RAPPORT DE STAGE

Sommaire

Introduction *(≈2 pages)*

Contexte et présentation de l'entreprise

Dans le cadre de mon stage académique d'une durée d'un mois, j'ai eu l'opportunité d'intégrer Iqera, une entreprise spécialisée dans [ajouter secteur exact : recouvrement de créances, services financiers, IT, etc.]. L'entreprise met en place des solutions technologiques innovantes afin d'optimiser la gestion des processus métiers et de renforcer la relation client. Dans ce contexte, l'innovation, la performance et la fiabilité des systèmes informatiques occupent une place centrale dans sa stratégie de développement.

Ce stage avait pour principal objectif de me permettre de mettre en pratique les connaissances théoriques acquises au cours de ma formation, tout en découvrant le fonctionnement d'un environnement professionnel structuré. Durant cette période, j'ai été affecté à un projet concret consistant à développer un formulaire de collecte d'avis destiné aux utilisateurs finaux des différentes plateformes digitales de la compagnie.

Ce formulaire avait pour but de centraliser les retours clients afin de permettre à l'entreprise d'améliorer l'expérience utilisateur, de corriger les éventuelles lacunes et de consolider la satisfaction de sa clientèle. Mon rôle a été de prendre en charge l'ensemble du cycle de développement du projet : de la conception à la mise en œuvre technique, en passant par l'intégration et les tests.

Ainsi, ce rapport retrace les étapes marquantes de mon stage, les outils utilisés, les difficultés rencontrées ainsi que les compétences que j'ai pu développer au fil de cette expérience enrichissante.

Besoins du client

Durant mon stage, l'entreprise a exprimé deux besoins principaux :

1. Mettre en place une plateforme simple permettant de collecter efficacement les retours des clients sur leurs applications.
2. Développer un système de monitoring capable d'analyser en temps réel les appels AJAX sur leurs plateformes en production, afin de suivre le temps de réponse et détecter les anomalies.

Objectifs du stage

Le stage avait pour principaux objectifs :

1. **Apprentissage et immersion**
    - Découvrir l'organisation interne d'une équipe de développement.
    - Apprendre à travailler dans un cadre collaboratif avec des outils modernes (ex. Git, BitBucket).
2. **Mise en pratique des compétences techniques**
    - Concevoir et développer une application web complète, incluant le **front-end** et le **back-end**.
    - Utiliser des technologies actuelles telles que **Laravel**, **MySQL** et **PHPMyAdmin**.
3. **Contribution au projet de l'entreprise**
    - Développer un formulaire permettant de **recueillir efficacement les avis clients**.
    - Assurer la **centralisation et la sauvegarde des données** pour exploitation future.

Problématique

Comment mettre en place rapidement, avec des technologies modernes, deux solutions répondant aux besoins de l'entreprise :

- un outil de feedback intuitif et fiable,
- un système de monitoring performant et sécurisé, intégré dans un environnement de production.

Méthodologie et démarche projet

La démarche adoptée a suivi les grandes étapes classiques d'un projet informatique :

1. Analyse des besoins et recherches documentaires.
2. Étude des solutions existantes et choix des technologies.
3. Conception des bases et interfaces (schémas, maquettes).
4. Développement et implémentation.
5. Tests, validation et mise en place.
6. Présentation et documentation des résultats.

Contenu

Partie I : Premier projet - Site web de collecte de feedbacks

1. Contexte et objectifs du projet

Lors de mon stage, il m'a été confié un projet complet, allant de la conception à la réalisation d'une application web interne. L'objectif principal était de développer un formulaire en ligne permettant de collecter les avis des clients concernant les plateformes et services proposés par la compagnie.

Ce formulaire avait pour but de :

- Faciliter la remontée des retours utilisateurs sur les produits et services.
- Centraliser les données collectées afin que l'entreprise puisse analyser les points positifs et négatifs.
- Améliorer la satisfaction client en prenant en compte ces retours dans une démarche continue d'amélioration.

Le projet devait donc répondre à des critères d'ergonomie (simplicité d'utilisation pour les clients), de fiabilité (sécurisation et bonne gestion des données), et de maintenabilité (possibilité pour l'entreprise de faire évoluer le système par la suite).

