

DOSSIER PROFESSIONNEL (DP)

Nom de naissance

➤ **PRE**

Nom d'usage

➤ **PRE**

Prénom

➤ **Tchèssi**

Adresse

➤ *20 rue Francis de pressensé 13001 Marseille*

Titre professionnel visé

Titre RNCP - Concepteur Développeur d'applications

MODALITÉ D'ACCÈS :

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

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.

Sommaire

Exemples de pratique professionnelle

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

p. 5

► Maquetter une application de Chat

p. 6

► Développer la partie front-end d'une interface utilisateur web

p. 8

► Développer la partie back-end d'une interface utilisateur web

p. 11

► Développer une interface de type desktop – Python

p. 15

► Développer une interface de type desktop – Panel-admin

p. 18

Concevoir et développer la persistance des données en intégrant les recommandations de sécurité

p.

► Concevoir une base de données

p. 21

► Développer des composants dans le langage d'une base de données

p. 24

► Intégrer les recommandations de sécurité

p. 27

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

p.

► Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement

p. 30

► Développer des composants métiers

p. 32

► Construire une application organisée en couches

p. 34

► Préparer et exécuter le déploiement d'une application

p. 36

► Préparer et exécuter les plans de tests d'une application

p. 39

Intitulé de l'activité-type n° 4

p.

- Intitulé de l'exemple n° 1

p.

- Intitulé de l'exemple n° 2

p.

Intitulé de l'exemple n° 3

p.

Titres, diplômes, CQP, attestations de formation *(facultatif)*

p.

41

Déclaration sur l'honneur

p.

42

Documents illustrant la pratique professionnelle *(facultatif)*

p.

43

Annexes *(Si le RC le prévoit)*

p.

45-55

EXEMPLES DE PRATIQUE PROFESSIONNELLE

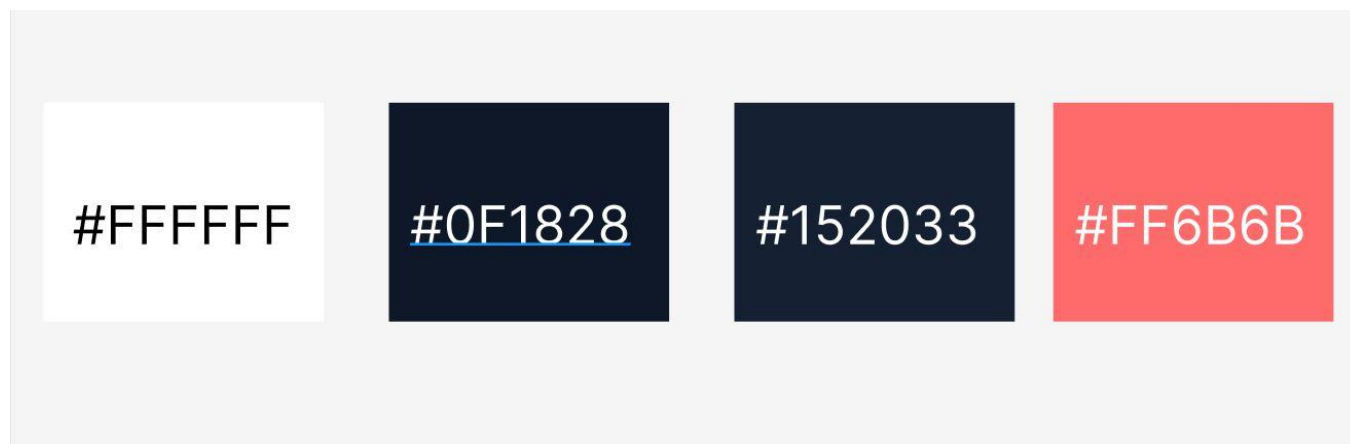
Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

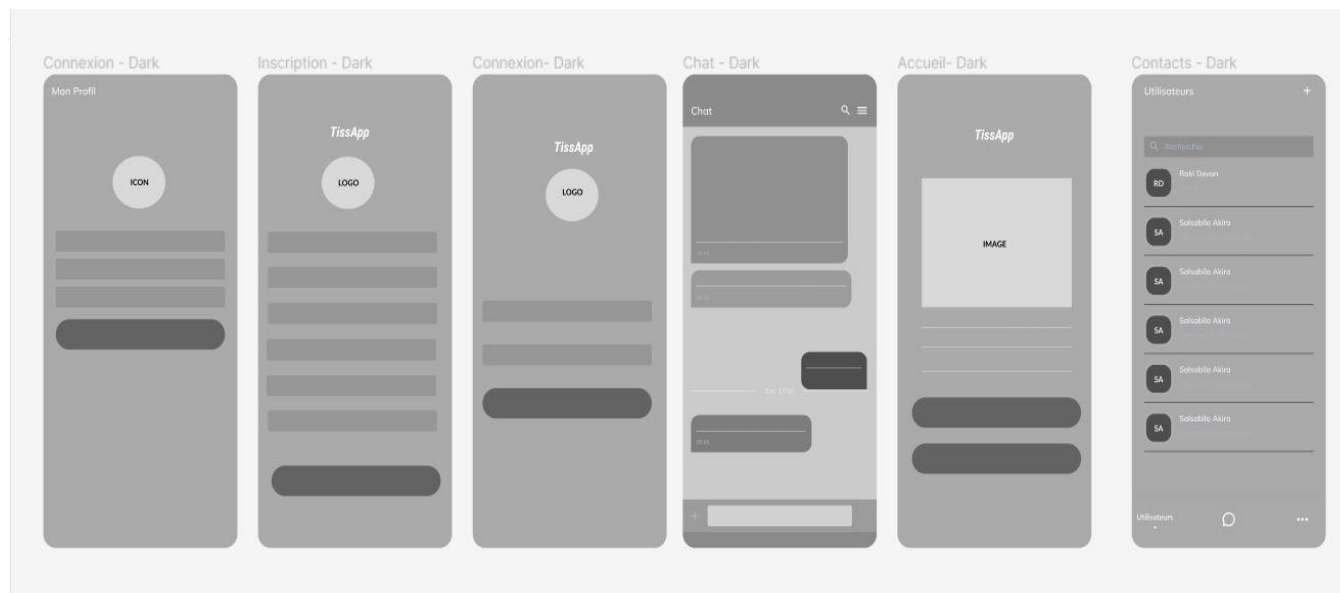
Exemple n°1 - Maquetter une application

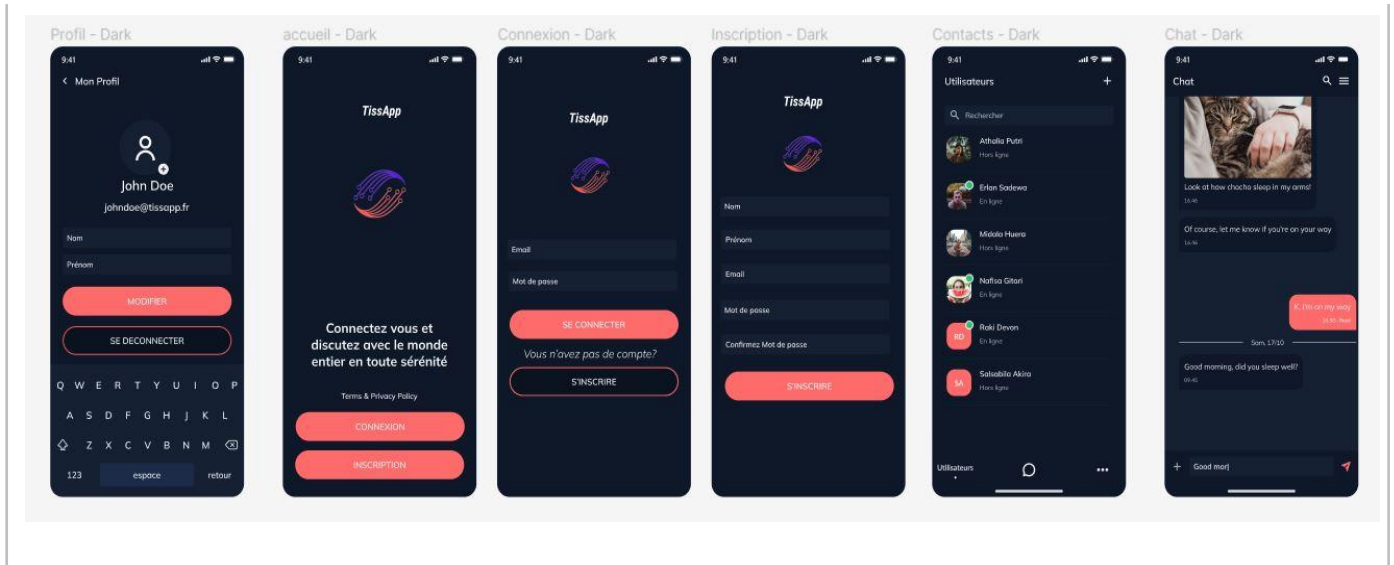
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Charte graphique :



Wireframe:





2. Précisez les moyens utilisés :

Cette maquette a été réalisée avec **figma** et **canvas** pour la création du logo

- Éditeur de texte:
J'ai utilisé Visual Studio Code, qui est un environnement de développement intégré (IDE). Il permet la prise en charge native de divers langages.
- Création des maquettes:
Pour réaliser les maquettes, j'ai utilisé Figma qui est un outil disponible en ligne et canvas pour la création du logo

3. Avec qui avez-vous travaillé ?

Tissapp est une application créée en collaboration avec Tchessi, Ismail, Samir et Salim. Ensemble, nous avons développé une stratégie de conception d'interface pour notre application. Nous avons l'initial de nos prénom pour donner le nom de notre application: **TissApp**

4. Contexte

Nom de l'entreprise, organisme ou association ➤ **La plateforme**

Chantier, atelier, service ➤ **Dans le cadre de ma formation : Concepteur développeur d'application web**

