



Zone Industrielle Saint Ulfrant, Rue de Saint Ulfrant, 27500 Pont-Audemer, France

RAPPORT DE STAGE

Développeur web

[Informations](#)

Stage du 9 mai au 10 juin 2022

Tuteur : Gregory Dubusse – Manager IT

CARON Tom
BTS SIO - StJoSup

Remerciements

Je voudrais tout d'abord remercier M. Gregory DUBUSSE, manager IT et maitre de stage, qui m'a non seulement laissé découvrir le métier de l'informatique au sein de l'entreprise et m'a laissé de l'autonomie pendant le stage. Je saisis cette occasion pour adresser mes profonds remerciements à l'équipe du secteur informatique avec laquelle j'ai passé tout mon stage. Ils ont su me mettre à l'aise dès le début et ne m'ont jamais laissé de côté. Merci particulièrement à Julian BEUTIN de m'avoir accompagné tout au long de mon stage, même à distance, et m'a appris un grand nombre de choses. Merci à Olivier GAUBOUT pour sa bonne humeur et sa motivation qu'il a trouvées et données pendant le stage. De même pour Cyril SAINT-AUBIN, je le remercie d'avoir trouvé une bonne idée de projet pour le stage. Merci ensuite à Nathalie LAMBERT qui a eu l'amabilité de répondre à mes questions et fournir les explications nécessaires. Merci également à Alexis LELANDAIS, du secteur de maintenance, qui a pris le temps de faire une visite du site de Pont-Audemer et de plus, prenait régulièrement des nouvelles du stage. Merci aussi à Christophe SCHAEFER, des services généraux, qui lui a fait une visite plus extérieur du site. Un grand merci général aux personnels de Thales pour leur accueil, leur sociabilité ainsi que leur amabilité.

Sommaire

Table des matières

Remerciements	1
Sommaire	1
Introduction	2
Présentation rapide	2
Objectifs et enjeux du stage	2
Annonce du plan	2
Développement	2
Présentation entreprise	2
Présentation du stage	5
Missions effectuées	5
Les apports.....	10
Conclusion	11
Annexe	11

Introduction

Présentation rapide

L'identité et la sécurité numériques sont cruciales dans tous les aspects du monde interconnecté dans lequel nous vivons. L'identité et l'authenticité des utilisateurs doivent être vérifiées, pratiquement en temps réel, avec le niveau de sécurité approprié. L'acquisition de Gemalto en 2019, société internationale de renommée mondiale spécialisée dans la sécurité numérique au service des entreprises et des gouvernements dans plus de 180 pays, positionne Thales comme un leader mondial de la sécurité numérique

Objectifs et enjeux du stage

Thales étant un leader mondial au niveau de la sécurité numérique, j'ai souhaité réaliser mon stage dans cette entreprise qui répond à cet enjeu de sécuriser les données du client tout en me formant aux métiers de développeur web « fullstack », ce stage me permet de me convaincre que le développement web m'attire et que je veux, à l'avenir, m'orienter professionnellement dans cette voie.

Annonce du plan

Nous verrons ainsi dans ce rapport comment Thales reste leader dans tous leurs marchés grâce à sa détermination de transformation numérique et digitale. Dans un premier temps, je décrirai l'entreprise avec une présentation plus approfondie. Puis, je présenterai les missions qu'on a pu me donner et que j'ai pu effectuer.

Développement

Présentation entreprise

Thales DIS de Pont-Audemer est un établissement fabricant principalement des cartes électroniques assemblées.



Figure 1: Les différents marchés de Thales

Chiffres clés	
Employés 81 000	Présence mondiale 70 pays
R&D autofinancée 1 milliard €	Chiffre d'affaire 16,2 milliards €



Plus de **3,000** institutions financières s'appuient sur Thales pour protéger leurs transactions et services bancaires.

Plus de **200** programmes gouvernementaux déployés pour l'identité régalienn, la biométrie et les forces de l'ordre.

