

Rapport de stage

CHAVAUDRA Nathan

8 Janvier 2025 au 28 Février 2025

Responsable de stage : Guillaume RAGNEAU

Tutrice de stage : Nathalie LEONARDI

Etablissement / Formation : Epsi Nantes - BTS SIO

Entreprise d'accueil : Exakis Nelite, 34 Pl. Viarme, 44000 Nantes

Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué de près ou de loin à la réussite de ce stage.

En premier lieu, je remercie sincèrement Nathalie Leonardi, ma superviseuse de stage, pour son encadrement, ses conseils précieux et son soutien tout au long de cette expérience. Son expertise et sa disponibilité ont été essentielles pour le bon déroulement de ce stage.

Je souhaite également remercier toute l'équipe CES pour leur accueil chaleureux, leur collaboration et l'ambiance agréable au sein de l'entreprise. Chaque membre de l'équipe a joué un rôle important dans mon intégration et m'a permis de m'épanouir tant sur le plan professionnel que personnel.

Enfin, je tiens à exprimer ma gratitude à toutes les personnes qui ont, de près ou de loin, contribué à la réussite de ce stage et à la réalisation de ce rapport.

Merci à tous.

Nathan Chavaudra

Introduction	4
Contexte et objectifs du stage	4
Présentation de l'entreprise.....	4
Historique	4
Secteurs d'activité	5
Description de la mission	6
Présentation détaillée de la mission	6
Le CES qu'est-ce que c'est ?	7
Objectifs et enjeux pour l'équipe CES	7
Démarche et méthodologie	8
Méthode Agile et Scrum	Erreur ! Signet non défini.
Organisation des sprints.....	Erreur ! Signet non défini.
Utilisation des outils.....	Erreur ! Signet non défini.
Réalisation de la mission	10
Processus de développement	Erreur ! Signet non défini.
Tâches réalisées.....	Erreur ! Signet non défini.
Documentation	8
Structure de la documentation sur confluence	Erreur ! Signet non défini.
Utilisation de la documentation pour le développement.....	Erreur ! Signet non défini.
Journal de bord	13
Conclusion	13
Analyse des Conditions de Travail	Erreur ! Signet non défini.
Apports de la Mission pour l'Entreprise.....	Erreur ! Signet non défini.
Compétences Professionnelles Acquises par le Stagiaire.....	Erreur ! Signet non défini.
Retours de l'équipe	16
Annexes	17

Introduction

Contexte et objectifs du stage

Mon stage de deuxième année de BTS SIO s'est déroulé au sein de l'équipe CES (Centre d'Expertise et de Services) chez Exakis Nelite Nantes. L'objectif principal de ma mission était de réaliser un projet simulé afin d'obtenir une certification en C# et JavaScript.

Présentation de l'entreprise

Historique

Premier logo



Logo actuel



- **2000** : Naissance d'Exakis.
 - Création de la société Exakis, avec son siège social situé à Bidart, dans les Pyrénées-Atlantiques.
 - Lancement des activités d'Exakis, se positionnant rapidement comme un leader dans l'intégration des solutions Microsoft en France.
- **2018** : Fusion avec Nelite.
 - Exakis fusionne avec Nelite, une société d'ingénierie spécialisée dans l'innovation digitale, pour créer un leader du marché dans le secteur de la transformation numérique.
 - Cette fusion a donné naissance à Exakis Nelite, consolidant ainsi sa position de champion français des solutions Microsoft.
- **2023** : Engagé pour un futur durable.
 - Exakis Nelite intensifie ses efforts pour intégrer une approche durable dans sa stratégie, en mettant en œuvre des pratiques responsables et en soutenant des initiatives écologiques.
 - L'entreprise se positionne comme un partenaire de confiance pour les entreprises en quête de transformation numérique éthique et durable.