Période d'exercice ➤ Du : **Janvier 2023** au **Mars 2023**

Activité-type 1 *Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité*

Exemple n° 2 ► *Développer la partie front-end d'une interface utilisateur web*

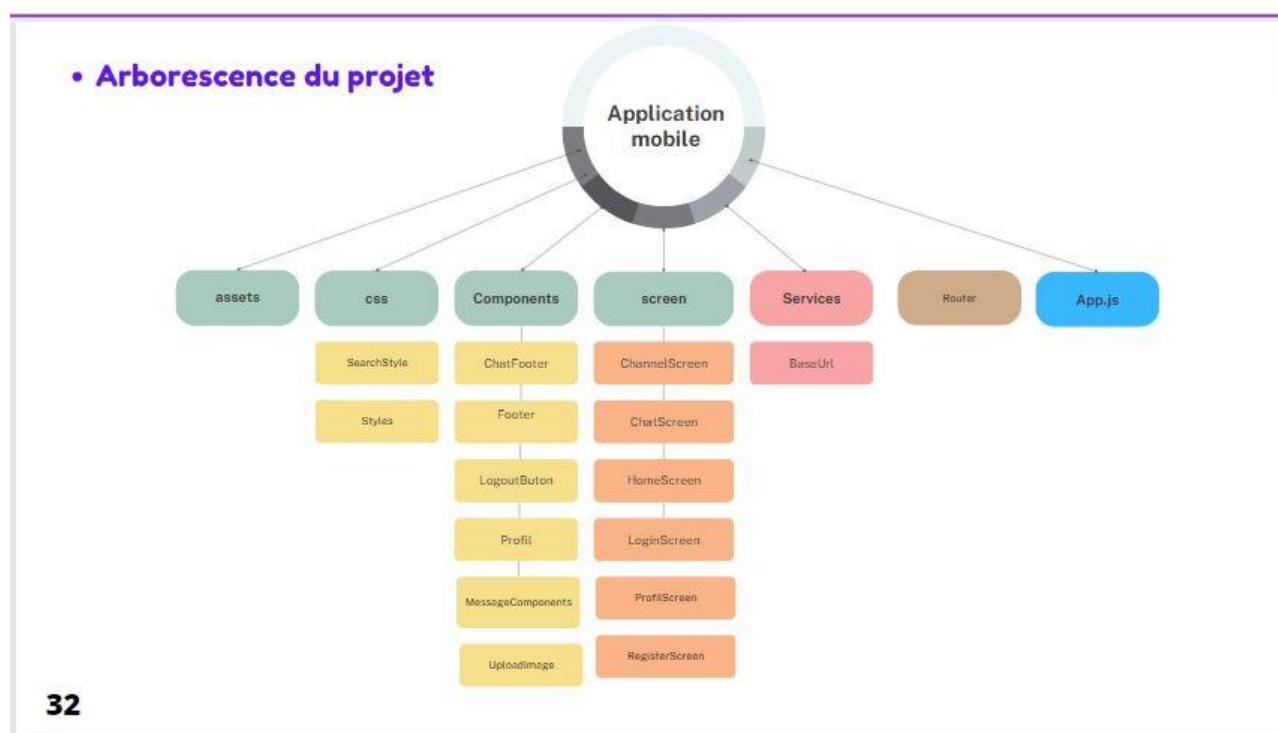
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

En tant que concepteur développeur d'application, les tâches et les opérations que j'ai effectuées varient en fonction du projet sur lequel j'ai travaillé. Cependant, voici quelques exemples de tâches courantes que j'ai réalisées lors du développement de la partie front-end de notre application web :

1. **Analyse des besoins** : J'ai commencé par une analyse détaillée des besoins de l'interface utilisateur. La première étape du développement front-end est de comprendre les besoins de l'utilisateur et les fonctionnalités attendues de l'application. Cela implique la lecture des spécifications, des maquettes et des exigences fonctionnelles, ainsi que des discussions avec l'équipe de développement.
2. **Prototypage** : J'ai utilisé des outils de conception comme canva et Figma pour créer des prototypes d'interface utilisateur détaillés, comprenant des wireframes à basse et haute fidélité ainsi que des maquettes interactives.
3. **Conception de l'interface utilisateur** : À partir des spécifications et des maquettes, je peux concevoir l'interface utilisateur de l'application web. Cela implique la création de wireframes et de maquettes fonctionnelles, ainsi que la définition de l'expérience utilisateur (UX).
4. **Configuration de l'environnement de développement** : La première étape consiste à configurer l'environnement de développement pour React Native. Cela implique l'installation de Expo, du SDK Android ou iOS, et de l'interface de ligne de commande React Native (CLI).
5. **Conception de l'interface utilisateur** : À partir des spécifications de l'application de chat, j'ai conçu l'interface utilisateur de l'application. Cela inclut la création de wireframes et de maquettes fonctionnelles pour la disposition de l'écran, les icônes, les couleurs et la police.
6. **Développement des composants React Native** : Une fois que la conception de l'interface utilisateur est terminée, j'ai commencé à développer les composants React Native pour l'application. Cela inclut des composants pour l'affichage de messages, des formulaires de saisie, des boutons et des icônes.
7. **Tests et débogage** : Tout au long du processus de développement, j'ai testé régulièrement l'application pour s'assurer qu'elle fonctionne correctement et que l'interface utilisateur est cohérente.
8. **Collaboration avec l'équipe de développement** : Enfin, j'ai collaboré avec l'équipe de

développement de 4 développeurs pour assurer la cohérence entre le front-end et le back-end de l'application de chat. Cela a impliqué des petites débriefs et des discussions pour s'assurer que tous les composants de l'application sont alignés sur les objectifs de notre projet. Nous avons utilisé Trello pour gérer des projets collaboratifs, car il permet aux membres de l'équipe de voir rapidement l'état d'avancement des tâches et de communiquer efficacement sur les progrès et les obstacles.

Arborescence du projet:

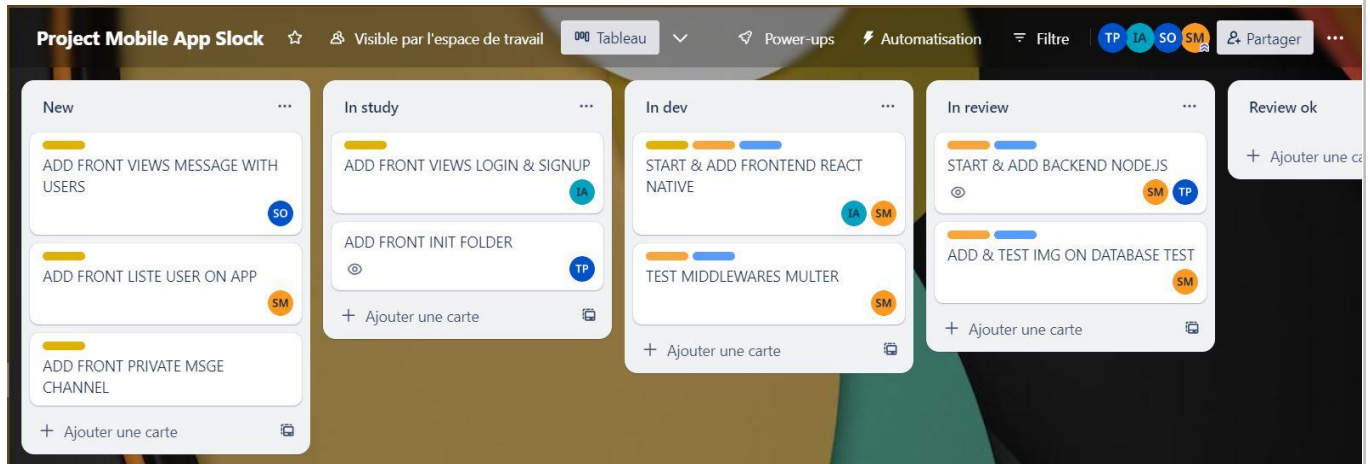


2. Précisez les moyens utilisés :

1. L'éditeur de code **Visual Studio code**
2. **React-native** pour la création de notre application
3. **Expo GO** pour exécuter l'application sur un émulateur et le mobile
4. **Figma** un outil de conception a été utilisé pour créer la maquette d'interface utilisateur et le prototype.
5. **Canvas** pour la création du logo
6. La plateforme de test et de débogage comme Expo a été utilisée pour tester et déboguer l'application.
7. Le système de contrôle de version tels que **Git** a été utilisé pour suivre les modifications apportées au code et pour faciliter la collaboration avec l'équipe.
8. **Jira et Trello**: C'est une application de gestion de projets en ligne qui permet aux utilisateurs de créer des tableaux, des listes et des cartes pour organiser et hiérarchiser les tâches et les projets. Elle utilise une approche visuelle de la gestion de projets, ce qui

facilite la visualisation de l'avancement des tâches et l'identification des éventuels blocages ou retards.