2. Cahier des charges

Le cahier des charges défini en début de mission comprenait les points suivants :

- Création d'un formulaire web responsive permettant aux clients de soumettre leurs avis.
- Gestion de plusieurs champs d'entrée (nom, email, message, note ou satisfaction, etc.).
- Enregistrement des avis dans une base de données MySQL locale.
- Développement en Laravel (PHP) pour le backend, avec un frontend simple, ergonomique et responsive.
- Utilisation d'un système de contrôle de version (Bitbucket) afin de suivre l'évolution du projet.
- Hébergement local durant toute la durée du stage, avec une mise en pré-production prévue après validation..

3. Outils et technologies utilisés

Pour mener à bien le projet, j'ai utilisé plusieurs outils et technologies :

- IDE : Visual Studio Code et PhpStorm (pour le confort et la rapidité de développement).
- Framework : Laravel (PHP) pour le backend, gestion du routing, et logique métier.
- Base de données : MySQL, administrée via phpMyAdmin pour les tests et la gestion des tables.
- Gestion de version : Bitbucket, pour sauvegarder et suivre les différentes versions du code.
- Front-end : HTML, CSS et Bootstrap pour assurer un rendu responsive et agréable.

4. Phase de Dveloppement

- Phase de conception :
  En raison du temps limité du stage, je n'ai pas pu réaliser un MCD complet avant de commencer. J'ai donc conçu directement la base de données sous MySQL, de manière simple et efficace.

- Phase de développement :
  J'ai travaillé à la fois sur le front-end (interface claire et responsive) et sur le back-end (logique de traitement, sauvegarde des données, validation des entrées).

- Phase de tests :
  Le projet a été testé uniquement en local avant ma présentation finale. Après mon départ, l'entreprise a poursuivi le déploiement et l'a intégré en préproduction.

5. Difficultés rencontrées

Comme pour tout projet, certaines difficultés se sont présentées :

1. **Adaptation au framework Laravel**
   - Au début, il m'a fallu un temps d'adaptation pour comprendre le **routing** et l'architecture MVC propre à Laravel.
2. **Gestion du temps**
   - La durée limitée du stage ne m'a pas permis de réaliser un schéma conceptuel détaillé (MCD). J'ai donc opté pour une approche pragmatique afin d'obtenir un

résultat concret et fonctionnel dans les délais impartis.

3. **Travail en autonomie**
    - Étant le seul stagiaire sur ce projet, j'ai dû assumer à la fois la partie **front-end** et la partie **back-end**. Cela m'a demandé une bonne organisation et un apprentissage rapide.

1. Bilan et apprentissages

Ce stage a été une expérience formatrice à plusieurs niveaux :

- **Sur le plan technique :**
    - **Amélioration de mes compétences en Laravel.**
    - **Meilleure maîtrise des bases de données MySQL.**
    - **Familiarisation avec les outils de versionning (BitBucket).**

- **Sur le plan organisationnel :**
    - **Capacité à gérer un projet complet en autonomie.**
    - **Adaptation aux contraintes de temps et de ressources.**

- **Sur le plan personnel :**
    - **Confiance en mes compétences de développeur.**
    - **Sens de la rigueur et de la responsabilité dans un cadre professionnel.**

Partie II : Deuxième projet – Système de monitoring AJAX

1. Analyse du projet

Dans le contexte actuel, les applications web ne cessent de gagner en complexité. Les utilisateurs attendent une expérience fluide, rapide et sans interruption. Les appels **AJAX** (Asynchronous JavaScript and XML) jouent un rôle central dans cette fluidité, car ils permettent de mettre à jour une partie d'une page sans avoir à la recharger intégralement. Cependant, cette souplesse introduit également de nouveaux défis en termes de suivi et de performance.

Les responsables métiers et techniques ont exprimé un besoin précis : disposer d'un **système de monitoring** capable d'analyser et de rapporter en temps réel les performances des appels AJAX.

Les objectifs étaient triples :

- **Suivre les temps de réponse des appels AJAX** : il est essentiel de savoir combien de temps un service met pour répondre à une requête afin d'identifier rapidement tout ralentissement.

- **Détecter les crashs et erreurs** : une requête échouée peut avoir un impact direct sur l'expérience utilisateur. Un suivi centralisé permet de comprendre la cause de ces échecs (erreur serveur, problème réseau, mauvaise configuration côté client).