Secteurs d'activité

Exakis Nelite est spécialisée dans le **conseil en systèmes et logiciels informatiques**, avec un focus particulier sur les technologies Microsoft :

- Modern Workplace
 - Transformation numérique des environnements de travail.
 - Solutions Microsoft 365 (Teams, SharePoint, OneDrive).
 - Collaboration et communication unifiée.
 - Accompagnement à l'adoption des nouvelles technologies par les utilisateurs.
- Cloud et Infrastructure
 - Migration vers Microsoft Azure.
 - Gestion et optimisation des infrastructures cloud et hybrides.
 - Sécurité et gouvernance des environnements cloud.
 - Modernisation des datacenters.
- Cybersécurité
 - Solutions de sécurité Microsoft (Defender, Sentinel, etc.).
 - Gouvernance des identités et des accès (IAM).
 - Protection des données et conformité réglementaire (RGPD).
 - Gestion des menaces et réponse aux incidents.
- Applications et Data
 - Développement d'applications métiers sur mesure.
 - Power Platform (Power BI, Power Apps, Power Automate).
 - Intelligence artificielle et services cognitifs Microsoft.
 - Stratégies Data & Analytics pour valorisation des données.
- Conseil et Accompagnement IT
 - Audit et stratégie IT sur mesure.
 - Change management et adoption digitale.
 - Accompagnement dans la gouvernance des systèmes d'information.
 - Services managés et support continu.

Description de la mission

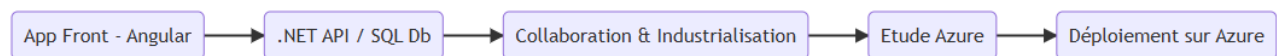
Branche d'activité : CES (Centre d'Expertise et de Services)

Durée de la mission

La mission s'est déroulée sur une période de 7 semaines.

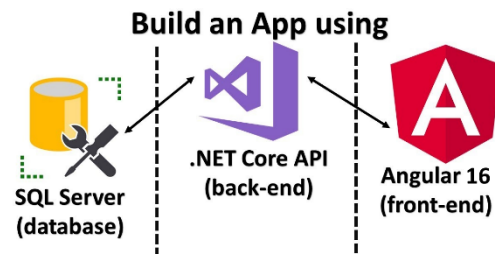
Présentation détaillée de la mission

Lors de mon stage chez Exakis Nelite, ma mission s'est déroulée en plusieurs étapes, allant du développement de l'application à son déploiement sur le [cloud Azure](#) comme ceci :



J'ai commencé par concevoir le Front-end de l'application en utilisant Angular, afin de mettre en place une interface utilisateur permettant la gestion des élèves via un CRUD (Create - Read - Update - Delete). Pour cette première phase, aucune API n'était utilisée, et j'ai donc opté pour un Mock, c'est-à-dire des objets simulés reproduisant le comportement des données réelles de manière contrôlée.

Une fois le [Front-end](#) validé, j'ai conçu la base de données SQL, puis développé une API RESTful en .NET (C#) pour relier l'interface utilisateur aux données réelles. L'API respectait les principes de l'architecture REST afin d'assurer une communication fluide entre le Front-end et la BDD.



Après le développement des différentes couches de l'application, j'ai travaillé sur l'industrialisation du projet, en mettant en place des bonnes pratiques de collaboration, telles que l'utilisation d'un repository Git pour partager et documenter mon travail. J'ai également rédigé de la documentation technique pour faciliter la compréhension du projet par d'autres développeurs.

Une fois le projet prêt, j'ai mené une étude sur les solutions de déploiement dans le cloud **Microsoft Azure**, en analysant les services adaptés (Azure App Service, SQL Azure, etc.) ainsi que les aspects liés à la scalabilité et aux coûts.

Le CES qu'est-ce que c'est ?

Le Centre d'Expertise et de Services (CES) d'Exakis Nelite est une entité dédiée à l'accompagnement des entreprises dans leur transformation numérique, en mettant l'accent sur l'intégration des solutions Microsoft.

L'équipe est composée de développeurs aux compétences variées, chacun apportant son expertise dans des domaines spécifiques. Cette diversité de profils permet d'aborder les projets sous différents angles et d'assurer une couverture complète des besoins techniques

Objectifs et enjeux pour l'équipe CES

- **Objectifs :**
 - Offrir des solutions techniques performantes et adaptées.
 - Optimiser la gestion des projets en garantissant une approche agile et industrialisée.
 - Développer des solutions alignées sur les attentes et contraintes métiers des clients.
- **Enjeux :**
 - Réussir à livrer des solutions dans des délais serrés tout en optimisant les ressources disponibles.
 - S'assurer que les solutions développées respectent les normes de sécurité et les réglementations en vigueur

Réalisation de la mission

Processus de développement

Le développement de l'application s'est déroulé en trois grandes étapes : d'abord la conception du **Frontend**, des **APIs** et de la **base de données**, puis la mise en place de **tests unitaires et d'intégration** pour assurer la stabilité du code, et enfin l'automatisation du **déploiement sur Azure** via **Azure DevOps**.