Mille milliards de transactions bancaires sécurisées par jour sur nos plateformes de cryptage de données.

Plus de **100** déploiements eSIM dans le monde avec des opérateurs mobiles et des OEMs.

Un contrat de plus de 11 ans pour fournir le nouveau **ePasseport du Royaume-uni**.

Nous protégeons chaque jour les identités et les données pour des **milliards** de personnes et d'objets.

Figure 2: informations sur Thales

Histoire de Thales Pont-Audemer

- 1982 : construction de l'usine
- 1983 : Production CSU stoppée après la 2^{ème} crise pétrolière.
Pont-Audemer devient sous-traitant de carte électroniques
- 1984 : Démarrage de la fabrication des publiphones et terminaux bancaires
- 1992 : L'usine accueille la fabrication de carte à puce
- 1996 : La production des terminaux bancaires est transférée à St-Etienne
- 1999 : La production des publiphones est transférée à Besançon
- 2015 : Transfert de la production de CMS de Gémenos vers Pont-Audemer

Site de Thales Pont-Audemer

- 306 employés
- Production sur 7 jours ouvrés
- Production de Modules, cartes et SMD
- 77 000m² dont 10 000m² de bâtiment et 8000m² de production
- Acquisition de plusieurs certifications

Organigramme de l'usine de Pont-Audemer

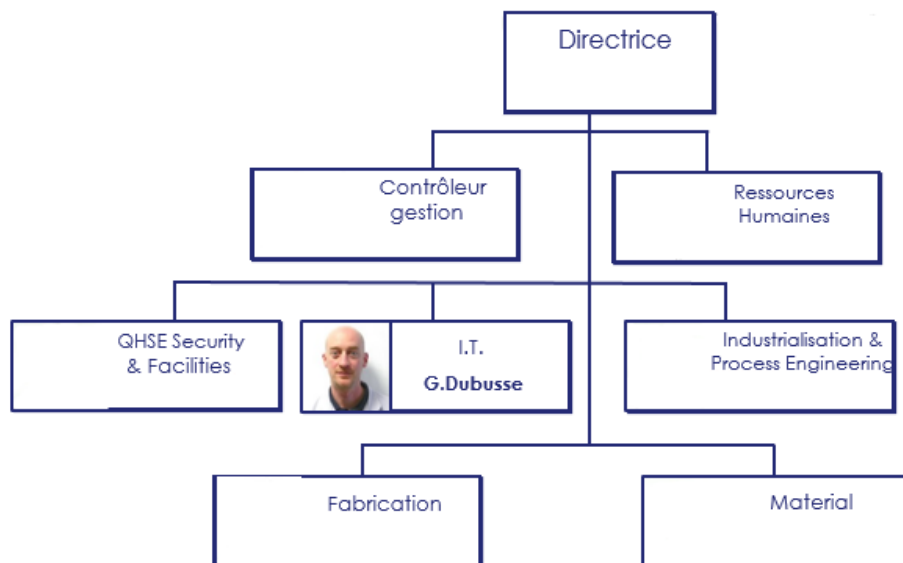


Figure 3: Organigramme Thales Pont-Audemer

Clientèle	
Interne	externe
Usines d'encartage Thales Gestion centralisée de stocks entre les usines	Opérateurs de téléphonie mobile Equipementiers automobiles Marchés publiques : péages

Présentation du stage

Le stage a été principalement dédié au développement web du côté back-end comme du côté front-end, apprendre les bases et prérequis pour créer des API, échanger les données entre API et base de données. Comprendre comment fonctionne le secteur informatique du site Thales.

Missions effectuées

L'une des premières missions données a été **de lancer un serveur sur express.js**, l'objectif de cet exercice est de pouvoir ensuite discuter avec en envoyant des requêtes. Pour cela j'ai utilisé node.js ainsi que la librairie 'express'. Cette mission a été réussie malgré que ce fût mes premiers pas sur le Framework express ainsi qu'avec le langage JavaScript.

