">Modifier</a> <a href="#">Supprimer</a>
Informatique, Génie logiciel	Bac +2	01.06.2020	01.01.2021	Stage conventionné	<a href="#">Modifier</a> <a href="#">Supprimer</a>

[Ajouter un stage](#)

## AFFICHAGE DES STAGES, PAGE detailEcole.html.twig – VERSION TABLETTE

Nom : WebForce3



Rue : 17-19 rue du Colonel Driant

Code Postal : 10000

Ville : Troyes

Type d'établissement : Autre

[Modifier les informations](#)

Domaine d'étude	Diplôme visé	Date de début	Date de fin	Type de Contrat	Actions
Informatique, Génie logiciel	Bac +2	02.09.2019	15.10.2019	Stage conventionné	<a href="#" style="color: orange;">Modifier</a> <a href="#" style="color: orange;">Supprimer</a>
Informatique, Génie logiciel	Bac +2	15.05.2020	30.06.2020	Stage conventionné	<a href="#" style="color: orange;">Modifier</a> <a href="#" style="color: orange;">Supprimer</a>
Informatique, Génie logiciel	Bac +2	01.06.2020	01.01.2021	Stage conventionné	<a href="#" style="color: orange;">Modifier</a> <a href="#" style="color: orange;">Supprimer</a>

[Ajouter un stage](#)

## AFFICHAGE DES STAGES, PAGE detailEcole.html.twig – VERSION MOBILE

Nom : WebForce3



Rue : 17-19 rue du Colonel Driant

Code Postal : 10000

Ville : Troyes

Type d'établissement : Autre

[Modifier les informations](#)

Domaine d'étude	Date de début	Actions
Informatique, Génie logiciel	02.09.2019	 
Informatique, Génie logiciel	15.05.2020	 
Informatique, Génie logiciel	01.06.2020	 

[Ajouter un stage](#)

## 2 TYPES DE BOUTONS EN FONCTIONS DES TAILLES D'ECRANS

```
{# BOUTONS SCREEN > 768px #}
<td id="cache-mobile" class="text-center border-bottom border-dark">
  <a type="submit" class="btn border-jaune d-block mt-1" href="{{ path('user_detail_ecole', { id: ecole.id }) }}>Voir plus</a>
  <a type="submit" class="btn border-jaune d-block mt-1" href="{{ path('user_edit_ecole', { id: ecole.id }) }}>Modifier</a>
  <a type="submit" class="btn btn-outline-danger d-block mt-1" href="{{ path('user_supprimer', { id: ecole.id }) }}>Supprimer</a>
</td>

{# BOUTONS SCREEN < 768px #}
<td id="show-mobile" class="border-dark" >
  {# BOUTON VOIR PLUS #}
  <a type="submit" class="d-block mt-2 btn-outline-warning" href="{{ path('user_detail_ecole', { id: ecole.id }) }}><img id="voir"></a>
  {# BOUTON MODIFIER #}
  <a type="submit" class="d-block mt-2 btn-outline-warning" href="{{ path('user_edit_ecole', { id: ecole.id }) }}><img id="modifier"></a>
  {# BOUTON SUPPRIMER #}
  <a type="submit" class="d-block mt-2 btn-outline-danger" href="{{ path('user_supprimer', { id: ecole.id }) }}><img id="supprimer"></a>
</td>
```