Capture d'écran de notre espace de travail:



3. Avec qui avez-vous travaillé ?

Avec **Samir, Salim** et **Ismail** ont travaillé ensemble sur un projet de développement de notre application mobile. On s'est réparti les tâches : Samir et moi étions chargés du développement de la partie front-end de l'application, Salim s'est concentré sur la création de la maquette et du prototype et Ismail s'est occupé de différents rôle et la création de base de données et **moi** de la gestion de projet et de la coordination entre les différents membres de l'équipe.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **La plateforme.**

Chantier, atelier, service ▶ **Dans le cadre de ma formation : Concepteur développeur d'application web**

Période d'exercice ▶ Du : **Janvier 2023** au : **Mai 2023**

Activité-type 1 *Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité*

Exemple n° 3 - *Développer la partie back-end d'une interface utilisateur web*

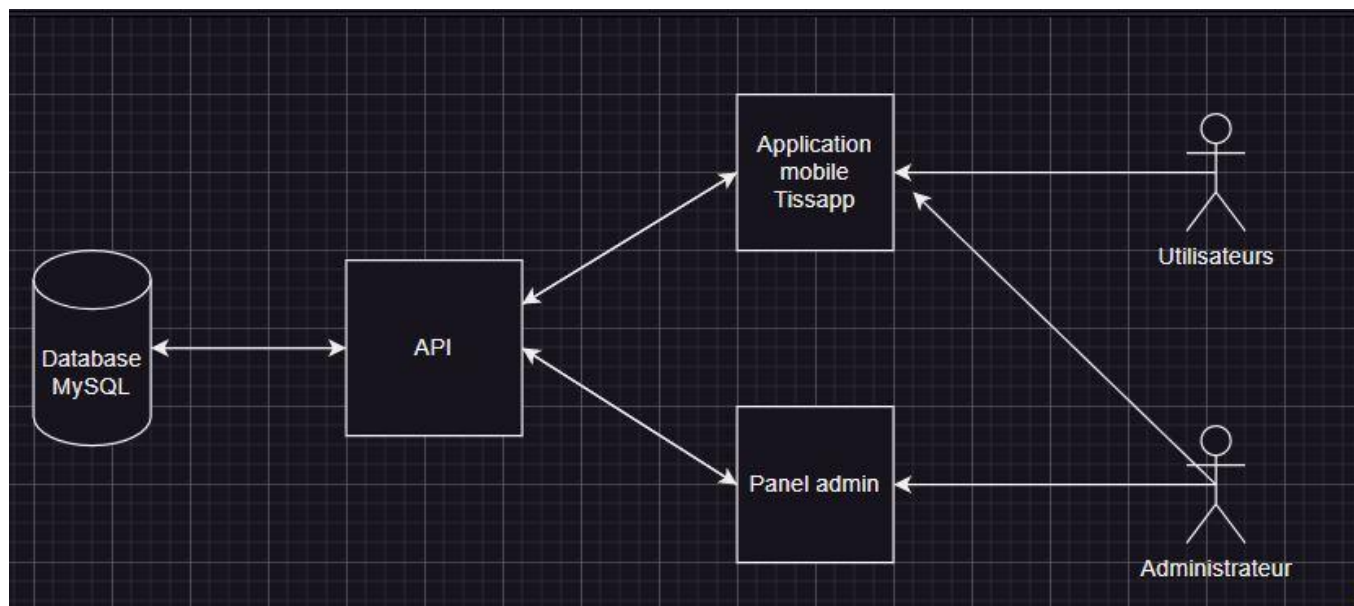
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

Le développement de la partie backend de notre application a impliqué la création et la gestion des serveurs, des bases de données, et des APIs qui permettent à l'application de fonctionner de manière fluide et fiable. Voici quelques tâches et opérations courantes :

1. **Architecture et Conception du Système** : Avant toute écriture de code, j'ai passé beaucoup de temps à concevoir l'architecture du système. Cela a impliqué la création de diagrammes de flux de données et d'architecture pour visualiser et planifier les relations et interactions entre les différents composants du système.
2. **Gestion des bases de données** : J'ai conçu la structure des bases de données nécessaires à notre application. Cela a inclus la création de schémas de base de données, la définition des relations entre les tables, et la mise en place d'indices pour optimiser les performances des requêtes.
3. **Développer des API** : une API (interface de programmation d'application) est une interface qui permet aux applications de communiquer entre elles. Dans le cas d'une interface utilisateur web, une API permet à l'interface utilisateur de communiquer avec le serveur. Les API peuvent être développées en utilisant différents protocoles et technologies, tels que REST. La création de routes et d'API pour permettre aux utilisateurs d'interagir avec l'interface utilisateur, comme l'envoi de données, la mise à jour de données et la récupération de données.
4. **Écrire du code serveur** : le code serveur est responsable de traiter les requêtes provenant de l'interface utilisateur et de retourner les réponses appropriées. Cela implique la mise en place de routes de serveur, la gestion des sessions utilisateur et la gestion des erreurs.
5. **Sécurité** : la sécurité est une considération importante lors du développement de toute application web. Cela implique la mise en place de pratiques de sécurité, telles que l'authentification et l'autorisation, la validation des entrées utilisateur, la protection contre les attaques de sécurité et la gestion des identités. La mise en place d'une gestion d'authentification et d'autorisation pour les utilisateurs de l'application, tels que la création de comptes utilisateur, la gestion de la session et les autorisations d'accès à certaines fonctionnalités de l'interface utilisateur.
6. **Tests et débogage** : Cela inclut l'exécution de tests pour vérifier la fonctionnalité du backend, ainsi que la résolution des problèmes rencontrés lors du développement.
7. **Déploiement** : Cela implique la mise en place de l'infrastructure pour le déploiement de l'application sur les serveurs de production.

8. **Maintenance et amélioration** : Cela implique la surveillance de l'application, la gestion des mises à jour et la résolution des problèmes qui peuvent survenir après le déploiement.

Architecture de l'application:

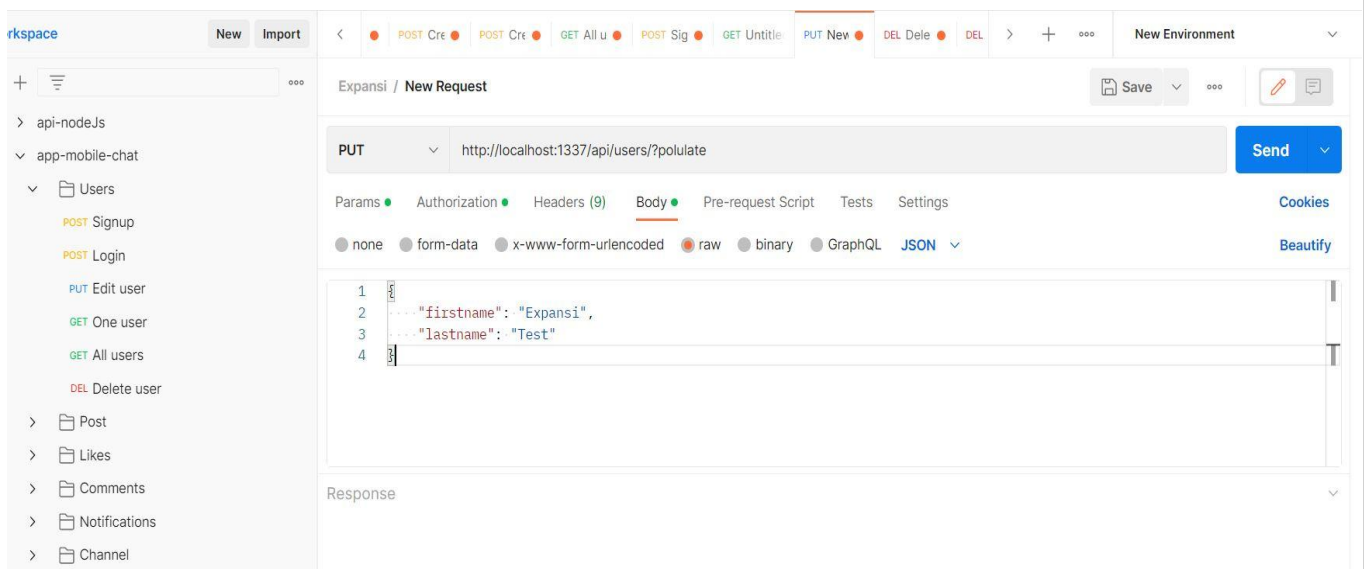


2. Précisez les moyens utilisés :

1. **Node.js** est un environnement d'exécution JavaScript côté serveur, qui permet aux développeurs d'utiliser JavaScript pour développer des applications back-end.
2. **Express.js** : Express est un framework web populaire pour Node.js, qui permet de créer facilement des applications web en utilisant des routes, des middleware et des modèles. Express offre également un ensemble de fonctionnalités pour la gestion des sessions, la gestion des cookies et la manipulation des fichiers.
3. **NPM** : NPM est un gestionnaire de paquets pour Node.js, qui permet de trouver, installer et gérer les dépendances de notre application. J'ai utilisé NPM pour installer des packages, des modules suivant:
4. **JWT (JSON Web Token)** est un standard ouvert pour la création de jetons d'accès sécurisés et autoportants entre deux parties. Les jetons JWT sont souvent utilisés pour l'authentification et l'autorisation dans les applications web et mobiles.
5. **Bcrypt** est une fonction de hachage de mot de passe sécurisée et un algorithme de cryptage utilisé pour stocker et vérifier les mots de passe dans les applications web. Lorsqu'un utilisateur s'inscrit sur un site Web et fournit un mot de passe, le mot de passe est haché à l'aide de Bcrypt avant d'être stocké dans la base de données. Lorsque l'utilisateur se connecte à nouveau, le mot de passe fourni est à nouveau haché et comparé au hachage stocké dans la base de données. Si les deux hachages sont identiques, cela signifie que le mot de passe fourni est correct et que l'utilisateur est authentifié. Cependant, il est important de noter que la sécurité d'un système ne dépend pas seulement de l'utilisation de Bcrypt pour stocker les mots de passe, mais également

