Une image contenant extérieur, volant, cerf-volant, bleu

Description générée automatiquement

**Documentation technique**

ToDo & Co

Version 1.0

00/00/0000

1. Sommaire

[1. Sommaire 1](#_Toc183274690)

[2. Contexte 2](#_Toc183274691)

[3. Problématique 3](#_Toc183274692)

[a) Migration technique d’un projet client 3](#_Toc183274693)

[b) Correction des anomalies sur un projet existant 3](#_Toc183274694)

[c) Nouvelle version de l’application « INGENEO » 3](#_Toc183274695)

[4. Missions confiées 4](#_Toc183274696)

[d) Missions pour la Problématique 1 : Migration technique d’un projet client 4](#_Toc183274697)

[e) Missions pour la Problématique 2 : Correction des anomalies sur un projet existant 12](#_Toc183274698)

[f) Missions pour la Problématique 3 : Nouvelle version de l’application « INGENEO » 17](#_Toc183274699)

[5. Bilan 21](#_Toc183274700)

1. Contexte

Je m'appelle Karol ZIELINSKI, j'ai 31 ans, et je suis actuellement en alternance dans le cadre de ma reconversion professionnelle vers le métier de développeur d'applications spécialisées en PHP et Symfony. Après plusieurs années d'expérience dans un autre secteur, j'ai décidé d'entamer cette nouvelle voie pour obtenir mon diplôme et évoluer dans un domaine qui me passionne. Cette alternance s'inscrit dans mon parcours de formation, et il est essentiel pour valider mes compétences et acquérir une expérience concrète.

L'organisation qui m'accueille pour cette alternance est Septeo Solutions Experts-Comptables, une entreprise qui propose des solutions logicielles dédiées aux professionnels de l'expertise comptable. Leur principal produit, INGENEO, offre un ensemble d'outils permettant aux experts-comptables de gérer efficacement et en toute simplicité les processus comptables, avec des fonctionnalités telles que la collecte automatique des factures, la reconnaissance et l'imputation des écritures comptables, ainsi que l'intégration fluide avec des logiciels de production comptable. En plus de faciliter la gestion comptable, l'entreprise permet aux experts de valoriser leur expertise auprès de leurs clients grâce à des modules comme les indicateurs de performance personnalisés et un espace client dédié, qui inclut des fonctionnalités telles que la gestion des documents (GED), la facturation et le suivi de la conformité à la facturation électronique.

J'ai choisi cette entreprise pour mon stage car j'y travaille déjà depuis janvier 2019 et j'ai eu l'opportunité de signer un contrat d'alternance avec eux à partir de janvier 2023, dans le cadre de mon projet de reconversion. Cette entreprise me permet de concilier l'acquisition de compétences professionnelles tout en poursuivant mon évolution interne.

Les objectifs que je me suis fixés pour ce stage sont principalement axés sur la montée en compétence dans le développement d'applications PHP/Symfony. Mon but est d'acquérir une expertise plus pointue dans ce domaine pour gagner rapidement en autonomie et en expérience, tout en participant activement aux projets de développement de l'entreprise.

1. Problématique
2. Migration technique d’un projet client

*Comment assurer une migration fluide d’un projet client vers une version technologique plus récente sans impact sur les fonctionnalités existantes et les performances ?*

La migration d’un projet client est une tâche délicate qui présente souvent des défis en termes de compatibilité et de continuité de service. En tant que développeur, un des enjeux est de garantir que la migration vers une version plus récente du Framework ne crée pas de rupture dans les fonctionnalités. Cette problématique implique de minimiser les risques de régressions et d’assurer que le système continue à répondre aux besoins fonctionnels et de performance du client.

1. Correction des anomalies sur un projet existant

*Comment optimiser le processus de correction des anomalies dans un projet existant pour garantir la stabilité et la qualité de service du produit ?*

Les projets existants peuvent accumuler des anomalies en raison de divers facteurs (changements de code, mises à jour non documentées, évolutions fonctionnelles, etc.). En tant que développeur, il est crucial de mettre en place une méthodologie efficace de gestion des anomalies qui permet de corriger rapidement les bugs sans créer de nouveaux dysfonctionnements. Ce processus peut inclure des tests unitaires, des revues de code et une analyse approfondie pour améliorer la robustesse et la maintenabilité du projet.