## EXTRAITS DES MEDIAS QUERIES

```
@media screen and (min-width: 576px) and (max-width: 767.99px){
  #inscription, #ecole, #stage h5{
    font-size: 0.7rem;
  }
  .card-etapes{
    width: 80%;
  }
}
```

```
***** MEDIA QUERIES *****
@media screen and (max-width: 575.99px){
  #texte-js{
    font-size: 1.5rem;
  }
  .accueil .container{
    width: 80vw;
  }
  .section2 h2{
    font-size: 1.5rem;
  }
  .card-etapes h5{
    font-size: 1rem;
  }
}
```

```
@media screen and (max-width: 991.99px){
  #concept{
    width: 80vw;
  }
  .home header{
    height: 100vh;
  }
}

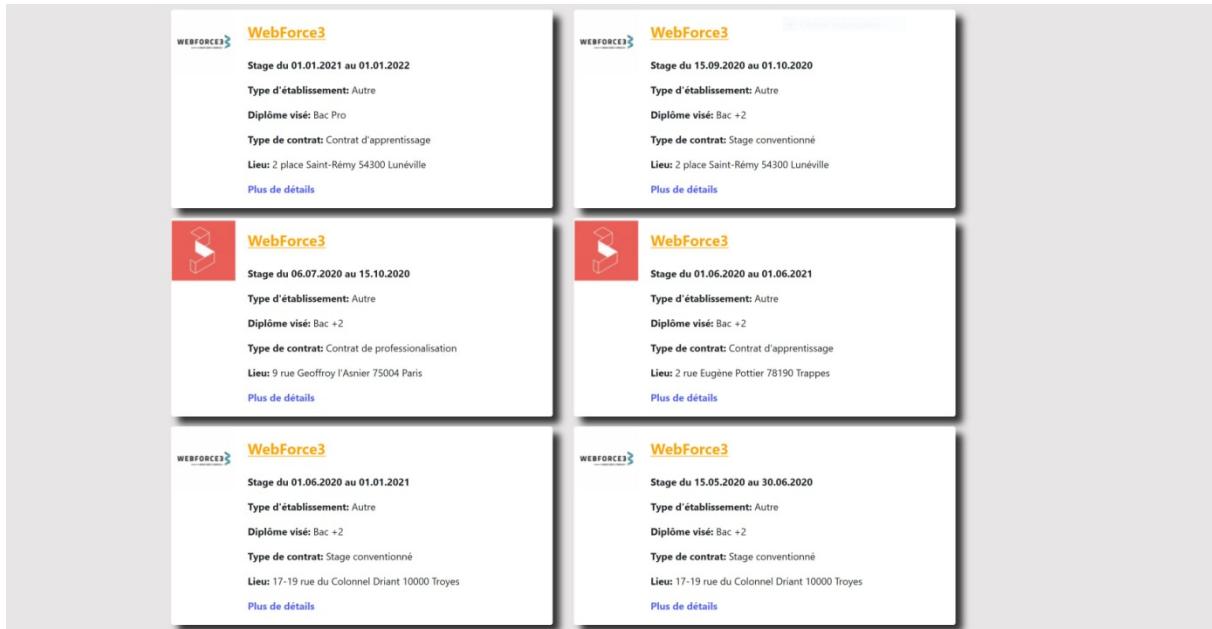
@media screen and (min-width: 992px){
  .accueil .container-fluid{
    width: 90vw;
  }
  .accueil .container{
    width: 58vw;
  }
}
```

```
@media screen and (max-width: 767.99px){
  #cache-mobile{
    display: none;
  }
  #show-mobile{
    display: block;
    padding-left: 32%;
    padding-right: 32%;
  }
}
```

# ANNEXE 4

## Cartes des résultats de recherche - ACTIVITE TYPE 1 Exemple 1

### VISUEL DES CARTES DE RESULTATS



### CSS MOUSE HOVER

```
.search-card:hover {  
    background-color: #rgba(197, 208, 255, 0.438);  
    transform: scale(1.05);  
    -webkit-transform: scale(1.05);  
    -moz-transform: scale(1.05);  
    -o-transform: scale(1.05);  
    -ms-transform: scale(1.05);  
}
```

# ANNEXE 5

## Formulaire - ACTIVITE TYPE 1 Exemple 2

### VISUEL FORMULAIRE

ENREGISTRER UN UTILISATEUR

Nom	Prenom	
Date de naissance	N° Sécurité Sociale	N° téléphone
jj / mm / aaaa		
Adresse		
Code Postal	Ville	
Email	Clé de preuve	
Commentaires		
ENVOYER		

### MESSAGE DE RESULTAT SUR UNE NOUVELLE PAGE

L'utilisateur est bien enregistré dans la Blockchain !