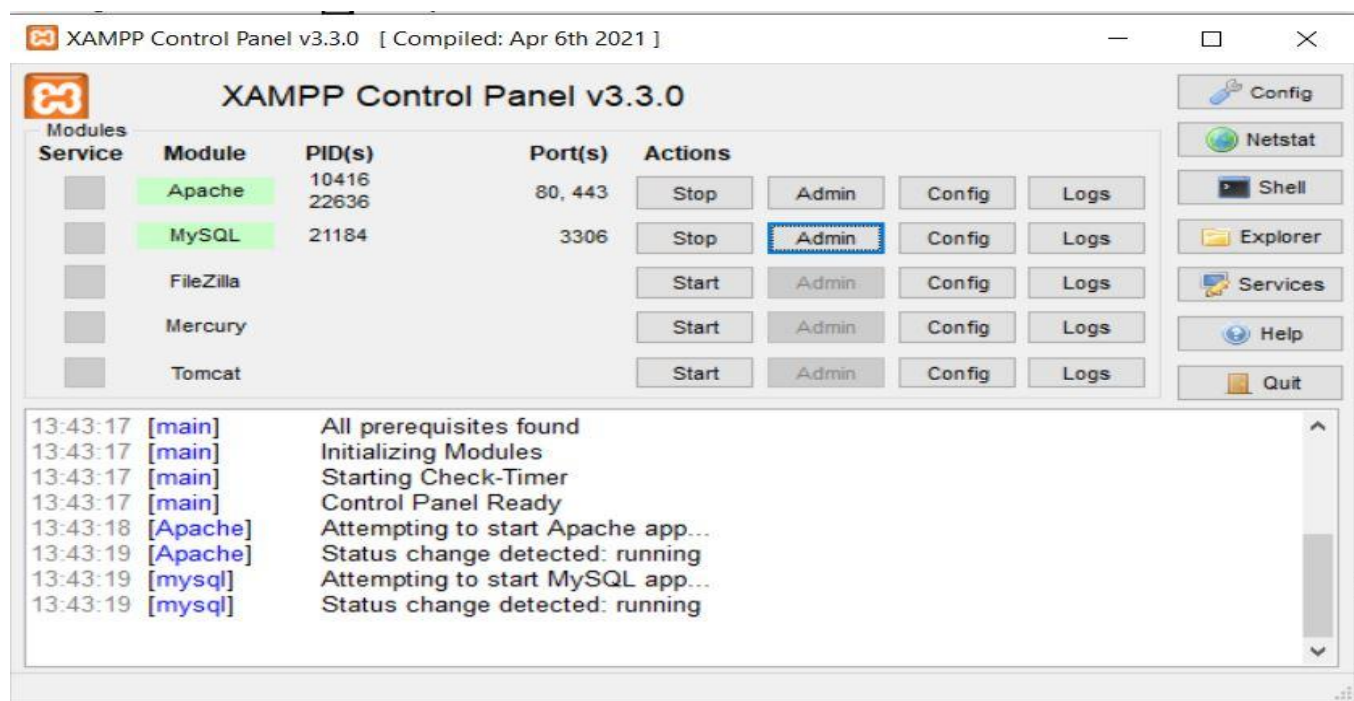
de la mise en œuvre globale de la sécurité dans l'application, y compris l'utilisation de HTTPS, la validation des entrées de l'utilisateur et la protection contre les attaques par injection SQL.

6. **Multer** est une bibliothèque de middleware pour gérer les fichiers multipart/form-data dans les applications web Node.js/Express. Les fichiers multipart/form-data sont généralement utilisés pour envoyer des fichiers binaires, tels que des images ou des vidéos, dans les requêtes HTTP.
7. **Sequelize** est une bibliothèque **ORM (Object-Relational Mapping)** pour Node.js qui permet de manipuler facilement des bases de données relationnelles à l'aide d'un langage de programmation orienté objet. Sequelize est compatible avec plusieurs bases de données relationnelles, y compris MySQL. Avec Sequelize on définit le modèle Sequelize qui correspond aux tables dans la base de données. Sequelize permet également de définir des relations entre les tables, comme des relations de clé étrangère.
8. **Nodemon** est un utilitaire de développement pour Node.js qui permet aux développeurs de surveiller les modifications de fichiers dans leur projet Node.js et de redémarrer automatiquement l'application à chaque fois qu'un changement est détecté.
9. **Postman** est un outil de développement d'API qui permet de tester, de déboguer et de documenter les API. Il fournit une interface graphique utilisateur (GUI) facile à utiliser pour envoyer des requêtes HTTP et HTTPS aux API, ainsi que pour analyser les réponses.



10. **XAMPP** est un environnement de développement complet qui permet de créer et de tester des sites web localement sur un ordinateur, sans avoir besoin d'une connexion Internet ou d'un hébergement en ligne. Il permet aux développeurs web de travailler sur des projets en local avant de les déployer sur un serveur en ligne, ce qui permet de gagner du temps et d'éviter les coûts liés à l'hébergement web pendant la phase de

développement.



3. Avec qui avez-vous travaillé ?

J'ai travaillé en étroite collaboration avec Samir, Salim et Ismail pour comprendre les besoins en matière de données et pour m'assurer que la base de données répondait à ces besoins. J'ai également travaillé dans le respect des réglementations relatives à la protection des données et de la vie privée. Les tâches de gestion de la base de données étaient souvent complexes et nécessitent une attention particulière pour éviter les erreurs pouvant entraîner une perte de données ou des problèmes de performances.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **La plateforme.**

Chantier, atelier, service ▶ **Dans le cadre de ma formation : Concepteur développeur d'application web**

Période d'exercice ▶ Du : **Janvier 2003** au : **Mai 2023**

5. Informations complémentaires (facultatif)

Activité-type 1 *Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité*

Exemple n° 4 - *Développer une interface de type desktop – Python*

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

La conception et le développement d'une interface de type desktop pour un jeu Sokoban que j'ai réalisé en Python impliquent plusieurs étapes. Voici comment j'ai réalisé cette tâche :

1. **Analyse des Exigences** : Tout d'abord, j'ai évalué les besoins de l'interface utilisateur pour le jeu Sokoban. Ceci comprend le design du plateau de jeu, la représentation des murs, des caisses, des zones cibles, et du personnage du joueur.
2. **Conception de l'Interface** : J'ai ensuite esquissé une conception de l'interface utilisateur qui a défini la disposition du plateau de jeu, le compteur de score, le niveau, et d'autres éléments de l'interface tels que les boutons pour contrôler le jeu.
3. **Développement de l'Interface** : À l'aide de **Pygame**, j'ai développé l'interface utilisateur du jeu. Pygame fournit des modules pour le rendu graphique, la gestion des événements d'entrée, et bien d'autres fonctionnalités utiles pour le développement de jeux. J'ai utilisé ces modules pour créer des sprites pour le personnage du joueur, les murs, les caisses et les zones cibles, et j'ai géré les entrées de l'utilisateur pour déplacer le personnage du joueur.
4. **Implémentation de la Logique du Jeu** : Une fois l'interface créée, j'ai implémenté la logique du jeu. Par exemple, déplacer une caisse lorsqu'elle est poussée par le joueur, empêcher le joueur et les caisses de passer à travers les murs, et détecter quand toutes les caisses sont sur les zones cibles pour signaler que le niveau est terminé.
5. **Tests** : Enfin, j'ai effectué des tests pour m'assurer que l'interface et la logique du jeu fonctionnaient correctement. J'ai joué au jeu dans différents scénarios, en testant les différentes fonctionnalités de l'interface et en m'assurant que la logique du jeu réagisse correctement à chaque action.

Ces tâches ont été réalisées dans un environnement de développement professionnel, en utilisant des outils de gestion de version comme Git pour le contrôle du code source.

2. Précisez les moyens utilisés :

Voici les moyens que j'ai utilisés pour développer l'interface de type desktop pour le jeu Sokoban en utilisant Python:

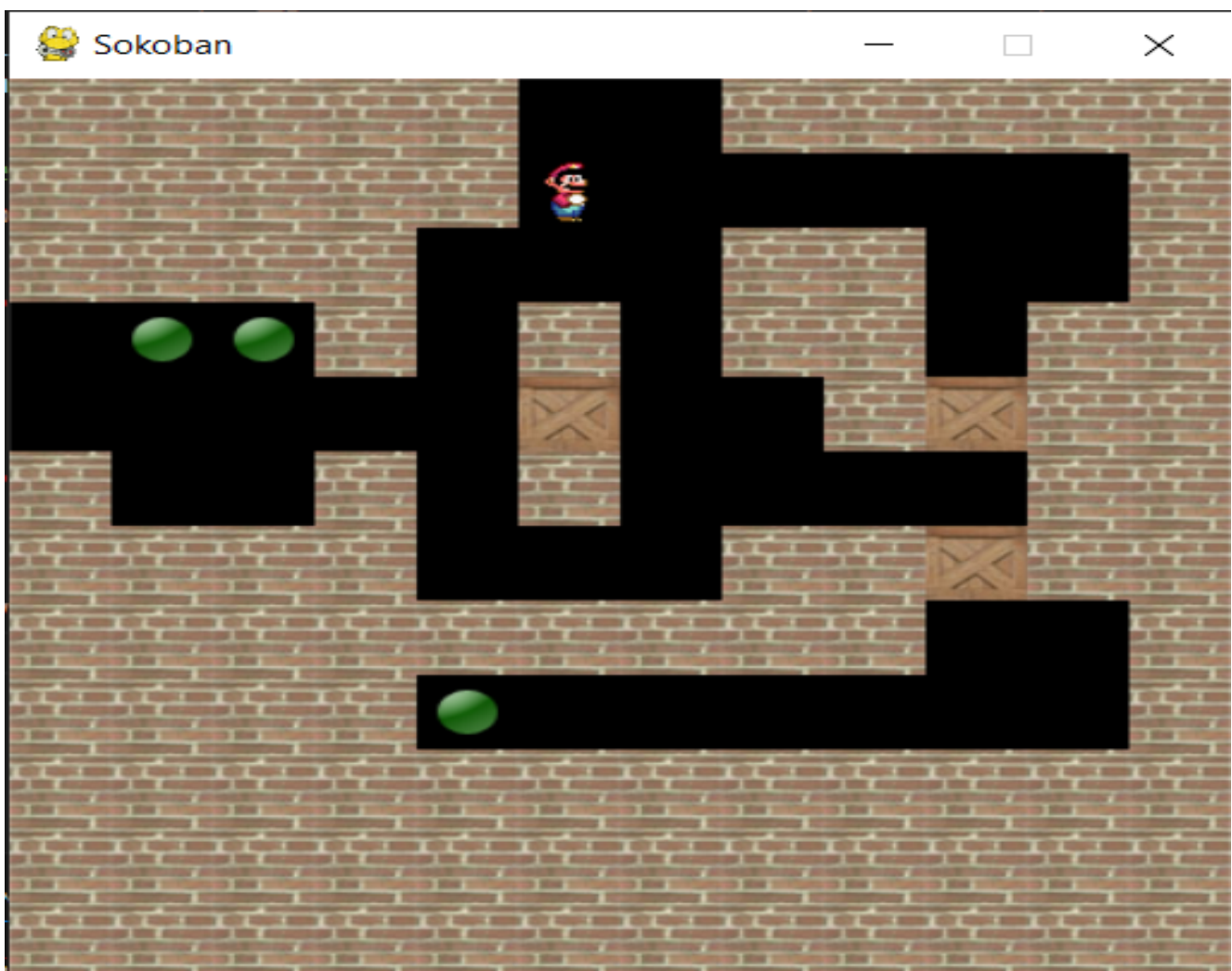
1. **Python** : Python est un langage de programmation puissant et flexible que j'ai utilisé pour la logique du jeu et le développement de l'interface utilisateur.
2. **Pygame** : Pygame est une bibliothèque Python gratuite et open source utilisée pour le développement de jeux vidéo. J'ai utilisé Pygame pour créer et gérer l'interface graphique du jeu, gérer les entrées de l'utilisateur, et effectuer des tâches liées au rendu

graphique.