J'ai eu comme autres consignes de devoir me **connecter avec l'API au bases de données MySQL et MondoDB** :

- **MySQL** → le but de se connecter à la base de données MySQL est de pouvoir discuter entre MySQL et Express.js. Pour faire cela j'ai dû utiliser la librairie 'Sequelize', le nom de la base de donnée, l'identifiant ainsi que le mot de passe. Les difficultés au début étaient de réussir à bien trier les fichiers et ne pas tout mettre dans un seul et même fichier.
- **MongoDB** → le but de se connecter à la base de données MongoDB est similaire à celui de MySQL, l'idée reste de pouvoir discuter entre Express.js et la base de donnée MongoDB. Pour cela j'ai dû installer une autre librairie qui est celle de 'Mongoose', copier le lien de la base de donnée jusqu'à ma collection. La difficulté ici est que même si c'est le même exercice de

devoir se connecter à une base de donnée, la façon de le faire est différente. Il faut donc éviter de confondre les différentes manières de le faire.

Après avoir appris le design pattern du **Modèle Vue Controller (MVC)**, j'ai dû l'appliquer à partir de ce moment aux différents projets d'API que je faisais. L'idée de faire ce modèle est de bien séparer la vue (ce que l'utilisateur voit), les contrôleurs (ce que le développeur voit) et les modèles. C'est une méthode qui permet aux développeurs de s'y retrouver plus rapidement parmi tous ses fichiers. Pour faire ce modèle j'ai donc dû créer plusieurs fichiers pour chaque modèle, chaque vue et chaque contrôleur. La difficulté a été de devoir être bien organisé dans les fichiers, savoir où aller quand on cherche quelques choses.

J'ai commencé à **utiliser des routes en GET et POST** pour effectuer des requêtes sur une table 'user' que j'avais créé dans la base de données MySQL. Le but de la route en GET était de pouvoir afficher sur une page un user dans la base de données et celui de la route en POST était de pouvoir ajouter un nouvel user dans la table. Cela permet de commencer à apprendre les requêtes de bases appelées 'CRUD'. Pour une meilleure organisation, on mettait toutes les routes dans un dossier 'route'. La difficulté au début est de savoir quelle méthode utiliser (GET ou POST) pour quelle requête.

Dans la même lancée que de devoir créer un user, j'ai dû **hacher le mot de passe** de l'utilisateur qui s'inscrivait. Le but est que le mot de passe ne puisse pas être vu ou décrypté quand on le récupère dans la base de données. J'ai utilisé plusieurs bibliothèques comme 'bcryptjs', 'md5' ou encore 'crypto' pour pouvoir hacher le mot de passe. Il existe plusieurs façons de hacher un mot de passe et aussi plusieurs algorithmes de hash, pour ma part j'ai utilisé l'algorithme 'SHA512' pour hacher le mot de passe.

Encore dans la foulée de créer un 'user' en base de données, on m'a appris à **se prémunir contre les failles XSS** qu'il peut y avoir dans les champs de formulaires lors de la création de compte etc. Le but de cette manipulation est qu'il faut éviter que le code qu'il y a derrière le formulaire interprète les morceaux de code envoyé par des utilisateurs malveillants dans les champs. Pour ce faire j'ai donc utilisé, des fonctions d'échappement ainsi que des filtres comme *isEmail()* pour obliger l'utilisateur à mettre un '@' ou encore *isString()* qui va mettre tous les caractères en string et donc ne va pas interpréter du HTML par exemple. Un développeur se doit de faire attention à cela pour éviter un hack par les champs.

Implémenter des middlewares dans les dossiers routes a été une autre mission que j'ai pu effectuer, cela permet de mieux contrôler les routes et fonctions en exécutant étape par étape puis en passant

à l'étape suivante grâce à la fonction `Next()`. Malheureusement les routes n'étant pas mon fort je n'avais pas réussi à faire fonctionner un middleware puis à passer à l'étape suivante ensuite.