1. Nouvelle version de l’application « INGENEO »

*Comment structurer efficacement le développement collaboratif de la nouvelle version de l’application « INGENEO » pour garantir un suivi optimal et une qualité de livraison ?*

Le développement d’une nouvelle version de l’application « INGENEO » nécessite une organisation minutieuse et une communication constante entre les membres de l’équipe. Afin de faciliter le suivi de l’avancement et de structurer le travail, plusieurs pratiques et outils ont été mis en place : des réunions quotidiennes pour synchroniser les tâches et suivre l’état d’avancement de chacun, des sprints de deux semaines pour favoriser la réactivité et le feedback, ainsi que l’utilisation d’Azure DevOps pour la gestion des tâches et des versions. Le dépôt est organisé avec deux branches principales : une branche « develop » sur laquelle tous les développeurs collaborent, et une branche « main » représentant la version en production. Ce cadre méthodologique permet à l’équipe d’assurer une gestion structurée et d’améliorer la qualité des livraisons finales.

1. Missions confiées
2. Missions pour la Problématique 1 : Migration technique d’un projet client

Pour la problématique de migration technique, les missions avaient pour but de garantir une transition fluide vers une version technologique plus récente sans affecter les fonctionnalités en place. Voici les étapes et missions principales :

1. **Analyse des besoins de migration** :

**Objectif** : Comprendre les spécificités de l’infrastructure actuelle du projet et définir les besoins pour la nouvelle version.

**Tâches** :

* Étudier la configuration actuelle du projet client, ses dépendances et ses particularités (compatibilité, modules spécifiques).

Pour commencer, j’ai étudié la configuration actuelle du projet client en installant le projet initial sur mon ordinateur. J’ai récupéré le projet sur GitHub avec la commande suivante :   
  
***« git clone https://adresse-du-repository.git »***

Je me suis rendu dans le composer.json pour voir les packages d’installés et connaitre les versions utilisées pour chaque dépendances et la version de PHP et de Symfony.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Dans le fichier composer.json, j’ai pu retrouver les informations dont j’avais besoin pour connaitre les versions actuelles du projet et pouvoir préparer la suite de la migration :   
  
php en version inférieur ou égale à la 5.5.9  
symfony en version 3.1

Et également les librairies utilisées :  
  
doctrine/orm en version 2.5  
doctrine/doctrine-bundle en version 1.6   
doctrine/doctrine-cache-bundle en version 1.2

1. **Préparation de l’environnement de migration** :

**Objectif : Créer un environnement de test représentatif pour effectuer les essais de migration sans impacter l’environnement de production.**

**Tâches :**

* **Configurer un environnement de préproduction similaire à la production.**

Pour configurer un environnement de préproduction qui serait similaire à la production, j’ai demandé en interne aux DevOps avec la confirmation du Directeur Technique pour qui me mette en place un serveur sur lequel je pourrais travailler qui serait en ligne et que chaque modification sur mon projet local soit testée également en production grâce à deux branch git que j’utiliserais pour mon travail. Donc en local une branch « features/ » et une autre branch « main » qui servira quant à elle, a la branch principal du projet en préproduction.

* Mettre en place une base de données de test pour simuler les opérations en production.

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquementJ’ai mis en place une base de données de teste avec des DataFxitures pour pouvoir simuler chaques fonctions du projet. Pour ce faire j’ai installé une librairie « fakerphp/faker » qui permet de remplir des éléments aléatoires en base de données. Voici une capture d’écran de comment a été mise en place la base de données :

On peut voir la capture d’écran que j’utilise la librairie « faker » par exemple pour remplir le contenu d’une tâche, j’ai utilisé « $faker->text() » qui me génère un texte aléatoire pour le contenu d’une tâche.  
  
Lorsque tout est en place pour remplir chaque tables de la base de données, il ne reste plus qu’à lancer la commande « php bin/console doctrine:fixtures:load ».