3. **IDE VsCode** : J'ai utilisé un environnement de développement intégré (IDE) pour Python, tel que VS Code, pour écrire, déboguer et tester mon code.
4. **Git** : Git est un système de contrôle de version que j'ai utilisé pour suivre et gérer les différentes versions de mon code pendant le développement.
5. **Images et ressources graphiques** : J'ai utilisé des images et des ressources graphiques pour créer des sprites pour le personnage du joueur, les murs, les caisses et les zones cibles. Ces ressources peuvent être créées à l'aide d'outils de conception graphique ou obtenues à partir de sources de ressources de jeu gratuites.
6. **Système d'exploitation** : J'ai utilisé un système d'exploitation de type desktop, comme Windows, pour le développement et les tests du jeu. Pygame est multiplateforme et peut fonctionner sur plusieurs systèmes d'exploitation.
7. **Outils de débogage** : J'ai utilisé des outils de débogage intégrés à l'IDE VS Code pour traquer et résoudre les problèmes de code.

En combinant ces outils et technologies, j'ai pu créer une interface de type desktop pour le jeu Sokoban en Python.

Capture d'écran de mon jeu Sokoban:



3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet dans la cadre de ma formation

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **La plateforme**

Chantier, atelier, service ▶ *Dans le cadre de ma formation : Concepteur développeur d'application web et mobile*

Période d'exercice ▶ Du : *17 Avril 2023* au : *30 Avril 2023*

5. Informations complémentaires (facultatif)

Activité-type 1

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n° 4 - Développer une interface de type desktop – Panel-admin

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

Lors de la conception et du développement des composants de l'interface utilisateur de l'application de chat et de l'interface de type desktop - Panel-admin, j'ai effectué diverses tâches pour garantir la meilleure expérience utilisateur tout en respectant les normes de sécurité. Voici une description détaillée des tâches que j'ai effectuées:

Conception de l'Interface Utilisateur:

Avant de commencer à coder, j'ai réalisé des maquettes de l'interface utilisateur de l'application en utilisant des outils de wireframing et de prototypage. Ces maquettes représentaient la disposition générale des composants de l'interface utilisateur, l'interaction entre les différentes pages et fonctionnalités, et l'expérience utilisateur globale.

Développement de l'Interface Utilisateur:

Après validation des maquettes, j'ai commencé le développement des composants de l'interface utilisateur. Pour cela, j'ai utilisé React.js comme framework et Material-UI comme bibliothèque de composants.

Material-UI: Material-UI est une bibliothèque populaire qui offre une variété de composants prêts à l'emploi conformes aux principes du Material Design. J'ai utilisé plusieurs de ces composants, tels que les boutons, les cartes, les dialogues et les formulaires, pour construire l'interface utilisateur de l'application. L'utilisation de Material-UI m'a permis d'obtenir une interface cohérente, réactive et attrayante avec un effort de développement minimal et surtout un gain de temps considérable.

Intégration des Recommandations de Sécurité:

J'ai également intégré des mesures de sécurité tout au long du développement de l'interface utilisateur. Par exemple, j'ai implémenté l'authentification à deux facteurs pour protéger l'accès aux comptes utilisateurs.

Material-UI m'a permis de créer une interface conviviale et intuitive qui permet aux administrateurs de gérer efficacement l'application. Voici quelques-unes des fonctionnalités que j'ai développées:

- **Tableau de bord:** J'ai utilisé des composants de Material-UI tels que les cartes pour afficher en temps réel les messages, et d'autres indicateurs clés.
- **Gestion des utilisateurs:** J'ai créé une page de gestion des utilisateurs avec des composants de table de Material-UI. Cela permet aux administrateurs d'ajouter, de supprimer et de modifier les droits d'accès des utilisateurs.
- **Fonctionnalités de modération:** J'ai ajouté des fonctionnalités pour modérer les messages et bannir les utilisateurs. Ces fonctionnalités ont été construites en utilisant des boîtes de dialogue et des boutons de Material-UI.

J'ai créé des Hooks personnalisés pour les appels API qui m'ont permis d'afficher les données des utilisateurs.

2. Précisez les moyens utilisés :

Moyens utilisés pour le développement du Panel-Admin :

1. **Environnement de développement** : Tout comme pour l'application de chat, j'ai utilisé Visual Studio Code comme environnement de développement intégré (IDE) pour le Panel-Admin. C'est un IDE riche en fonctionnalités qui offre une grande variété d'extensions utiles pour faciliter le développement.
2. **Langages de programmation et bibliothèques** : Le Panel-Admin a également été développé en utilisant JavaScript avec le framework React.js pour la partie front-end. Material-UI, une bibliothèque populaire de composants UI pour React, a été utilisée pour construire une interface utilisateur cohérente et attrayante.
3. **Hooks personnalisés** : J'ai créé et utilisé des hooks personnalisés pour encapsuler et réutiliser la logique de l'application. Ces hooks personnalisés ont aidé à rendre le code plus lisible et plus maintenable.
4. **Services** : J'ai utilisé des services pour encapsuler la logique des appels API, ce qui a permis de séparer la logique de gestion des données de la logique de l'interface utilisateur. Cela a rendu le code plus organisé et a facilité le test et la maintenance.
5. **APIs** : J'ai utilisé différentes APIs pour intégrer des fonctionnalités spécifiques dans le Panel-Admin, y compris une API pour gérer les utilisateurs et une autre pour modérer les messages.
6. **Outils de contrôle de version** : J'ai utilisé Git pour le contrôle de version, avec GitHub comme dépôt distant. Cela a permis de garder une trace de toutes les modifications du code, de travailler en équipe de manière efficace et de résoudre facilement les conflits de code.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet dans la cadre de ma formation

4. Contexte

Nom de l'entreprise, organisme ou association - **La plateforme**

Chantier, atelier, service - *Dans le cadre de ma formation : Concepteur développeur d'application web et mobile*

Période d'exercice - Du : *Avril 2023* au : *Mai 2023*

5. Informations complémentaires (facultatif)



Activité-type 2 *Concevoir et développer la persistance des données en intégrant les recommandations de sécurité*

Exemple n° 1 - *Concevoir une base de données*

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

Pour commencer la persistance des données fait référence à la capacité de stocker les données sur un support de stockage durable, tel qu'un disque dur ou une base de données, afin de les récupérer ultérieurement. Pour concevoir et développer la persistance des données, J'ai suivi les étapes suivantes :

1. **Identifier les besoins** : Tout d'abord, j'ai identifié les besoins de l'application pour laquelle la base de données sera utilisée. J'ai cherché à comprendre les types de données qui doivent être stockés, la fréquence à laquelle ces données seront utilisées et comment elles seront liées les unes aux autres.
2. **Créer un schéma conceptuel** : Ensuite, j'ai créé un schéma conceptuel, qui décrit les entités principales de la base de données, les relations entre ces entités, et les contraintes de l'application. Cela m'a aidé à clarifier la structure globale de la base de données avant de commencer à travailler sur les détails techniques. Le schéma de la base de données est la structure qui définit la façon dont les données sont stockées dans la base de données.
3. **Créer un schéma logique** : Après avoir créé le schéma conceptuel, j'ai créé le schéma logique, qui décrit les tables spécifiques de la base de données, les champs de chaque table et les clés primaires et étrangères qui établissent des relations entre les tables.
4. **Normaliser les données** : La normalisation des données est un processus qui permet d'organiser les données de manière à éliminer les redondances et à améliorer l'efficacité de la base de données. Cela implique de séparer les données en tables distinctes et de minimiser les champs répétitifs.
5. **Vérifier la qualité des données** : Il est important de vérifier la qualité des données pour s'assurer que les données stockées dans la base de données sont fiables et cohérentes. Cela implique de définir des règles d'intégrité de données, de valider les entrées utilisateur et de nettoyer les données avant de les stocker.

Il existe plusieurs outils et méthodes utilisés pour concevoir une base de données relationnelle efficace: Dans mon cas les méthodes utilisées sont les suivantes:

6. Le **MCD**, ou **Modèle Conceptuel de Données**, est un outil important dans la conception de bases de données relationnelles. Il permet de modéliser les entités et les relations entre ces entités dans la base de données. Il est utilisé pour définir la structure globale de la base de données, sans entrer dans les détails techniques. Il permet de décrire les
7. différentes entités qui composent la base de données, ainsi que les relations entre ces

entités. Chaque entité est décrite à l'aide de ses attributs, qui sont les caractéristiques ou les propriétés de l'entité.

8. Le **MLD**, ou **Modèle Logique de Données**, est un outil utilisé dans la conception de bases de données relationnelles pour représenter la structure de la base de données en utilisant un langage de modélisation formel. Le MLD est créé à partir du MCD (Modèle Conceptuel de Données) et il permet de définir la structure détaillée de la base de données et il permet de représenter la base de données sous forme de tables relationnelles.