- **Obtenir des rapports détaillés pour l'optimisation** : au-delà de la détection d'incidents, l'outil devait permettre une analyse fine dans une perspective d'amélioration continue. Ces rapports servent à orienter les choix techniques, prioriser les correctifs et planifier les optimisations futures.

Ce projet s'inscrivait donc dans une démarche stratégique de l'entreprise visant à **garantir la satisfaction client** et à **renforcer la fiabilité des applications**.

---

2. Choix des solutions

Face à ce besoin, plusieurs solutions de monitoring ont été étudiées. Parmi elles, les approches de type **RUM (Real User Monitoring)** ont retenu une attention particulière. Le RUM permet de mesurer les performances réelles vécues par les utilisateurs finaux, contrairement aux tests synthétiques qui simulent des scénarios prédéfinis.

Parmi les différentes solutions disponibles sur le marché, l'entreprise a finalement choisi **Raygun**. Plusieurs critères ont motivé ce choix :

- **Richesse fonctionnelle** : Raygun offre un ensemble complet de fonctionnalités couvrant aussi bien le suivi des performances que la gestion des erreurs. Il ne se limite pas à la collecte brute de données mais propose également des analyses prêtes à l'emploi et des tableaux de bord intuitifs.

- **Compatibilité avec l'existant** : Raygun s'intègre facilement dans une architecture déjà en place, sans nécessiter de refonte profonde. Sa compatibilité avec les technologies déjà utilisées dans l'entreprise (frameworks front-end, back-end et politiques de sécurité) a été un atout décisif.

- **Facilité d'intégration et de déploiement** : le processus d'ajout des snippets JavaScript et la mise en place du suivi ne requièrent pas une expertise technique pointue. Cela a permis aux équipes de démarrer rapidement et de limiter les coûts d'implémentation.

En définitive, Raygun a été considéré comme l'outil le plus adapté, car il répondait à la fois aux besoins opérationnels immédiats et aux ambitions stratégiques de l'entreprise.

---

3. Conception

La conception du système de monitoring s'est articulée autour de plusieurs axes techniques et organisationnels :

- **Ajout de snippets JavaScript dans les pages** : le principe était d'insérer un petit code fourni par Raygun dans les pages de l'application afin d'assurer la remontée des informations de performance et d'erreurs. Ce snippet agit comme un capteur qui envoie les données vers la plateforme Raygun.

- **Gestion de la sécurité via le CSP (Content Security Policy)** : l'entreprise ayant des politiques de sécurité strictes, il était impératif de respecter les directives du CSP. Celui-ci interdit notamment l'usage de scripts inline, ce qui a nécessité une réflexion particulière

pour externaliser le code de suivi tout en maintenant la conformité avec les règles internes.

- **Paramétrage pour le suivi des temps de réponse et erreurs** : le snippet n'est que le point de départ. Une configuration fine a été effectuée pour capturer non seulement les temps de réponse des appels AJAX mais également la nature des erreurs (erreur réseau, HTTP 500, timeout, etc.).

- **Mise en place d'un tableau de bord** : la conception incluait également la création d'un espace visuel de suivi. Ce tableau de bord, configurable et dynamique, permet aux équipes d'avoir une vue consolidée des performances, avec la possibilité de filtrer par application, par type d'erreur ou par période de temps.

Cette phase de conception a constitué une étape cruciale, car elle garantissait que le déploiement à venir respecterait les contraintes techniques, tout en répondant aux attentes fonctionnelles.

---

4. Développement

La phase de développement a consisté à transformer la conception théorique en une solution concrète, utilisable et fiable. Elle s'est déroulée en plusieurs étapes :

1. Installation et configuration de Raygun : les équipes ont commencé par créer un espace de projet dédié dans Raygun, configurant les paramètres de base comme les environnements (développement, pré-production, production) et les clés d'API.

2. Ajout progressif du code de suivi sur les pages de pré-production : plutôt que de tout déployer d'un coup, l'intégration s'est faite de manière progressive. Les premières pages ciblées étaient celles jugées critiques en termes de performance. Cela a permis de limiter les risques et de corriger les problèmes au fur et à mesure.