Hash de l'utilisateur :

7a9b848e9b6b95149d4a5d1b0dd838552dd20e567f0a8724f970dcbe7abc01a4

# ANNEXE 6

## Bundle Liip Imagine - ACTIVITE TYPE 2 Exemple 1

### DANS LE DOSSIER PACKAGES, LE FICHIER `liip_imagine.yalm`

```
# See docs how to configure the bundle: https://symfony.com/doc/current/bundles/LiipImagineBundle/basic-usage.html
liip_imagine:
    # valid drivers options include "gd" or "gmagick" or "imagick"
    driver: "gd"
    resolvers:
        default:
            web_path: ~

    filter_sets:
        cache: ~

        # the name of the "filter set"
        my_thumb_liste:
            # adjust the image quality to 75%
            quality: 75

            # List of transformations to apply (the "filters")
            filters:
                # create a thumbnail: set size to 120x90 and use the "outbound" mode
                # to crop the image when the size ratio of the input differs
                thumbnail: { size : [90, 90], mode : outbound }

        # the name of the "filter set"
        my_thumb_ecole:
            # adjust the image quality to 75%
            quality: 75

            # List of transformations to apply (the "filters")
            filters:
                # create a thumbnail: set size to 120x90 and use the "outbound" mode
                # to crop the image when the size ratio of the input differs
                thumbnail: { size : [190, 190], mode : outbound }
```

## ANNEXE 7

Pages « Détail Ecole » et « Stages » - ACTIVITE TYPE 2 Exemple 1

### AFFICHAGE DES STAGES

```
{% for stage in ecole.stage %}

<tr>
    <th scope="row" class="text-center align-middle">{{ stage.etude.nom }} </th>
    <td id="cache-mobile" class="text-center align-middle">{{ stage.diplome.nom }} </td>
    <td class="text-center align-middle">{{ stage.dateDebut | date('d.m.Y') }}</td>
    <td id="cache-mobile" class="text-center align-middle">{{ stage.dateFin | date('d.m.Y') }}</td>
    <td id="cache-mobile" class="text-center align-middle">{{ stage.contrat.type }}</td>

    {# BOUTONS SCREEN > 768px #-}
    <td id="cache-mobile" class="text-center">
        <a type="submit" class="btn border-jaune d-block mt-1" href="{{ path('user_edit_stage', { id: stage.id }) }}>Modifier</a>
        <a type="submit" class="btn btn-outline-danger d-block mt-1" href="{{ path('user_supprimer_stage', { id: stage.id }) }}>Supprimer</a>
    </td>

    {# BOUTONS SCREEN < 768px #-}
    <td id="show-mobile" class="text-center">
        {# BOUTON MODIFIER #}
        <a type="submit" class="d-block mt-2 btn-outline-warning" href="{{ path('user_edit_stage', { id: stage.id }) }}><img id="modifier"></a>
        {# BOUTON SUPPRIMER #}
        <a type="submit" class="btn-outline-danger d-block mt-2" href="{{ path('user_supprimer_stage', { id: stage.id }) }}><img id="supprimer"></a>
    </td>
</tr>
{% endfor %}
```

### FORMULAIRE ajouterStage.html.twig

```
{{ form_start(formulaire) }}

    <div class="form-group">
        {{ form_label(formulaire.date_debut) }}
        {{ form_widget(formulaire.date_debut, { 'attr': { 'class': 'form-control' } }) }}
    </div>
    <div class="form-group">
        {{ form_label(formulaire.date_fin) }}
        {{ form_widget(formulaire.date_fin, { 'attr': { 'class': 'form-control' } }) }}
    </div>
    <div class="form-group">
        {{ form_label(formulaire.etude) }}
        {{ form_widget(formulaire.etude, { 'attr': { 'class': 'form-control' } }) }}
    </div>
    <div class="form-group">
        {{ form_label(formulaire.diplome) }}
        {{ form_widget(formulaire.diplome, { 'attr': { 'class': 'form-control' } }) }}
    </div>
    <div class="form-group">
        {{ form_label(formulaire.contrat) }}
        {{ form_widget(formulaire.contrat, { 'attr': { 'class': 'form-control' } }) }}
    </div>

    <button class="btn bg-jaune">Valider</button>
{{ form_end(formulaire) }}
```