Voici le résultat final en base de donnée :   
  
Une image contenant texte, Police, nombre, logiciel

Description générée automatiquement

Les tâches sont bien présentes et on retrouve bien le contenu des tâches avec la librairie « faker ».

* Installer et configurer les dépendances nécessaires, notamment les nouvelles versions des outils ou bibliothèques PHP/Symfony.

Une image contenant texte, capture d’écran

Description générée automatiquement

Aux vues de toutes les montées en version à faire et de la taille du projet et de ses dépendances, j’ai décidé de partir directement sur un projet vierge avec la version de PHP 8.2 et de Symfony 6.4 et ensuite d’installer les librairies utilisées avec les versions compatibles.

1. **Adaptation du code aux nouvelles versions** :

**Objectif** : Mettre à jour le code pour qu’il soit compatible avec les nouvelles versions des outils et frameworks.

**Tâches** :

* Identifier et modifier les fonctions obsolètes ou incompatibles avec la nouvelle version de Symfony ou des packages utilisés.

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel

Description générée automatiquement  
Sachant que la version Symfony initiale était la version 3.1 les annotations étaient utilisées dans l’ensemble du code mais que sur la version 6.4 de Symfony et de la version 8 de PHP l’utilisation des attribues a été recommandé, j’ai dû modifier les controllers, les entitées pour bien mettre à jour le projet.  
  
  
Voici les captures d’écrans d’une entité du passage des annotations en attribut :  
  
Symfony 3.1 : Symfony 6.4 :

1. **Tests de régression et validation** :

**Objectif** : Vérifier que la migration ne crée pas de nouvelles anomalies et que les fonctionnalités restent intactes.

**Tâches** :

* Effectuer des tests de régression complets pour valider le bon fonctionnement des fonctionnalités principales.

J’ai installé le package PhpUnit pour réaliser les tests unitaires et fonctionnels de l’application et valider les fonctionnements de l’intégralité du code avec cette commande :   
  
**« composer require phpunit/phpunit »**

Suite à l’installation de PhpUnit, je me suis informé grâce la documentation que j’ai trouvé sur le site officiel de PhpUnit (<https://docs.phpunit.de/en/11.4/installation.html>) pour la configuration et le fonctionnement de du package.

Une image contenant texte, capture d’écran

Description générée automatiquement

J’ai du créer un dossier « tests » avec comme sous-dossier « Controller » , « Entity », « Form » pour réaliser les tests de l’entièreté du projet.

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquementDans ce fichier, on peut voir que je test les fonctionnalités de l’entité « Task » et de toutes ces fonctions.   
  
Pour chaque test réalisé, je voulais voir le résultat et je voulais que les tests réalisés couvrent tous le code.   
  
Pour lancer les tests et avoir un résultat visuel, l’utilisation de xDebug était requise et pour sa mise en place, j’ai suivi la documentation du site officiel (xdebug.org).  
  
Lorsque tout à était mis en place, il ne me rester plus qu’à lancer la commande suivante pour réaliser les tests et avoir le rapport final :   
  
« **vendor/bin/phpunit --coverage-html public/test-coverage** »



Maintenant, pour accéder aux résultats visuels de mes tests, il fallait accéder au lien suivant :   
  
**« http://127.0.0.1:8000/test-coverage/ »**

J’ai cliqué sur l’entité « Task » et je peux voir que l’intégralité des fonctions était testée avec un taux de coverage à 100%.

* Réaliser des tests de performance pour garantir que les temps de réponse restent conformes.

Pour les tests de performance, j’ai décider d’utiliser le profiler de Symfony qui m’a donnée les résultats attendus.   
  
Les tests ont été réalisé sur toutes les pages accessibles de l’application, voici une capture d’écran qui détails les performances :   
  
  
  
Ici nous voyons que le temps d’exécution total de la page est de 760 ms ce qui est correct pour une application web qui ne comporte pas beaucoup de données et qui est une application assez simple.   
  