Tâches réalisées

Le projet comporte deux interfaces distinctes. Le **Frontend Élève**, développé en **Angular**, permet uniquement la recherche et l'affichage des élèves.

De son côté, le **Frontend Professeur**, également développé avec **Angular**, offre un **CRUD complet**, permettant d'ajouter, modifier, supprimer et rechercher des élèves.

Pour gérer la communication entre le frontend et la base de données, deux APIs en **.NET Core** ont été mises en place. L'**API REST classique avec contrôleurs** est utilisée par le **Frontend Élève** et permet uniquement la recherche et l'affichage des élèves.

En revanche, le **Frontend Professeur** repose sur une **API Minimal avec Endpoints**, offrant un **CRUD complet** avec une structure plus légère et performante.

La base de données a été conçue en **SQL Server**, avec une gestion optimisée des relations entre les entités.

Entity Framework Core a été utilisé pour faciliter les interactions avec la base, tandis que des requêtes optimisées ont été mises en place pour améliorer les performances.

Une attention particulière a été portée aux tests, avec l'implémentation de **tests unitaires** sur les services et les API grâce à **xUnit et Moq**. Des **tests d'intégration** ont également été réalisés pour valider la communication entre l'API et la base de données.

L'ensemble du projet a été industrialisé avec **Azure DevOps**. Un pipeline **CI/CD** a été mis en place pour automatiser le **build, les tests et le déploiement**.

L'application, composée des APIs et des Frontend, est hébergée sur **Azure App Service**. Toute la documentation du projet, y compris l'architecture, les bonnes pratiques et les instructions de déploiement, a été centralisée via les **Wikis d'Azure DevOps** pour faciliter la collaboration et la maintenance.

Démarche et méthodologie

Nouveau client

L'intégration d'un nouveau client au sein de l'entreprise suit un processus structuré qui permet d'assurer une prise en charge efficace et une exécution optimale du projet.

Dès qu'un client exprime un besoin, comme par exemple la création d'un site web, il entre en contact avec l'équipe commerciale. Ces derniers jouent un rôle d'intermédiaire essentiel en collectant toutes les informations nécessaires pour bien comprendre la demande et en transmettant ces éléments à l'équipe en charge de l'avant-vente.

Partie Avant-Vente

L'avant-vente intervient alors pour analyser la demande et la chiffrer de manière détaillée. Cette phase est essentielle, car elle permet d'établir une estimation des efforts nécessaires pour répondre aux attentes du client.

À cet effet, l'équipe utilise un document appelé TSP (sous format Excel), qui liste précisément les différentes actions à réaliser ainsi que les ressources nécessaires à la mise en œuvre du projet. L'objectif de cette étape est de fournir une évaluation précise du coût et de la faisabilité du projet, tout en garantissant que toutes les parties prenantes aient une vision claire des implications techniques et fonctionnelles.

Présentation Kick-Off

Une fois cette estimation réalisée, les commerciaux, accompagnés du responsable SL, prennent le relais pour présenter cette proposition au client.

Cette présentation prend la forme d'un "Kick-Off" commercial, au cours duquel les différentes étapes du projet sont expliquées en détail. Cette réunion permet non seulement de valider la compréhension du besoin initial ainsi que de répondre aux questions éventuelles, mais également d'obtenir un accord de principe avant d'entamer les travaux.

Validation de la part du Client

Après cette validation, le projet passe entre les mains de consultants et de responsables SL, qui prennent en charge la gestion opérationnelle de la mission.

Une nouvelle présentation est alors réalisée auprès du client pour détailler les aspects techniques et fonctionnels, confirmant ainsi les objectifs fixés et les livrables attendus.