Pour faire apparaître des données comme un 'user' il fallait bien une page (la vue). Pour la vue j'ai donc dû utiliser un **nouveau langage de programmation pour le front 'Pug' ainsi qu'un framework** dédié au front qui est 'bootstrap'. Cela permet de bien séparer l'interface graphique et les contrôleurs de l'application. Pug est comme HTML mais en plus simplifié, lisible et bootstrap nous offre la possibilité de faire beaucoup de style. Pour afficher tous les user j'ai donc utilisé bootstrap pour faire un tableau ou mettre les données à l'intérieur.

Pour la sécurisation, **générer un token JWT** était une des tâches primordiales. Le but est de créer un jeton de validité qui contient les informations de connexions pour que deux parties puissent échanger. Le jeton est séparé en 3 parties, une partie header qui contient le type d'algorithme ainsi que le type de token, une partie payload qui contient les informations qui doivent être transmises (en l'occurrence des informations de connexions ici) et une partie signature qui authentifie l'expéditeur grâce à une clé secrète. Pour ce faire, j'ai utilisé une librairie qui s'appelle 'jsonwebtoken'. La mise en pratique a été plus compliquée que l'apprentissage du token JWT.

Pour finir les requêtes de bases du CRUD, **j'ai fait une route en UPDATE puis en DELETE** pour la table 'user'. La route UPDATE servait à pouvoir modifier des données sur un user déjà enregistré, cette route utilise la méthode UPDATE. Puis la route DELETE servait à pouvoir supprimer un user choisit dans la base de données, cette route utilise la méthode DELETE. Cela m'a permis de voir les 4 opérations de base du CRUD (Create, Read, Update et Delete). Il existe encore plusieurs méthodes mais ceux-ci sont les 4 opérations qui ont complété mon stage.

Une autre mission donnée a été de refaire le modèle CRUD avec une table 'user' mais cette fois-ci en utilisant des **fichiers TypeScript (.ts)**. TypeScript est un autre langage de programmation beaucoup plus typé que le JavaScript, il peut donc faire d'autres fonctionnalités que le JavaScript ne fait pas, mais il faut donc aussi mieux annoter.

Ayant dit que je programmais principalement en C#, on m'a demandé de **programmer rapidement le jeu du « juste prix » en C#** pour voir si j'étais familiarisé avec ce type de langage de programmation. J'ai donc, en utilisant Visual Studio 2019, programmer rapidement ce jeu.

Un mini projet m'a été donné, il consistait à utiliser les **4 opérations de bases du CRUD sur une nouvelle table cette fois-ci de capteurs** puis pouvoir les afficher dans un tableau sur une page. Le but était de voir si j'arrivais à réunir et mettre en œuvre ce que j'avais appris auparavant. Pour afficher un ou tous les capteurs j'ai utilisé la méthode GET, pour créer un capteur j'ai utilisé la méthode POST,

pour pouvoir modifier un capteur j'ai utilisé la méthode UPDATE et pour pouvoir supprimer un capteur j'ai utilisé la méthode DELETE. Ensuite pour afficher les données sélectionnées sur une page j'ai utilisé Pug ainsi que le framework bootstrap pour le tableau et les alertes pour notifier quand un ajout, une suppression ou une modification a été réussi. La difficulté a été de devoir re-associer bootstrap à Pug.

J'ai ensuite commencé le projet principal du stage qui est **le Loto Foot**. Cette idée leurs est venue car un ancien loto foot a été créé auparavant, à la précédente coupe du monde pour que les employés de Thales puissent faire des pronostics et que les meilleurs pronostiqueurs gagnent des lots. Pour mieux comprendre le but du projet voici le cahier des charges du projet :

Cahier des charges

Nom du projet : Pronosticos

Entreprise : Thales DIS

Collaborateur : DUBUSSE Grégory, CARON Tom, BEUTIN Julian

But du projet : Créer une application web permettant aux employés du site de Pont-Audemer Thales DIS de renseigner à l'avance des pronostics sur des compétitions sportives.