### DANS UserController.php, LA FONCTION ajoutStage()

```
/*
 * @Route("/ajout/stage/{id}", name="ajouter_stage")
 */
public function ajoutStage(Ecole $id, Request $request)
{
    // Instanciation de notre entité associé au formulaire
    $stage = new Stage();

    // Prépare le formulaire à la vue
    $formulaire = $this->createForm(StageType::class, $stage);
    $formulaire->handleRequest($request);

    // Vérifie que le formulaire est bien envoyé et que les données sont valides
    if($formulaire->isSubmitted() && $formulaire->isValid()){

        // $stage->setValid(false);
        $stage->setEcole($id);

        // On 'démarre' le manager de Doctrine
        $doctrine = $this->getDoctrine()->getManager();
        $doctrine->persist($stage);
        $doctrine->flush();

        // Prépare un message à afficher côté vue
        $this->addFlash('success', 'Votre stage est bien enregistré !');
    }

    return $this->render('user/ajouterStage.html.twig', [
        'formulaire' => $formulaire->createView(),
        'id_ecole' => $id->getId()
    ]);
}
```

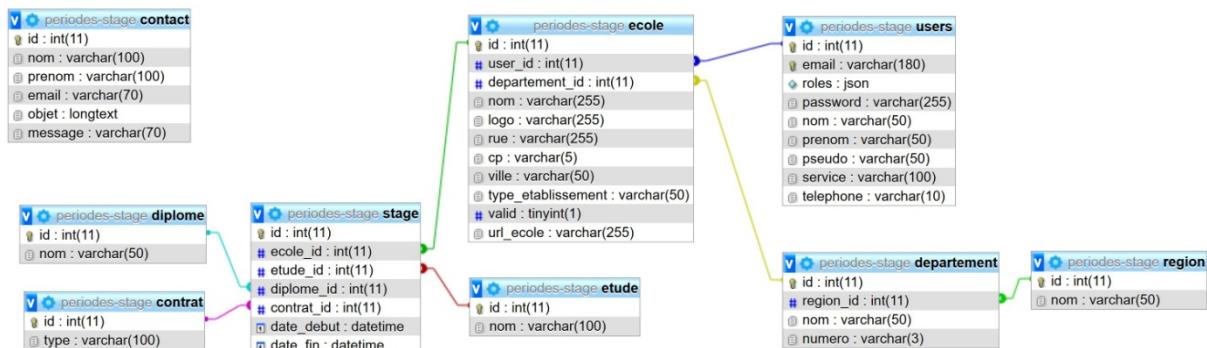
# ANNEXE 8

## Liens de retour + BDD - ACTIVITE TYPE 2 Exemple 1

### UN EXEMPLE DE LIEN DE RETOUR

```
{# TITRE #}
<div class="col-12 lg-mb-5 retour" >
    <a class="ml-lg-5" href="{{ path('user_index') }}><i class="fas fa-arrow-circle-left"></i> Retour à votre
        espace personnel</a>
    <h1 class="text-center font-weight-bolder pt-3 mb-5 txt-bleu-clair">Espace {{ ecole.nom }}</h1>
</div>
```