***NB : Le temps cible pour ce genre d’application est entre 500 et 1500ms.***

1. **Documentation et mise en production** :

**Objectif** : Documenter les étapes de la migration et les configurations pour faciliter la maintenance future.

**Tâches** :

* Rédiger un guide de migration pour les futurs développements et mises à jour.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Pour les futurs développeurs, j’ai mis en place une documentation pour contribuer au projet avec ma méthodologie. Ce fichier est présent dans le projet est se nomme : **CONTRIBUTING.md**J’ai également rédigé un guide de migration pour les futurs développeurs.  
  
Cette documentation retrace toutes les étapes de la migration d’un projet Symfony 3.1 vers la version 6.4 de Symfony.   
  
Cette documentation est présente en cliquant sur le lien suivant :  
  
  
  
Elle retrace chaque étape à réaliser pour arriver à la version finale tout en ayant chaque fonctionnalité fonctionnelle.

1. Missions pour la Problématique 2 : Correction des anomalies sur un projet existant

Les missions autour de la correction des anomalies visaient à optimiser le processus de détection et de correction des bugs pour assurer la stabilité et la qualité de service. Voici les étapes de cette mission :

1. **Identification et priorisation des anomalies** :

**Objectif** : Dresser un inventaire des anomalies en fonction de leur criticité pour le produit et l’expérience utilisateur.

**Tâches** :

* + - Classifier les anomalies par niveau de priorité (bloquants, majeurs, mineurs).

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquementNous avons 4 niveaux de priorités :  
  
4 – Low : Des corrections minimes comme une faute d’orthographe.

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement3 – Medium : Des petits ajustements dans les fonctionnalités de l’application  
  
2 – High : Fonctionnalité partiellement fonctionnel ou critique pour un client et demande urgente  
  
1 – Critical : Application ou fonctionnalité totalement inutilisable.

Pour classer les tâches selon leur niveau de criticité, nous nous basons sur les critères cités au-dessus.

1. **Analyse et reproduction des bugs** :

**Objectif** : Reproduire les erreurs signalées pour mieux comprendre leurs causes.

**Tâches** :

* Trouver la provenance de la problématique.

Voici une des erreurs présentes sur le projet :   
  
Une image contenant texte, logiciel, Page web, Site web

Description générée automatiquement  
  
Pour résumé la problématique, sur le projet il est possible de créer des classeurs de différent type, « Achat », « Vente », « Note de frais » etc… Pour les classeurs d’achat et de vente, il y a la possibilité de lier un des journaux au classeur. Ces journaux ont également un type « Achat », « Vente » ou « Autres ». Dans ce cas, si on sélectionne un classeur de type « Vente » alors on devrait avoir uniquement les journaux de type « Vente » présents dans le formulaire mais ce n’est pas le cas actuellement.   
  
Pour trouver l’erreur, je savais qu’il manquait un filtrage au niveau des journaux dans le controller et également dans le formulaire car actuellement, on a le choix des tous les journaux.  
  
  
  
Pour trouver le controller qui géré cette partie du code, j’ai fais une recherche dans le code pour me trouver celui qui correspond aux classeurs, le résultat le plus probable était donc « ClasseurController.php », Le formulaire pour le choix des journaux ne s’ouvre qu’à l’édition du classeur, donc j’ai vérifié s’il y avait une fonction « edit ». Ce qui est bien le cas.  
  
Dans cette fonction edit, j’ai regardé à quel moment le formulaire était appelé et vérifié les paramètres passés.

Je sais que le type de classeur n’est pas de TYPE\_CLASSEMENT et donc je me que le forumaire est le « ClasseurType », je vois que des paramètres sont envoyé vers ce formulaire à partir de la fonction « getFormParams ».

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

J’ai donc trouvé et vérifié la fonction getFormsParams pour voir ce qu’elles renvoient au formulaire.  
  
Le getFormsParams renvoie l’intégralité des journaux et non pas par type, j’ai donc trouvé ou je devais faire la modification pour résoudre la problématique.

1. **Correction des anomalies** :

**Objectif** : Résoudre les problèmes identifiés en apportant des modifications de code tout en évitant les régressions.

**Tâches** :

* Corriger les erreurs identifiées en respectant les bonnes pratiques et les standards de l’équipe.