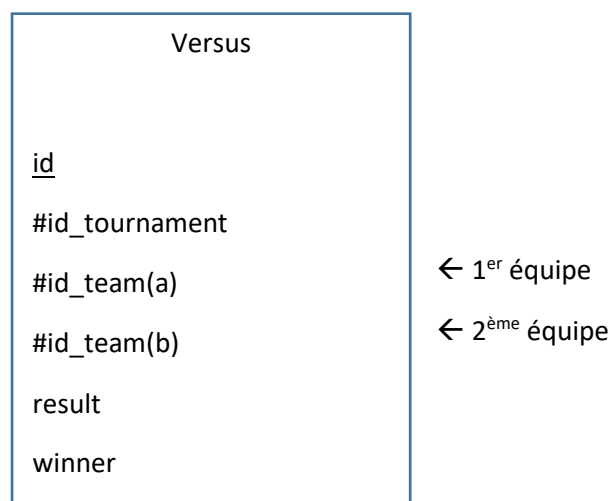
Description technique :

- Créer une application web Full-Stack utilisant la technologie Node.JS
- L'architecture du projet devra respecter un MVC (Modèle-Vue-Contrôleur) classique
- L'application ne devra utiliser que des ORM (Object-Relational-Mapping) pour communiquer avec la base de données (Sequelize)
- La base de données utilisera la technologie MySQL
- Le front-end devra utiliser le Framework « Bootstrap »
- Les utilisateurs pourront et devront s'authentifier via SSO (Single-Sign-On) via le protocole LDAP
- L'application devra être le plus sécurisée possible en utilisant des librairies comme Helmet ou encore Express-Validator
- Les actions des utilisateurs devront être sauvegardé dans une table « logs »
- Un panel administrateur devra permettre de faire des tâches « CRUD » de compétitions, matchs, etc...,
- Le panel administrateur devra aussi être capable de bannir des utilisateurs mais aussi d'afficher les actions des utilisateurs
- Chaque utilisateur devra être capable de renseigner ses pronostics ainsi que les buts
- Une table « ranking » devra permettre de suivre en temps réel les actions des utilisateurs
- Les données des pronostics des utilisateurs devront être stockées sous formes de JSON
- Possibilité de rajouter un captcha
- Le site web devra supporter le HTTPS
- Le front-end devra respecter les couleurs « corporate »

Figure 4: Cahier des charges projet Loto foot

Pour commencer ce nouveau projet j'ai donc commencé à créer une **nouvelle base de données 'loto_foot' + toutes les tables** qui vont avec 'role', 'user', 'logs', 'team', groupe', 'tournoiement', 'player', 'versus', 'pronostic' + leurs clés primaires et étrangères à chacun. Le but après était de voir si toutes les tables étaient bien reliées entre elles par les clés primaire / étrangère. La difficulté ici était de ne pas se tromper en reliant les clés avec 'HasOne' = 'peut avoir qu'une seule' ou 'HasMany' = 'peut avoir plusieurs'.

Pour vérifier que tout était bien relié et que l'on pouvait bien discuter avec toutes les données, je devais **faire apparaître un versus** (là où se trouve le plus de clés étrangères) pour voir si cela fonctionne. Pour ce faire je devais utiliser la méthode GET pour afficher quelque chose. Le problème a été que dans la table versus il y a 2 team, donc 2 id_team en clé étrangère. Sauf que quand les données du versus s'affichées cela affiché les deux même id_team. Voici une représentation schématique pour mieux comprendre :



Ce que cela devrait afficher :

Versus	Tournament	Id team 1	Id team 2	Result	Winner
#1	#1	#1	#2	3-0	#1

Ce qui était affiché :

Versus	Tournament	Id team 1	Id team 2	Result	Winner
#1	#1	#1	#1	3-0	#1

Après avoir fait un le jeu du « **juste prix** » sur **C#**, j'ai dû le faire également sur **node.js**. Malgré que le jeu soit le même et que la façon de le faire soit similaire, j'ai éprouvé des difficultés à discuter avec la console. La boucle que j'ai fait se répéter à l'infini donc la console ne faisait que de demander « proposez un chiffre ».