### BASE DE DONNEES



# ANNEXE 9

## BKP Version 1 - ACTIVITE TYPE 2 Exemple 2

### INSERTION BDD

```
// Formulaire envoyé
if(isset($_POST['submit'])){

    // Vérification des champs (envoyés et non vides)
    if(isset($_POST['nom']) && !empty($_POST['nom']))
        && isset($_POST['prenom']) && !empty($_POST['prenom'])
        && isset($_POST['date_de_naissance']) && !empty($_POST['date_de_naissance'])
        && isset($_POST['secu']) && !empty($_POST['secu'])
        && isset($_POST['email']) && !empty($_POST['email'])
        && isset($_POST['adresse']) && !empty($_POST['adresse'])
        && isset($_POST['code_postal']) && !empty($_POST['code_postal'])
        && isset($_POST['ville']) && !empty($_POST['ville'])
        && isset($_POST['telephone']) && !empty($_POST['telephone'])
        && isset($_POST['texte']) && !empty($_POST['texte'])
        && isset($_POST['cle']) && !empty($_POST['cle'])
    ){

        // Encodage utf8
        $nom = trim_utf8_encode($_POST['nom']);
        $prenom = trim_utf8_encode($_POST['prenom']);
        $date_de_naissance = trim_utf8_encode($_POST['date_de_naissance']);
        $secu = trim_utf8_encode($_POST['secu']);
        $email = trim_utf8_encode($_POST['email']);
        $adresse = trim_utf8_encode($_POST['adresse']);
        $code_postal = trim_utf8_encode($_POST['code_postal']);
        $ville = trim_utf8_encode($_POST['ville']);
        $telephone = trim_utf8_encode($_POST['telephone']);
        $texte = trim_utf8_encode($_POST['texte']);
        $cle = trim_utf8_encode($_POST['cle']);
    }
}
```

```
// Vérifie si l'adresse email n'est pas déjà enregistrée
$requete = $db->prepare("SELECT COUNT(id) AS nb_email FROM utilisateurs WHERE email = :email");
$requete->bindValue(':email', $email, PDO::PARAM_STR);
$requete->execute();

// Récupère le résultat de la requête ci-dessus
$resultat = $requete->fetch();

// Si l'adresse email n'existe pas...
if($resultat['nb_email'] == 0) {

    // ... On vérifie que le code postal fait 5 caractères
    if(strlen($_POST['code_postal']) == 5){

        // ... On vérifie que le n° de sécu fait 15 caractères
        if(strlen($_POST['secu']) == 15){

            // ... On vérifie que le n° de téléphone fait 10 caractères
            if(strlen($_POST['telephone']) == 10){

                // ... et on prépare la requête
                $requete = $db->prepare(
                    "INSERT INTO utilisateurs (nom, prenom, date_de_naissance, secu, email,
                     adresse, code_postal, ville, telephone, texte)
                     VALUES (:nom, :prenom, :date_de_naissance, :secu, :email, :adresse,
                     :code_postal, :ville, :telephone, :texte)
                     ");
            }
        }
    }
}
```

```

        $requete->bindValue(':nom', $nom, PDO::PARAM_STR);
        $requete->bindValue(':prenom', $prenom, PDO::PARAM_STR);
        $requete->bindValue(':date_de_naissance', $date_de_naissance, PDO::PARAM_STR);
        $requete->bindValue(':secu', $secu, PDO::PARAM_STR);
        $requete->bindValue(':email', $email, PDO::PARAM_STR);
        $requete->bindValue(':adresse', $adresse, PDO::PARAM_STR);
        $requete->bindValue(':code_postal', $code_postal, PDO::PARAM_STR);
        $requete->bindValue(':ville', $ville, PDO::PARAM_STR);
        $requete->bindValue(':telephone', $telephone, PDO::PARAM_STR);
        $requete->bindValue(':texte', $texte, PDO::PARAM_STR);

        $requete->execute();
    
```