3. Vérification des logs et correction des erreurs : chaque ajout de snippet était suivi d'une analyse minutieuse des logs générés. Les erreurs rencontrées (mauvaise configuration CSP, absence de permission serveur, erreurs JavaScript inattendues) étaient immédiatement corrigées pour assurer la fiabilité du suivi.

4. Tests sur plusieurs scénarios : une campagne de tests a été menée pour s'assurer de la robustesse du dispositif. Ces tests incluaient :
    - Des appels AJAX multiples déclenchés simultanément afin d'observer le comportement du suivi.
    - Des crashs simulés (interruption volontaire du serveur ou modification des réponses HTTP).
    - Des tests en conditions réelles avec différents navigateurs et réseaux (faible bande passante, latence élevée).

Grâce à cette approche progressive et méthodique, l'outil a pu être intégré de manière fluide dans le système existant.

---

5. Les Résultats

Au terme du projet, l'entreprise s'est dotée d'un **outil interne complet de monitoring AJAX**. Les bénéfices constatés ont été nombreux :

- **Suivi en temps réel** : les équipes peuvent désormais visualiser immédiatement l'état des appels AJAX, détectant instantanément les ralentissements ou les erreurs.

- **Analyses détaillées** : les rapports permettent de comprendre la répartition des temps de réponse, d'identifier les pages ou services les plus lents et d'évaluer l'impact des problèmes sur les utilisateurs.

- **Anticipation des incidents** : grâce aux alertes et à la surveillance continue, il est possible de détecter les anomalies avant qu'elles ne deviennent critiques.

- **Amélioration de la qualité de service** : le monitoring a contribué à renforcer la confiance des utilisateurs en garantissant une meilleure stabilité et réactivité des applications.

---

6. Difficultés et solutions

Comme tout projet informatique, la mise en place du monitoring AJAX a rencontré plusieurs obstacles :

- **Problèmes d'accès serveur** : dans un premier temps, certaines restrictions d'accès empêchaient la collecte des données nécessaires. Après négociation et demande formelle de permissions, ces accès ont été accordés et le projet a pu se poursuivre normalement.

- **Limites du CSP (interdiction des scripts inline)** : cette contrainte sécuritaire interdisait l'utilisation directe du code JavaScript fourni par Raygun. La solution a consisté à externaliser entièrement les scripts et à adapter la configuration CSP afin d'autoriser uniquement les domaines strictement nécessaires au fonctionnement du monitoring.

Ces difficultés, bien que contraignantes, ont permis aux équipes d'acquérir une meilleure compréhension des enjeux de sécurité et de déploiement en environnement sensible.

---

7. Ressenti et Commentaire

Au terme de cette mission, le ressenti des équipes a été particulièrement positif. Ce projet a été perçu comme stratégique car il contribue directement à l'amélioration de la qualité de service et à la satisfaction des utilisateurs.

Les directeurs ont exprimé une **grande satisfaction personnelle et professionnelle lors d'un entretien à distance** d'avoir pu contribuer à la mise en place d'un dispositif reconnu par les responsables comme un atout majeur. Ce projet représente un bel exemple de réussite collective, combinant rigueur technique, respect des contraintes de sécurité et vision orientée vers l'expérience utilisateur.

Conclusion *(1–2 pages)*

Ce stage d'un mois m'a permis d'enrichir mes connaissances et de mettre en pratique mes acquis. J'ai travaillé sur deux projets différents mais complémentaires :

- le développement d'un site web simple mais utile pour collecter les retours clients,
- la mise en place d'un système de monitoring avancé pour analyser les performances des applications.

Ces projets m’ont permis de progresser sur plusieurs plans :

- **technique** : maîtrise de Laravel, Vue.js, TailwindCSS, Raygun, et amélioration de mes compétences en bases de données MySQL et en gestion de serveurs.
- **méthodologique** : compréhension de la démarche projet (analyse, conception, développement, tests, déploiement).
- **personnel** : capacité d’adaptation, autonomie, esprit de recherche, et travail en équipe.

Ce bilan est très positif et constitue une expérience déterminante pour mon futur parcours professionnel.

Annexes *(3–5 pages)*

- Captures d’écran du formulaire développé.
- Schémas UML/MCD de la base de données.
- Exemple de code (snippet d’intégration Raygun).
- Exemple de tableau de bord Raygun.
- Documentation utilisateur fournie à l’entreprise.