Mise en Production

Une fois cette dernière validation obtenue, l'équipe démarre la phase de réalisation. Celle-ci comprend plusieurs volets, notamment la conception UX, le développement technique et la mise en place des supports nécessaires pour assurer la pérennité et la maintenance du projet.

Rétrospective

Enfin, une fois le projet finalisé, la livraison est effectuée auprès du client. Cette dernière phase marque la conclusion du processus, assurant que le produit final correspond aux attentes et aux exigences initiales.

Le projet peut ensuite entrer dans une phase de support et d'accompagnement post-livraison, garantissant ainsi une transition fluide et une satisfaction optimale du client.

Documentation

La documentation est un élément clé dans le développement et la maintenance des projets informatiques. Au sein de l'équipe **CES (Centre d'Expertise et de Service)**, toutes les informations ont été soigneusement centralisées sur **Azure DevOps**, en particulier via les **Wikis**. Ce choix a permis une organisation efficace, une accessibilité rapide des données et une collaboration fluide entre les membres de l'équipe.

Structure de la documentation sur Azure DevOps

Centralisation de l'information

- **Wikis et Espaces dédiés** : L'ensemble de la documentation a été structuré en **pages** et **sections** dédiées, chacune correspondant à un domaine précis du projet (ex. API, Front-End, architecture, bonnes pratiques).
- **Templates et Standards** : Des modèles ont été utilisés pour garantir une uniformité dans la documentation, en incluant des sections standardisées comme la **description**, les **objectifs**, les **spécifications techniques** et les **critères d'acceptation**.

Classification par importance

- **Organisation hiérarchique** : Les informations sont classées par **niveau de criticité** et par **domaine fonctionnel**, facilitant ainsi leur recherche et leur consultation.
- **Utilisation de tags et de références croisées** : Les documents sont interconnectés via des liens internes et des tags pour une navigation intuitive et rapide.

Utilisation de la documentation dans le développement

Gestion des anomalies (Bugs)

1. **Identification du problème** : Lorsqu'un bug est détecté, il est consigné sous forme de **Work Item** dans **Azure DevOps**, avec une description détaillée du problème, les étapes pour le reproduire et son impact sur le projet.
2. **Recherche d'informations** : Avant toute correction, les développeurs consultent la documentation existante sur **Azure DevOps Wiki** pour vérifier si des problèmes similaires ont déjà été traités.
3. **Correction et mise à jour** : Une fois la correction effectuée, la documentation est mise à jour avec les détails de l'anomalie et la solution appliquée, garantissant ainsi une traçabilité et une référence future.

Développement d'évolutions

1. **Définition des besoins** : Les évolutions sont documentées dans **Azure DevOps** sous forme de **User Stories**, avec des critères d'acceptation précis.
2. **Consultation du Wiki** : Les développeurs se réfèrent aux pages existantes pour comprendre l'architecture et identifier les composants à modifier.
3. **Implémentation** : En s'appuyant sur la documentation et les bonnes pratiques, les évolutions sont développées et testées.
4. **Mise à jour du Wiki** : Une fois l'évolution finalisée, les modifications sont détaillées dans **Azure DevOps Wiki**, assurant ainsi la continuité des connaissances.

Création de composants

1. **Initialisation du composant** : Une nouvelle section du **Wiki** est créée, décrivant le composant, ses objectifs et ses spécifications techniques.
2. **Développement** : Les développeurs suivent les guides documentés et respectent les standards définis.
3. **Tests et validation** : Les critères d'acceptation et les tests unitaires sont également documentés pour assurer la qualité du composant.
4. **Publication et mise à jour** : Une fois validé, le composant est intégré au projet et la documentation est finalisée avec des instructions d'intégration et d'utilisation.