## ECRITURE BLOCKCHAIN

```


/*
 * Blockchain ---
 */

// Informations de base
$platformName = 'SpuroBKP';
$time = $fw->ledger->getTime();
$apiData = [
    'prenom' => urlencode($prenom),
    'nom' => urlencode($nom),
];

// Hash des informations voulues
$hash = hash('sha256', strtolower( $nom . $prenom . urlencode($date_de_naissance) . $secu . $email . $cle));
$hashHash = hash('sha256', $hash); // Signature électronique (hash du hash)

// Stocker des informations dans un tableau
$apiData = [
    'prenom' => urlencode($prenom),
    'nom' => urlencode($nom),
    'user_hash' => $hashHash
];

// Encode le tableau ci-dessus en JSON
$jsonData = json_encode($apiData);

// Stocke le hash, le timestamp et des données dans un tableau
$document = [
    'hash' => $hash,
    'date' => $time,
    'data' => $jsonData
];

// Génère un print et le sauvegarde
$print = $fw->ledger->generatePrint($hash, $time, $platformName);
$fw->ledger->savePrint($print, $document);

// Création du block et mise à jour du registre
$fw->ledger->saveBlock();
$fw->ledger->handleUpdate();

/*
 * Fin ---
 */


```

# ANNEXE 10

## BKP Version 3 avec AJAX - ACTIVITE TYPE 2 Exemple 2

### SCRIPT AJAX

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"></script>

<script>

    // SCRIPT AJAX POUR EMPECHER LE RECHARGEMENT DE LA PAGE

    // On va chercher l'ID du formulaire
    $("#form").ready(function(){

        // Si le formulaire est validé (click sur le bouton)
        $("#submit").click(function(e){
            e.preventDefault(); // on empêche le bouton d'envoyer le formulaire

            // La méthode .post() permet d'envoyer des données vers le serveur avec une requête HTTP POST
            $.post(
                'insertion.php', // URL du script PHP qui traite l'insertion en BDD
                {
                    // Récupération de la valeur des inputs à passer à insertion.php
                    nom : $("#nom").val(),
                    prenom : $("#prenom").val(),
                    date_de_naissance : $("#date_de_naissance").val(),
                    secu : $("#secu").val(),
                    telephone : $("#telephone").val(),
                    adresse : $("#adresse").val(),
                    code_postal : $("#code_postal").val(),
                    ville : $("#ville").val(),
                    email : $("#email").val(),
                    texte : $("#texte").val()
                },
                // Indique une fonction à exécuter lorsque la méthode est terminée
                function(data){
                    // "data" contient les données résultant de la demande
                    if(data == 'Error'){
                        // ERROR : message si erreur
                        $("#resultat").html("<div>Erreur lors de la connexion</div>");
                        $('#resultat').fadeIn().html(data);
                        setTimeout(function() {
                            $('#resultat').fadeOut("slow");
                        }, 10000 );
                    }
                    else{
                        // SUCCESS : message si réussite
                        $("#resultat").html("<div>L'utilisateur été enregistré avec succès !</div>");
                        $('#resultat').fadeIn().html(data);
                        setTimeout(function() {
                            $('#resultat').fadeOut("slow");
                        }, 10000 );
                    }
                },
                'text' //
            );
        });
    });

</script>
```

# ANNEXE 11

## BKP Ecriture dans le Registre - ACTIVITE TYPE 2 Exemple 2

### EXPLICATION DE L'ECRITURE DANS LE REGISTRE

Le Registre est hébergé sur chaque ordinateur faisant partie du réseau de la Blockchain.

Il est mis à jour en temps réel (REGISTRE GLOBAL). Mais concernant la Blockchain Spuro il y a également un Registre Local qui me permet d'effectuer des tests sur serveur local.