Je vois qu’il y a différents paramètres envoyés et pour répondre aux besoins du ticket, j’ai ajouté un tri des journaux pour ne récupérer que les journaux de type « TYPE\_ACHAT » et les journaux de type « TYPE\_VENTE » pour envoyer ces informations dans le formulaire du classeur.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

A la fin de cette fonction, on retourne les éléments en option :  
  
Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Et pour finir, dans mon formulaire ClasseurType, je mets une condition que si j’ouvre un classeur de type « Achat » alors je veux récupérer les journaux de type « Achat » et de même pour les classeurs de type « Vente » :  
  
Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

1. Missions pour la Problématique 3 : Nouvelle version de l’application « INGENEO »
2. **Mise en place et utilisation des réunions quotidiennes** :

**Objectif** : Assurer une communication continue entre les membres de l’équipe pour coordonner l’avancement du projet.

**Tâches** :

* Participer à des réunions de type "stand-up" chaque matin pour partager l’avancement des tâches, identifier les obstacles et coordonner les priorités.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquementNotre directeur technique à mis en place une réunion quotidienne tous les matins pour parler de l’avancement de chaque tâche, identifier les problématiques bloquants et coordonner les priorités. Cette réunion est placée le matin à 9h30.

1. **Suivi des tâches et gestion des versions via Azure DevOps** :

**Objectif** : Utiliser un outil de gestion de projet pour suivre l’avancement des tâches, assigner les responsabilités et documenter les progrès.

**Tâches** :

* Utiliser Azure DevOps pour créer et assigner des tickets de tâches, en les organisant par ordre de priorité et par phase de développement.

Nous utilisons Azure DevOps comme outils de ticket. Plusieurs colonnes sont mises en place pour bien différentier l’état de chaque ticket : « Bugs & recette », « US (TODO) », « En cours », « A merge », « A recetter », « Validé ».  
  
Voici en détails à quoi servent chaque onglet :  
  
**Bugs & recette** : Utilisé lorsqu’un bug et découvert dans l’application ou un ticket a été traité mais que le fonctionnement attendu n’est pas correct.  
  
**US (TODO)** : Utilisé pour les nouvelles fonctionnalités de l’application  
  
**En cours** : Utilisé lorsqu’on travaille sur un des tickets.   
  
**A merge** : Utilisé lorsqu’une tâche est terminée mais les modifications n’ont pas encore été transmisse à la branch principale.  
  
**A recetter** : Utilisé lorsque le merge à été fait et que la personne qui s’occupe de cette partie vérifie le bon fonctionnement après le travail effectué par le développeur.  
  
**Validé**: Utilisé si le fonctionnement attendu est correct et que la tâche a été traité en intégralité.

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

1. **Gestion du dépôt de code et de la stratégie de branches** :

**Objectif** : Garantir une gestion ordonnée du code et des versions pour réduire les risques d’erreurs en production.

**Tâches** :

* Contribuer au code en utilisant deux branches principales sur Azure DevOps : la branche « develop » pour le développement collaboratif, et la branche « main » pour la version de production.

Actuellement, nous travaillons sur 2 branch principale :  
  
« develop » : Qui est utilisé en local pour les nouveaux développements en cours.  
  
« main » : Qui est utilisé comme branch principal pour l’applications en production.   
  
  
Pour contribuer au projet, nous mettons à jour notre branch « develop » en local, puis nous travaillons sur une tâche spécifique », lorsque le travail est terminé, nous avons mis en place une nomenclature pour les commits qui est la suivante :  
  
 **git commit -m « NuméroDuTicket – Description du ticket »**

Dès que le dernier commit est push sur la branch develop et que le travail est terminé et que le ticket est passé dans l’onglet « A merge », alors un merge est demandé pour push les modifications de la branch develop sur la branch main.

1. Bilan

**Introduction**

Mon parcours au sein de Septeo Solutions Experts-Comptables a marqué un tournant important dans ma carrière professionnelle. À 31 ans, j’ai entrepris une reconversion vers le métier de développeur d’applications PHP/Symfony, et cette alternance a été l’occasion de transformer ma passion pour le développement en une compétence concrète. Ces deux années m’ont permis de construire les bases solides d’une nouvelle carrière et d’intégrer pleinement un secteur qui allie innovation et rigueur technique.