Base de données phpMyadmin:

The screenshot shows the phpMyAdmin web interface. On the left is a sidebar with a tree view of databases and tables. The main area displays the 'tissapp' database with a table list. The table list has columns: Table, Action, Lignes, Type, Interclassement, Taille, and Perte. The tables listed are: comments, likes, notifications, posts, sequelizemeta, and users. Below the table list, there is a summary row for '6 tables' and a 'Somme' row. At the bottom, there is a button to 'Créer une nouvelle table' and a form with fields for 'Nom de table' and 'Nombre de colonnes'.

Table	Action	Lignes	Type	Interclassement	Taille	Perte
comments	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	16,0 kio	-
likes	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	32,0 kio	-
notifications	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	16,0 kio	-
posts	Parcourir Structure Rechercher Insérer Vider Supprimer	85	InnoDB	utf8mb4_general_ci	32,0 kio	-
sequelizemeta	Parcourir Structure Rechercher Insérer Vider Supprimer	13	InnoDB	utf8_unicode_ci	32,0 kio	-
users	Parcourir Structure Rechercher Insérer Vider Supprimer	28	InnoDB	utf8mb4_general_ci	32,0 kio	-
6 tables	Somme	126	InnoDB	utf8mb4_general_ci	160,0 kio	0 o

2. Précisez les moyens utilisés :

Lucidchart: C'est un outil en ligne qui facilite la création de diagrammes complexes. Je l'ai utilisé pour élaborer les diagrammes d'architecture, les organigrammes, et les diagrammes UML pour les applications que j'ai conçues. Cette plateforme a été essentielle pour la visualisation des processus de travail et la communication d'idées de conception au sein de l'équipe.

Draw.io Integration pour **VSCode** : Draw.io est un outil de diagramme flexible et gratuit. En utilisant l'extension Draw.io dans Visual Studio Code, j'ai pu facilement créer et éditer des diagrammes directement dans mon environnement de développement. Cela a amélioré l'efficacité de mon flux de travail et a permis une intégration plus fluide des diagrammes dans la documentation du projet.

3. Avec qui avez-vous travaillé ?

Samir , Salim, Ismail : Samir était un collègue avec qui j'ai travaillé sur plusieurs projets. Il était un développeur back-end talentueux et ensemble, nous avons réussi à réaliser plusieurs applications web complexes. Samir s'est toujours montré efficace dans son travail et a contribué à plusieurs idées innovantes pour nos projets.

Ismail : Ismail était à l'origine d'un certain nombre de nos initiatives. Sa connaissance approfondie du domaine nous ont aidés à naviguer efficacement dans le processus de développement.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **La plateforme**

Chantier, atelier, service ▶ **Dans le cadre de ma formation : Concepteur développeur d'application web**

Période d'exercice ▶ Du : *Janvier 2023* au : *Mai 2023*

5. Informations complémentaires (facultatif)

Activité-type 2 *Concevoir et développer la persistance des données en intégrant les recommandations de sécurité*

Exemple n°2 - *Développer des composants dans le langage d'une base de données*

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

Au cours de mes projets, j'ai accordé une grande importance à la persistance des données pour garantir que les informations restent disponibles et cohérentes même après que l'application a été fermée ou a subi une panne. J'ai utilisé des bases de données relationnelles et non relationnelles, comme **MySQL**, **MongoDB** pour stocker et récupérer les données de manière efficace.

Cela impliquait la création de modèles de données, l'écriture de requêtes complexes et l'optimisation de la performance de la base de données. Dans ce processus, j'ai toujours pris en compte les recommandations de sécurité pour protéger les données, notamment le chiffrement des données sensibles, l'utilisation de protocoles de transfert sécurisés et la mise en œuvre de mesures d'authentification et d'autorisation pour limiter l'accès aux données.

Ces tâches ont été réalisées dans le cadre de projets d'équipe, sous la direction d'un chef de projet "moi" que nous avons décidé ensemble et en collaboration avec Samir, Salim et Ismail. Les exigences de sécurité et de performance étaient souvent élevées, en particulier pour les projets impliquant des données sensibles ou confidentielles.

J'ai également travaillé sur le développement de composants spécifiques à la base de données. Cela inclut la création de procédures stockées et de fonctions pour encapsuler la logique de la base de données, l'écriture de déclencheurs pour automatiser certaines opérations et la mise en œuvre de vues pour simplifier l'accès aux données.

J'ai utilisé le langage SQL pour les bases de données relationnelles et le langage de requête spécifique pour les bases de données non relationnelles comme MongoDB. Ces tâches nécessitent une compréhension approfondie du langage de la base de données, une attention particulière aux détails et une capacité à résoudre des problèmes complexes.

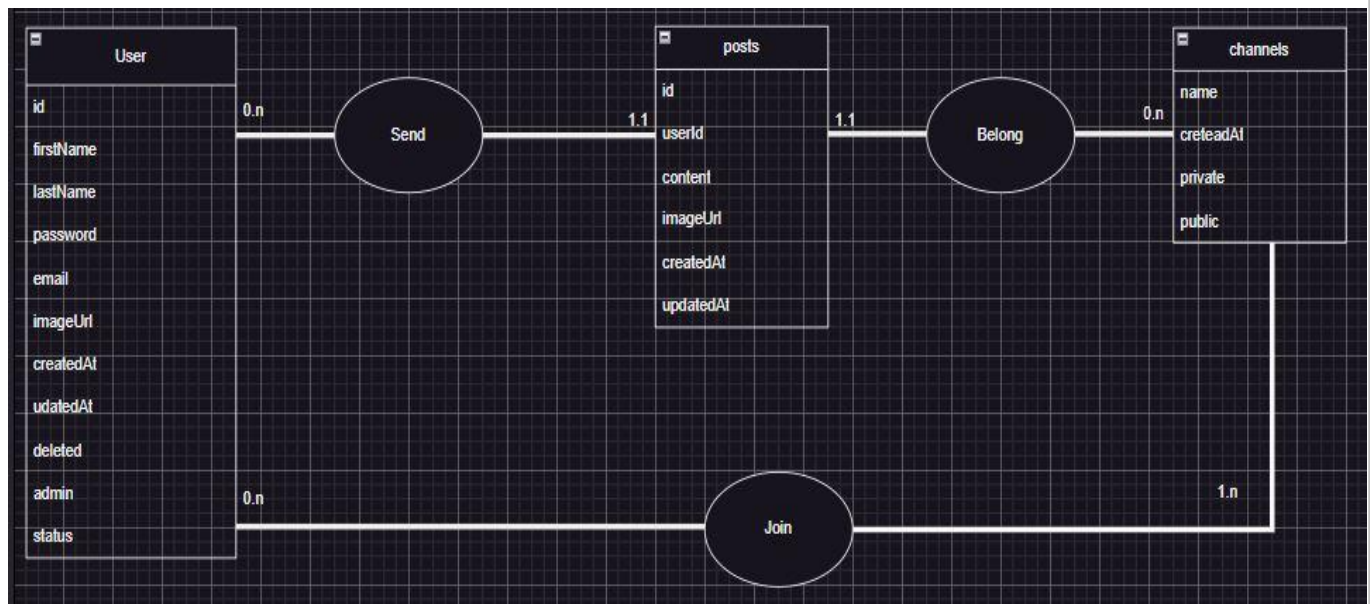
Ces tâches ont été effectuées dans le contexte de projets avec des exigences spécifiques en matière de gestion des données. J'ai souvent travaillé en collaboration avec d'autres membres de l'équipe pour assurer la cohérence de la conception et du développement de la base de données.

2. Précisez les moyens utilisés :

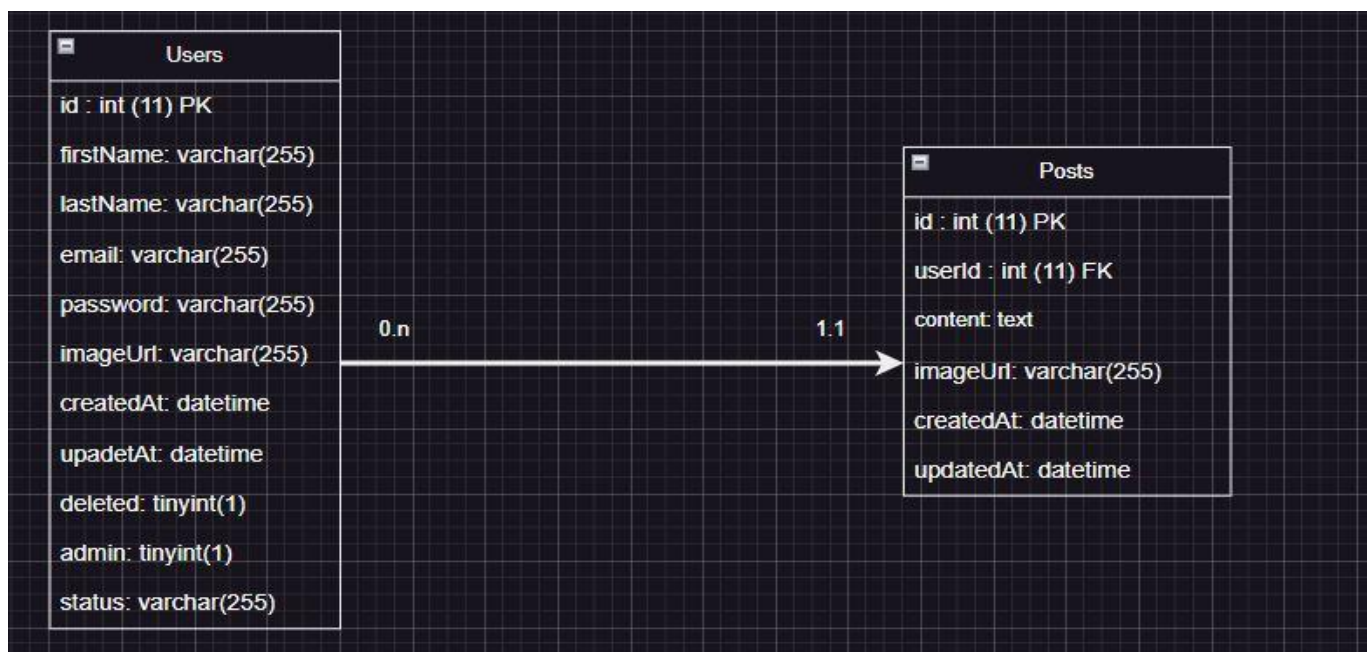
Pour concevoir et développer la persistance des données en intégrant les recommandations de sécurité, et pour développer des composants dans le langage d'une base de données, j'ai utilisé plusieurs outils et technologies :