```
b3035dc54c4a1cedec708d6f8c17bb6d3fd99ba119b8cb3bd9608c8c47fb7672 1571398652 data:{ "prenom": "X", "nom": "X",  
"user_hash": "dbb77b6e593549d34e38f7b43558bc67b9861df6aeebd10dce65816d96d2e88"  
06cdffd13a8028933fbdf86175b913fb100d38245d39561613307feeee9dd7c1d 1571398652  
733453c6144b78deba215f7e32ec1b1bdb3e4ae1e9b44d8515bafe223e697333 1571398652  
ceb35704a6dcf9068a7d1837d34734102f1813061634c8321b953e4f3f3c874d5 1571398943 data:{ "prenom": "W", "nom": "W",  
"user_hash": "d8705604618848eeb3754f6f2fa7bedb3af5ad4ab0105b7306733bc5a7e39b62"  
4c43bec19e8032bfcce4ade8aae869de2940e7ceccf0df66779e7f07ea36c662 1571398943  
3529cc931c6b1d389fa247596971878e4241c0f62032fab8c8bb3143d59b700a 1571398943  
5353577c9af1f887284f00f10d296c3af089d8998f7cd582eef43aaef914aed2 1571645600 data:{ "prenom": "000", "nom": "000",  
"user_hash": "73b547d3b20273581fb16a944f0da9fd159565eb8d40e591b3da365630d5593a"}  
270dc10e3beb227b5ec38691c3e00756158e1dc4a8d5d02168a57b9023f8a74a 1571645600  
a79860b41699b7ae1f30725c3045aa51837e9169e5dc1d268a7d7585813aa57 1571645600  
ebf46d1b616a48a9f2e0fabae44e42275cf09dd8f34a8a8b261b78fe89ba264 1571901321 data:{ "prenom": "XYZ", "nom": "XYZ",  
"user_hash": "28da960d76bcd57a331fd033ff378aec9bdf5e93dad6481fd38742cce8e67d56"  
a3e31b42e4457e03ccac6873c854220b914d6bc5aa133666c88a19dd5bb79baf 1571901321  
1bbc8f8564fde3d0ba5145b02a82ad115a4ef9d6e143f887b9ca5fb99977f4 1571901321  
ec8e69136875540eb8832e300e1a53b3a911c196cba0e62eb5cf927c3120 1572127870 data:{ "prenom": "azerty", "nom": "azerty",  
"user_hash": "967b8db5af3360eee958cc1770b489314ff804263250ab2fb69745d39fe5a402"}  
8dfc7bd089fa25dc059273861c2b31a603064a8f723e8151331108c0c3ac09cb 1572127870  
565a72f611b2192376310b5b2225cdc67b876dd6e5089ad946e95909e10ab060 1572127870  
7802b9b99f21329090903dcf6f8e146e35d6f513d38847154100320042ca7b1e 1572129639 data:{ "prenom": "zefzefze", "nom": "szdzdfzef",  
"user_hash": "15b5508720d8cd2f7e26ca529950da4ff1d43379485ee756d6fd3652e3122a76"}  
e47bfd7fdec52b4f2d62df706a9598d9ff176d7ffa479da09e18374693ab74f 1572129639  
9a081094711fc226b74995010a06bf242d48505db04142648bde5bb038f93bd3 1572129639  
438385d1aeb60e7366b68e807f2d14995e0623147bd9ae27d84e48e4a9d57c76 1572511290 data:{ "prenom": "HARRY", "nom": "POTTER",  
"user_hash": "89a8a5d1c532a5d2688161c7f4a2ce9768095549f6001554ebbef1edeed49e36"}  
70aecb2ec9a906e8dd4678ddb44c0964c279d00ede387929c2365046a9222f 1572511290  
15c167af5eaf9a341516af07a62ab2adc1f030aab967f63bce625723a03aa5b9 1572511290
```

Un Block fait 4 lignes

- **En jaune** : le hash de toutes les données du formulaire
- **En rouge** : l'horodatage
- **En violet** : le Nom et le Prénom de l'utilisateur sont en clairs pour pouvoir retrouver la signature électronique correspondant à l'identité d'un utilisateur
- **En vert** : le hash de l'utilisateur, sa signature électronique (certaines données du formulaire dont la clé de preuve)
- **En orange** : le hash du Block
- **En bleu** : le hash de tous les Blocks du Registre