**Évolution des compétences techniques**

Durant ces deux années, j’ai bénéficié d’une montée en compétence progressive, structurée par les problématiques et projets confiés. Les missions m’ont offert l’opportunité de travailler sur des aspects variés du développement logiciel, me permettant ainsi d’acquérir une expertise approfondie dans des domaines clés :

1. **Maîtrise de PHP/Symfony** :
   * J’ai renforcé mes bases en développement back-end en PHP et Symfony, en participant activement à des projets complexes tels que la migration technique d’un projet client ou la création d’une nouvelle version d’INGENEO.
   * J’ai développé des fonctionnalités robustes tout en respectant les standards de qualité de l’entreprise, notamment à travers l’utilisation des tests unitaires et fonctionnels.
2. **Gestion de projets collaboratifs** :
   * J’ai intégré des méthodologies agiles, avec des sprints de deux semaines, des réunions quotidiennes et une organisation rigoureuse via Azure DevOps.
   * J’ai appris à travailler efficacement en équipe grâce à une gestion structurée des branches de code (main et develop) et des revues de code collaboratives.
3. **Approche méthodologique pour la correction et l’optimisation** :
   * J’ai amélioré mes compétences en résolution de problèmes, en gérant des anomalies sur des projets existants. Cela m’a permis de développer des méthodes rigoureuses d’analyse et de correction des bugs.
4. **Expérience dans la gestion des versions et des migrations** :
   * La migration technique d’un projet client m’a permis de mieux comprendre les défis liés à l’évolution technologique d’une application, notamment en ce qui concerne la compatibilité et la performance.

**Contributions aux projets de l’entreprise**

Mon intégration dans les projets de Septeo a été marquée par des contributions significatives, notamment dans le cadre du développement de **l’application INGENEO**, le produit phare de l’entreprise. J’ai participé activement à :

* **L’amélioration continue du produit** :
  + En corrigeant des anomalies sur des modules existants pour assurer la stabilité et la satisfaction des utilisateurs finaux.
  + En optimisant des fonctionnalités pour répondre aux exigences du marché.
* **Le développement de la nouvelle version d’INGENEO** :
  + En collaborant avec l’équipe pour structurer le développement et assurer un suivi rigoureux.
  + En proposant des solutions techniques adaptées aux besoins spécifiques des experts-comptables.

**Développement personnel et professionnel**

En parallèle des compétences techniques, cette expérience a aussi marqué mon développement personnel :

1. **Autonomie et prise d’initiative** :
   * Je suis devenu plus autonome dans la gestion de mes tâches, capable de proposer des solutions aux problématiques rencontrées.
2. **Adaptabilité et gestion des priorités** :
   * La diversité des projets m’a appris à m’adapter rapidement aux changements et à gérer efficacement mes priorités.
3. **Communication et collaboration** :
   * J’ai développé mes compétences en communication grâce aux réunions quotidiennes et aux revues de code. Cela m’a permis de mieux comprendre les attentes de l’équipe et de contribuer de manière proactive.

**Perspectives et ambitions futures**

Cette alternance a confirmé ma passion pour le développement d’applications et m’a donné les outils nécessaires pour réussir dans ce domaine. Mes objectifs futurs incluent :

* **Continuer à approfondir mes compétences techniques** :
  + Notamment en explorant des domaines comme les API REST, la sécurité des applications web ou encore l’optimisation des performances.
* **Participer à des projets innovants** :
  + En proposant des idées et des solutions qui allient créativité et rigueur technique.
* **Évoluer vers des responsabilités accrues** :
  + Mon objectif est de gagner en leadership technique pour, à terme, encadrer des projets ou accompagner d’autres développeurs.

**Conclusion**

Ces deux années chez Septeo Solutions Experts-Comptables ont été une expérience déterminante dans ma reconversion professionnelle. Elles m’ont permis de transformer mes ambitions en réalisations concrètes, tout en me préparant à évoluer dans un secteur exigeant et stimulant. Grâce au soutien de l’entreprise et à mon implication, je quitte cette alternance avec des bases solides et la confiance nécessaire pour continuer à progresser en tant que développeur d’applications PHP/Symfony.