Pour le projet du loto foot, j'ai commencé par **créer un formulaire d'inscription et de connexion** en appliquant tout ce que j'avais vu auparavant (Méthode POST, sécuriser les failles XSS, ...) avec Pug et bootstrap. Le formulaire devait servir aux employés de Thales qui voulait s'inscrire ou se connecter au loto foot pour pouvoir faire leurs pronostics. Inclure du style.css en plus des fichiers qu'il y avait déjà était plus compliqué pour mélanger Pug/HTML/CSS/Bootstrap.

A côté du projet de loto foot j'ai eu un petit exercice où je devais lancer une **Virtual Machine sous Linux**. J'ai utilisé pour cela VMPlayer avec un iso ubuntu que j'ai trouvé sur internet. L'objectif était de lancer un server SSH et voir également si j'étais capable de lancer une VM tout seul.

Pour plus de sécurité au niveau du loto foot, nous avons décidé qu'un **email de confirmation d'inscription sera** envoyé à celui qui s'inscrit. Le but est de voir comment on envoie un mail de confirmation de création ou changement de mot de passe par exemple. Pour ce faire j'ai utilisé la librairie 'nodemailer' en renseignant l'email avec lequel on enverrai le mail de confirmation. La difficulté a été de rajouter des paramètres que je ne connaissais pas comme le 'tls' ou le 'secure'.

J'ai appris ce qu'était une requête Axios et **effectué une requête Axios** vers une autre API, qui elle gérée déjà l'envoi de mail de confirmation. Le but était de dialoguer avec cette API qui gérée déjà l'envoi de mail pour pouvoir l'utiliser pour le projet du loto foot plutôt que de devoir refaire le code.

Les apports

- **Apports professionnels**

Mes compétences professionnelles se sont améliorées tout au long du stage. La compétence principale est l'apprentissage d'un nouveau langage informatique, le JavaScript avec le Framework qui va avec. Du vocabulaire et lexique que j'ai appris m'ont permis de m'enrichir. La maîtrise de mon anglais s'est amélioré grâce à tous les exercices que j'ai eu à faire. De nouvelles techniques de travail sur les projets en général.

- **Apports personnels**

Grâce à ce stage, j'ai pu développer des compétences personnelles qui m'aideront dans le futur par la suite, comme le travail en équipe, le fait de devoir s'entraider les uns des autres. En plus du travail en équipe, les discussions avec le personnel du site m'ont aidé à développer ma communication avec les autres personnes. Ce stage a développé une grande motivation pour continuer mes études et atteindre un bon niveau pour vouloir, à mon tour, aider autant que l'on m'a aidé.

Conclusion

Ce stage a été très enrichissant car il m'a permis de découvrir le secteur informatique dans une entreprise, les différentes tâches et contraintes de l'entreprise. Grâce à toutes les tâches que l'on m'a confié, j'ai pu développer des compétences personnelles et professionnelles. L'entrée dans cette entreprise confirme mon chemin vers le développement. L'entreprise Thales, qui m'a accueilli pendant ce stage, m'a beaucoup apporté et je suis ravi d'y avoir effectué mon stage. Suite à cette expérience, j'aimerais dans le futur continuer ma formation sur un bachelor numérique en alternance.

Annexe

Figure 1: Les différents marchés de Thales

Figure 2: informations sur Thales

Figure 3: Organigramme Thales Pont-Audemer

Figure 4: Cahier des charges projet Loto foot