Pour la persistance des données et la sécurité :

1. **Systèmes de gestion de bases de données** : J'ai utilisé des systèmes tels que MySQL, et xampp pour stocker et gérer les données de manière efficace.
2. **Modélisation des Données avec Sequelize** : J'ai utilisé Sequelize pour définir des modèles pour chaque entité dans l'application (par exemple, Utilisateurs, Messages). Ces modèles correspondent aux tables dans la base de données et définissent la structure des données.
3. **Outils de chiffrement** : J'ai utilisé diverses bibliothèques et outils de chiffrement pour protéger les données sensibles. Par exemple, **bcrypt** pour le hachage des mots de passe et **Json Web Token** pour la création de token d'authentification.
Outils de développement de base de données : J'ai utilisé des outils tels que SQL Workbench, phpMyAdmin, MongoDB Atlas(pour MongoDB), etc., pour développer et gérer les bases de données. J'ai aussi travaillé en entreprise avec Strapi qui est un CMS backend.
4. **Outils de modélisation de données** : J'ai utilisé des outils comme **Lucidchart** et **Draw.io** pour créer des diagrammes ER (Entity-Relationship) et visualiser la structure de la base de données.
5. **Environnement de développement intégré (IDE)** : J'ai utilisé des IDEs comme Visual Studio Code qui offre des extensions pour travailler efficacement avec les bases de données.



Exemple MLD de l'application TissApp:



3. Avec qui avez-vous travaillé ?

J'ai travaillé avec Salim , Samir et Ismail et Strapi avec Julien Atomas notre lead développeur en entreprise

4. Contexte

Nom de l'entreprise, organisme ou association - **La plateforme**

Chantier, atelier, service - **Dans le cadre de ma formation : Concepteur développeur d'application web et de mon alternance en entreprise (Atelier de la plateforme)**

Période d'exercice - Du : *Février 2023* au : *Avril 2023*

5. Informations complémentaires (facultatif)

Activité-type 2 *Concevoir et développer la persistance des données en intégrant les recommandations de sécurité*

Exemple n° 3 - Intégrer les recommandations de sécurité

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

Dans le développement de l'application de chat, assurer la persistance des données et l'intégration des recommandations de sécurité étaient des préoccupations primordiales. Voici les détails des tâches que j'ai effectuées :

Intégration des Recommandations de Sécurité

1. **Authentification et Autorisation** : J'ai intégré un système d'authentification basé sur des tokens JWT (JSON Web Token) pour garantir que seuls les utilisateurs authentifiés peuvent accéder à certaines routes et opérations. De plus, j'ai utilisé des mécanismes d'autorisation pour s'assurer que les utilisateurs n'ont accès qu'aux données qui leur sont pertinentes.
2. **Sécurité des Données** : Toutes les données sensibles, comme les mots de passe des utilisateurs, ont été sécurisées en utilisant des méthodes de hachage avec bcrypt. Les connexions ont été sécurisées en utilisant le protocole HTTPS.
3. **Sécurisation des API** : J'ai mis en place des limites de taux pour prévenir les attaques. J'ai également validé et nettoyé toutes les entrées des utilisateurs pour prévenir les injections SQL et les attaques XSS.

Ces tâches ont été réalisées dans le respect des meilleures pratiques de sécurité et en accord avec les spécifications du projet. Mon objectif était de garantir la sécurité des données des utilisateurs tout en fournissant une expérience utilisateur fluide et réactive.

2. Précisez les moyens utilisés :

Lors de la conception et du développement de la persistance des données, ainsi que de l'intégration des recommandations de sécurité, j'ai utilisé les moyens suivants :

1. **Sequelize** : Comme ORM (Object-Relational Mapping) pour Node.js, j'ai utilisé Sequelize pour faciliter les interactions avec la base de données SQL. Sequelize a été utilisé pour définir des modèles de données, exécuter des opérations CRUD et gérer les transactions.
2. **MySQL** : J'ai utilisé MySQL comme système de gestion de base de données. C'est une base de données relationnelle qui est largement utilisée et fiable.
3. **Node.js et Express.js** : Ces technologies ont été utilisées pour créer le serveur backend et développer les API RESTful.
4. **JWT (JSON Web Tokens)** : J'ai utilisé JWT pour l'authentification et l'autorisation. Les JWT sont des tokens qui permettent de transmettre des informations de manière sécurisée entre les parties.
5. **Bcrypt** : Pour sécuriser les mots de passe des utilisateurs, j'ai utilisé la bibliothèque Bcrypt pour hacher et saler les mots de passe avant de les stocker dans la base de données.
6. **HTTPS** : J'ai utilisé le protocole HTTPS pour sécuriser la communication entre le client et le serveur.
7. **Validation des Entrées** : Pour prévenir les injections SQL et les attaques XSS, j'ai utilisé des bibliothèques de validation des entrées.
8. **Outils de Développement** : Pour le développement, j'ai utilisé des outils comme Visual Studio Code (L'éditeur de code), Postman (pour tester les API), et Git pour le contrôle de version.

Ces moyens ont été utilisés pour assurer la persistance des données et la sécurité dans l'application de chat.

3. Avec qui avez-vous travaillé ?

J'ai travaillé en collaboration avec Samir, Salim et Ismail

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **La plateforme.**

Chantier, atelier, service ▶ **Dans le cadre de ma formation : Concepteur développeur d'application web**

Période d'exercice ▶ Du : Cliquez ici au : Cliquez ici

DOSSIER PROFESSIONNEL ^(DP)

5. Informations complémentaires *(facultatif)*

--

Activité-type 3

Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité

Exemple n°1 - Collaborer à la gestion d'un projet informatique et à l'organisation de l'environnement de développement

2. Précisez les moyens utilisés :

Dans le contexte des tâches et des opérations que j'ai précédemment décrites, voici les principaux moyens et outils que j'ai utilisés :

1. **Outils de développement** : Pour le codage de l'application, j'ai utilisé **Visual Studio Code** comme environnement de développement intégré (**IDE**). Pour le développement de la base de données, j'ai utilisé **PHPMyAdmin Mysql**, et pour le développement du back-end et du front-end, j'ai utilisé respectivement **Node.js** et **React Native**
2. **Outils de gestion de version** : J'ai utilisé **Git** pour le contrôle de version du code, ce qui permettait à toute l'équipe de collaborer efficacement sur le même code base. Les dépôts Git étaient hébergés sur **GitHub**, offrant une interface facile à utiliser pour la gestion de version..
3. **Outils de gestion de projet** : Pour planifier, suivre et gérer le projet, nous avons utilisé des outils comme **Jira** et **Trello**. Ces outils nous ont aidé à organiser les tâches, à suivre leur progression et à collaborer plus efficacement en tant qu'équipe.
4. **Outils de communication** : Pour faciliter la communication en équipe, nous avons utilisé des outils comme **Pass Bolt** pour le partage des données sensibles, et **Google meet** pour les réunions d'équipe plus formelles. Les e-mails ont également été utilisés pour les communications officielles et les échanges avec des parties externes.

3. Avec qui avez-vous travaillé ?

Chef de projet (Ana Stonko): qui organisent et gèrent le projet, s'assurant que tout est livré en temps et en heure. Ils jouent un rôle essentiel dans la coordination de l'équipe et la gestion des tâches et des délais. Et pour le projet de groupe c'est en collaboration avec Samir, Salim et Ismail pour concevoir et développer l'application de Chat.
J'ai également collaboré avec notre lead développeur Julien Athomas, pour assurer la qualité de l'application.

4. Contexte

DOSSIER PROFESSIONNEL ^(DP)

Nom de l'entreprise, organisme ou association ▶

L'atelier de la plateforme et la plateforme

Chantier, atelier, service ▶

Dans le cadre de ma formation : Concepteur développeur d'application web et en entreprise

Période d'exercice ▶

Du : Cliquez ici

au :

Cliquez ici

5. Informations complémentaires (facultatif)

Activité-type 3 *Concevoir et développer une application multicouche répartie en intégrant les recommandations de sécurité*

Exemple n°2 - *Développer des composants métiers*

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

Le développement de composants métiers est une étape cruciale dans la création d'une application. Les composants métiers sont des parties de l'application qui réalisent les fonctionnalités spécifiques à l'activité de l'entreprise ou à l'objectif de l'application. Dans le cadre du développement d'une application de chat, voici une description des tâches ou opérations spécifiques que j'ai effectuées et les conditions dans lesquelles elles ont été effectuées :

1. **Compréhension des exigences** : En collaboration avec l'équipe, j'ai aidé à définir et à comprendre les exigences spécifiques de l'application de chat, comme les fonctionnalités nécessaires, les attentes des utilisateurs, et les objectifs de l'entreprise. Nous avons discuté des fonctionnalités clés, comme les discussions privées et de groupe, l'envoi de d'images et de messages en temps réels
2. **Conception des composants métier** : J'ai traduit les exigences en conceptions de haut niveau pour chaque composant. Par exemple, pour la gestion des messages, j'ai conçu une architecture qui comprenait la façon dont les messages seraient créés, comment ils seraient stockés dans la base de données, comment ils seraient récupérés pour être affichés à l'utilisateur, etc. J'ai également pris en compte des questions telles que la confidentialité et la sécurité des messages, la charge de travail potentielle sur le serveur, et la rapidité de la livraison des messages.
3. **Développement des composants métier** : Cette étape a impliqué beaucoup de codage. Par exemple, pour le composant de gestion des messages, j'ai écrit des fonctions pour créer de nouveaux messages, les stocker dans la base de données, les récupérer en réponse à une demande de l'utilisateur, et les envoyer à tous les participants d'une conversation. J'ai également écrit des fonctions pour gérer les erreurs, comme une demande de récupération d'un message qui n'existe pas. Pendant tout ce temps, j'ai veillé à suivre les meilleures pratiques de codage, en écrivant un code clair, bien commenté et efficace.
4. **Tests des composants métier** : Pour chaque composant, j'ai écrit une série de tests automatisés. Ces tests vérifient que chaque fonction faisait correctement ce qu'elle était censée faire, avec des entrées valides et invalides. J'ai utilisé des frameworks de test comme Jest pour écrire et exécuter ces tests.
5. **Documentation** : J'ai rédigé une documentation détaillée pour chaque dossier dans le backend , frontend et panel admin: J'ai créé un fichier readme qui explique comment cloner le projet et comment le démarrer pour les tester.