Processus documentaire type sur Azure DevOps

1. **Création d'une page Wiki** : Une nouvelle page est créée pour chaque nouvelle tâche, composant ou fonctionnalité.
 2. **Ajout d'informations** : La page est complétée avec les détails techniques, des schémas, des extraits de code et des captures d'écran.
 3. **Revue et validation** : La documentation est relue par les membres de l'équipe pour assurer sa pertinence et son exactitude.
 4. **Publication et accessibilité** : La documentation est mise à disposition de l'ensemble de l'équipe et mise à jour régulièrement pour suivre l'évolution du projet.
-

Journal de bord

1ère semaine

- **Mardi**
 - Présentation de l'open space, des équipes et remise des postes de travail.
 - Installation du poste de travail avec un compte professionnel.
 - Présentation du projet de formation Dev Junior.
- **Mercredi**
 - Initialisation du projet Angular et création du repository GitHub.
 - Mise en place de la structure des pages (Home, Search) et des templates des cards.
 - Recherche sur le langage Angular.
- **Jeudi**
 - Validation de la deuxième étape du projet par le référent technique.
 - Création de la base de données et mise en place de l'API RESTful.
 - Connexion de l'API à la BDD et ajout des fonctions CRUD.
- **Vendredi**
 - Intégration BDD → API → Angular.
 - Résolution des problèmes liés à Angular 17.0.0 (mauvaise liaison des URLs API, mise à jour instable).
 - Recherche sur la mise à jour d'Angular (17.0.0 → 19.0.0) et analyse des problèmes rencontrés.

2ème semaine

- **Lundi**
 - Révision et remise en place du code suite aux mises à jour.
 - Correction des problèmes de redirection vers la page Search.
- **Mardi**
 - Présentation du code avec des commentaires détaillés.
 - Ajout d'alertes de validation (succès/erreur) pour les formulaires.
 - Échange avec la référente technique sur l'architecture du code (API et Angular).
 - Refactorisation de l'architecture et ajout d'un fichier de configuration pour les appels API.
- **Mercredi**
 - Ajout de la colonne « Sexe » dans la BDD pour distinguer les élèves (Fille/Garçon).
 - Intégration d'une photo générique selon le sexe.
 - Ajout de la fonctionnalité permettant de lancer une recherche en appuyant sur « Entrée ».
 - Recherche sur le support AzureDevOps.
- **Jeudi**
 - Création d'un repository sur AzureDevOps et ajout des membres requis.
 - Mise en place de l'architecture des repositories.
 - Rédaction du wiki pour documenter le projet.
- **Vendredi**
 - Recherche approfondie sur Azure et ses concurrents.

3ème semaine

- **Lundi**
 - Documentation sur Minimal API.
- **Mardi**
 - Création du projet Minimal API.
 - Ajout du projet à la solution API.
- **Mercredi & Jeudi**
 - Travail sur Minimal API.
- **Vendredi**
 - Réalisation d'un pattern pour structurer le projet.

4ème semaine

- **Lundi**
 - Correction des conflits à la suite de la mise en production.
 - Rédaction du rapport de stage.
- **Mardi**
 - Suite du rapport de stage.
 - Mise en place de l'extension Mapperly pour générer automatiquement des DTOs dans l'API.
- **Mercredi**
 - Ajout manuel de `AddApplicationRoute` pour isoler l'appel aux services dans la couche business.
- **Jeudi**
 - Mise en place du pattern repository.
- **Vendredi**
 - Participation à un événement interne avec l'équipe de développement.
 - Tentative d'implémentation d'un pattern complexe, jugé trop difficile dans le contexte actuel.

Conclusion

Analyse des conditions de travail

Mon stage au sein de l'équipe **CES (Centre d'Expertise et de Service)** chez **Exakis Nelite Nantes** s'est déroulé dans un cadre structurant et formateur. L'organisation Agile et les rituels comme les **Daily Meetings** ont permis un suivi régulier des avancées et une meilleure gestion des tâches. Travailler sur un projet interne dédié aux stagiaires m'a offert un environnement propice à l'apprentissage, tout en me confrontant aux standards et exigences d'un projet en entreprise. L'accompagnement des développeurs expérimentés et l'accès aux bonnes pratiques de développement ont fortement contribué à ma montée en compétences.

Apports de la mission pour l'entreprise

Ce projet interne avait pour objectif principal de permettre aux stagiaires de se former sur les technologies et outils utilisés dans l'entreprise. À travers ce stage, j'ai pu mettre en application des concepts avancés en **développement Full-Stack** en travaillant sur une architecture complète impliquant **Angular**, **.NET Minimal API** et **SQL Server**. Mes contributions ont permis d'améliorer les fonctionnalités du projet et d'optimiser certaines parties du code, notamment en intégrant des **Patterns de Conception** ainsi que des **Isolation de Couches** et des bonnes pratiques pour assurer une meilleure structuration et maintenabilité.

Compétences professionnelles acquises

Ce stage m'a permis de développer et d'approfondir plusieurs compétences clés :

- **Développement Full-Stack** : Renforcement de mes compétences en **Angular** pour la partie front-end et en **.NET Minimal API** pour le back-end, avec une gestion optimisée des bases de données en **SQL Server**.
- **Architecture logicielle et bonnes pratiques** : Mise en place de **patterns** et utilisation d'outils comme **Mapperly** pour améliorer la gestion et la transformation des données entre les couches applicatives.
- **Méthodologie Agile et gestion de projet** : Travail en **Scrum**, gestion des tâches via **Azure DevOps**, participation aux rituels Agile.
- **Travail en équipe et autonomie** : Collaboration avec d'autres stagiaires et développeurs expérimentés, échanges techniques pour résoudre des problématiques et prise d'initiatives sur le développement de nouvelles fonctionnalités.

Bilan

Ce stage chez **Exakis Nelite Nantes** a été une expérience enrichissante, me permettant d'évoluer dans un **cadre professionnel exigeant tout en bénéficiant d'un projet adapté à l'apprentissage**. J'ai pu monter en compétences sur des technologies clés du développement web et backend, tout en adoptant les méthodologies et outils utilisés en entreprise. Cette expérience m'a conforté dans mon projet professionnel et m'a préparé à intégrer des environnements de travail similaires à l'avenir.

Retours de l'équipe

“Quel est votre poste dans l'équipe ?”

- Consultante confirmée
- Consultant confirmé

"Comment évaluez-vous les compétences du stagiaire en matière de travail en équipe et de communication ? Vous donnerez une note de 0 à 5 et si vous le souhaitez rajouter une explication"

- 5, Très bon travail en équipe et bonne communication, aussi bien entre Nathan et son collègue stagiaire, qu'avec le reste de l'équipe.
- 5

" A quel point le stagiaire a-t-il respecté les méthodes de travail dans la gestion de ses tâches ?"

- 5, Nathan a respecté les méthodes de travail ainsi que les tâches qu'on lui a demandées.
- 5

"Quels aspects du travail du stagiaire (techniques, méthodologiques, relationnels) nécessiteraient selon vous des améliorations ? "

- Je n'en vois pas à ce jour. Excellent travail d'ensemble et très bonne implication.
- Techniques et méthode, mais ce sont des choses qu'on acquiert avec l'expérience.

Annexes

Repository Project

Project-Formation-Exakis-Nelite

About this project

Languages

C# TypeScript HTML

Wiki / Wiki du projet AngularEPS

Introduction

AngularEPS est une application web permettant aux professeurs d'EPS de lister et gérer leurs élèves. Le projet est structuré autour de trois composants principaux :

- Frontend Angular** : Fournit l'interface utilisateur.
- API Backend** : Permet la gestion des données et la communication avec la base de données.
- Base de données (DDD)** : Stocke les informations des élèves.

Fonctionnalités principales

- CRUD élèves** : Ajouter, consulter, modifier et supprimer les profils des élèves.
- Recherche avancée** : Trouver des élèves par critères (nom et prénom).
- Interface responsive** : Conçue pour une utilisation fluide sur différents appareils.

Structure du projet

```
Repos :
|-- Frontend - Projet -Formation/      # Frontend Angular
|-- Backend - Projet -Formation/      # Backend API
|-- BDD - Projet -Formation/          # Script BDD
```

1. Frontend Angular

Le repos Angular contient le code source de l'interface utilisateur. Il inclut :

- Composants** : Gèrent les différentes pages et sections de l'application.

Project stats (Period: Last 7 days)

Repos

0 Pull requests opened

3 Commits by 1 authors

Pipelines

0%

Builds succeeded

Members (5)

Aperçu de l'application Web



RECHERCHER

Ajout d'un élève

Nom

Prénom

Age

Sexe

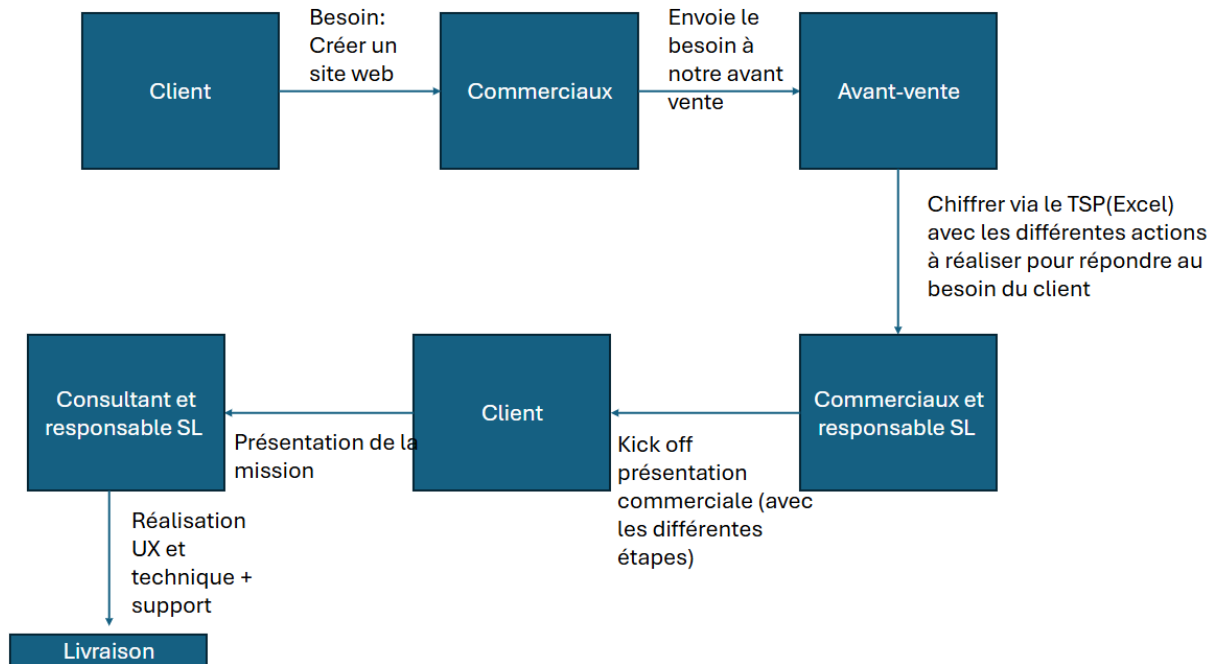
☐ Fille

☐ Garçon

École

AJOUTER UN ÉLÈVE

Schéma



Exemple de Documentation

Introduction

L'objectif de cette étape est de construire une application **Angular** qui permet de gérer une liste avec une approche CRUD (Create - Read - Update - Delete), pour l'instant sans API.



Contexte

C'est la rentrée des classes 🎒 🍎 .. tu es un professeur d'EPS 😊 🙌
Dans ta jeunesse tu as développé quelques applications et tu te dis qu'une petite application Angular pour gérer la liste de tes élèves pourrait t'être utile.

Objectif

L'application doit pouvoir :

1. Charger une liste d'élève (mocké) dès son chargement. Un élève sera représenté par les informations "**prénom nom**".
2. Parce qu'il y aura toujours **UN** élève insupportable dans la classe dès le premier jour, l'application doit offrir la possibilité de virer l'élève (suppression de la liste).
3. Parce qu'il y aura toujours **UN** élève qui arrivera deux semaines en retard, l'application doit offrir la possibilité d'inscrire ce nouvel élève en renseignant son prénom et son nom (ça rajoutera l'élève dans la liste).
4. Une autre page de l'application doit être accessible afin d'offrir une fonctionnalité de filtre des élèves en fonction de leur prénom.
5. Travailler un peu le style de l'application

Critères technique

Vous devez respecter les consignes suivantes :

- NodeJS **LTS** (idéalement voir pour utiliser soit **NVS** soit **NVM**, pour switcher facilement de version de NodeJS)
- Angular **LTS**
- Language type préprocesseur CSS : **SASS** ou **SCSS**
- Vous êtes libre d'utiliser des bibliothèques, notamment pour le visuel (UX/UI).