6. **Revue de code** : À chaque fois que j'ai terminé un composant, j'ai demandé à un ou plusieurs autres développeurs de l'équipe de revoir mon code. Ils ont vérifié la logique, la clarté, l'efficacité, et la conformité aux normes de l'équipe. J'ai ensuite apporté les modifications nécessaires en fonction de leurs commentaires.

2. Précisez les moyens utilisés :

Dans le développement de l'application de chat d'entreprise, les moyens utilisés étaient divers et incluent des outils de développement logiciel, des plateformes de gestion de projet et des services cloud. Voici une liste plus détaillée :

1. **Environnements de développement** : J'ai utilisé Visual Studio Code comme environnement de développement intégré (IDE). C'est un outil puissant et flexible qui offre de nombreuses fonctionnalités utiles, notamment l'édition de code, le débogage et l'intégration Git.
2. **Langages de programmation et frameworks** : J'ai utilisé **Node.js** pour le backend de l'application et **React Native** pour le front-end. Pour la gestion des données en temps réel, j'ai utilisé **Socket.io**. Ces technologies sont largement utilisées dans le développement d'applications web modernes et offrent de nombreuses fonctionnalités utiles pour le développement d'applications de chat.
3. **Base de données** : J'ai utilisé **PHPMyadmin** comme système de gestion de base de données.
4. **Gestion de version** : J'ai utilisé Git pour le contrôle de version du code. Git permet à plusieurs développeurs de travailler sur le même projet sans se gêner mutuellement.
5. **Outils de collaboration et de gestion de projet** : J'ai utilisé Trello et Jira pour la gestion des tâches et la planification des sprints.
6. **Tests** : J'ai utilisé Jest pour écrire et exécuter des tests unitaires et de fonctionnalités.

3. Avec qui avez-vous travaillé ?

J'ai réalisé ce travail avec Samir, Salim et Ismail

4. Contexte

Nom de l'entreprise, organisme ou association ▶ *La plateforme*

Chantier, atelier, service ▶

Dans le cadre de ma formation : Concepteur développeur d'application web

Période d'exercice ▶ Du : *Janvier 2023* au *Mars 2023*

Activité-type 3 *Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité*

Exemple n° 3 ►

Construire une application organisée en couche

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

Dans le processus de construction d'une application organisée en couches, j'ai effectué plusieurs tâches importantes pour assurer une structure propre et maintenable. Voici les étapes que j'ai suivies :

Planification et Conception

1. **Compréhension des Exigences** : J'ai d'abord passé du temps à comprendre les besoins et les exigences du projet. Cela m'a aidé à déterminer combien de couches seraient nécessaires et quels seraient leurs rôles respectifs.
2. **Conception de l'architecture** : Sur la base des exigences, j'ai conçu une architecture en couches pour l'application. Typiquement, l'application a été divisée en quatre couches principales : la couche de présentation (**frontend**), la couche de logique métier (**backend**), la couche admin (**panel-admin**) et la couche d'accès aux données.

Développement

1. **Développement de la couche de Présentation** : J'ai utilisé React Native pour développer l'interface utilisateur de l'application. Cette couche était responsable de la présentation des données à l'utilisateur et de la collecte des entrées de l'utilisateur.
2. **Développement de la couche de Logique Métier** : J'ai utilisé Node.js et Express.js pour développer la logique métier de l'application. Cette couche était responsable de la gestion des requêtes de l'utilisateur, de l'exécution de la logique appropriée, et de l'envoi des réponses.
3. **Développement de la couche d'Accès aux Données** : J'ai utilisé Sequelize pour interagir avec la base de données MySQL. Cette couche était responsable de la gestion de toutes les interactions avec la base de données, y compris la récupération, l'insertion, la mise à jour et la suppression des données.

Intégration et Test

1. **Intégration des couches** : Après le développement de chaque couche, j'ai travaillé à leur intégration. Cela impliquait de s'assurer que chaque couche pouvait communiquer efficacement avec les autres.
2. **Test de l'Application** : Une fois l'application développée et intégrée, j'ai effectué des tests rigoureux pour m'assurer que tout fonctionnait comme prévu. J'ai utilisé des outils de test automatique pour tester la fonctionnalité de l'application et m'assurer qu'elle répondait aux exigences.

Ces tâches ont été effectuées dans un environnement de développement professionnel, en respectant les meilleures pratiques et normes de l'industrie. Mon objectif était de garantir la qualité du code, la performance de l'application, et finalement, la satisfaction de l'utilisateur.

2. Précisez les moyens utilisés :

Pour la construction d'une application organisée en couches, j'ai utilisé les moyens suivants :

1. **React Native** : Pour la couche de présentation, j'ai utilisé React Native. Il s'agit d'un framework pour le développement d'applications mobiles en JavaScript qui permet de créer des interfaces utilisateur réactives et performantes.
2. **Node.js et Express.js** : Pour la couche de logique métier, j'ai utilisé Node.js, une plateforme de développement d'applications côté serveur, et Express.js, un framework pour Node.js.
3. **Sequelize** : Pour la couche d'accès aux données, j'ai utilisé Sequelize, un ORM pour Node.js. Sequelize facilite l'interaction avec la base de données SQL. **MySQL** : Comme système de gestion de base de données, j'ai utilisé MySQL. Il s'agit d'une base de données relationnelle largement utilisée qui offre une grande flexibilité et des performances fiables.
4. **Visual Studio Code** : J'ai utilisé Visual Studio Code comme éditeur de code. Il offre une gamme d'outils utiles pour le développement, tels que la coloration syntaxique, le formatage automatique du code, et le support pour de nombreux langages de programmation.
5. **Git** : Pour le contrôle de version, j'ai utilisé Git. Il permet de suivre les modifications apportées au code, facilite la collaboration avec d'autres développeurs, et aide à prévenir la perte de travail en cas de problèmes.
6. **Postman** : Pour tester les API, j'ai utilisé Postman. Il permet d'envoyer des requêtes à l'API et de vérifier les réponses.
7. **Jest** : Pour les tests unitaires et d'intégration, j'ai utilisé Jest, un framework de test en JavaScript.

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec Salim, Samir et Ismail

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **La plateforme.**

Chantier, atelier, service ▶ **Dans le cadre de ma formation : Concepteur développeur d'application web**

Période d'exercice ▶ Du : **Janvier 2023** au : **Mai 2023**

Activité-type 3

Concevoir et développer des composants d'interface utilisateur en intégrant les recommandations de sécurité

Exemple n° 4 - Préparer et exécuter le déploiement d'une application

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

Le déploiement d'une application est une étape cruciale du développement logiciel. Pour le déploiement d'une application de blog en Symfony, voici les tâches que j'ai réalisées :

1. **Test et Validation** : J'ai réalisé une série de tests unitaires et fonctionnels pour vérifier le bon fonctionnement de chaque partie de l'application. J'ai utilisé **API Platform**, un framework de test standard pour les applications Symfony. J'ai également vérifié que toutes les fonctionnalités du blog fonctionnaient correctement sur différents appareils et navigateurs.
2. **Optimisation** : J'ai exécuté les commandes de Symfony pour vider le cache et optimiser l'Autoloader pour améliorer les performances de l'application en production. De plus, j'ai utilisé des outils comme CSS Minifié et JavaScript Minifier pour réduire la taille des fichiers CSS et JavaScript.
3. **Préparation de l'Environnement de Production** : Dans l'interface de Plesk, j'ai créé un nouveau domaine pour l'application et installé un certificat SSL pour sécuriser les communications entre le serveur et les clients. J'ai également créé une nouvelle base de données MySQL pour l'application et un utilisateur avec les permissions nécessaires.
4. **Transfert de l'Application** : J'ai utilisé un client FTP comme FileZilla pour transférer tous les fichiers de l'application du serveur de développement au serveur de production. J'ai également exporté la base de données du serveur de développement à l'aide de phpMyAdmin, puis je l'ai importée dans la nouvelle base de données sur le serveur de production.
5. **Configuration de Symfony** : J'ai modifié le fichier .env de Symfony pour mettre à jour les informations de connexion à la base de données et définir l'environnement sur prod pour activer le mode de production.
6. **Installation des Dépendances** : J'ai utilisé Composer pour installer toutes les dépendances requises par l'application. En exécutant la commande `composer install --no-dev --optimize-autoloader`, j'ai installé seulement les dépendances nécessaires pour la production et optimisé l'autoloader pour améliorer les performances.
7. **Démarrage de l'Application** : Une fois que tout a été configuré, j'ai accédé à l'application via le navigateur pour vérifier qu'elle fonctionne correctement. J'ai navigué sur plusieurs pages du blog, créé un nouveau post, commenté un post existant, etc. pour m'assurer que toutes les fonctionnalités fonctionnent comme prévu.

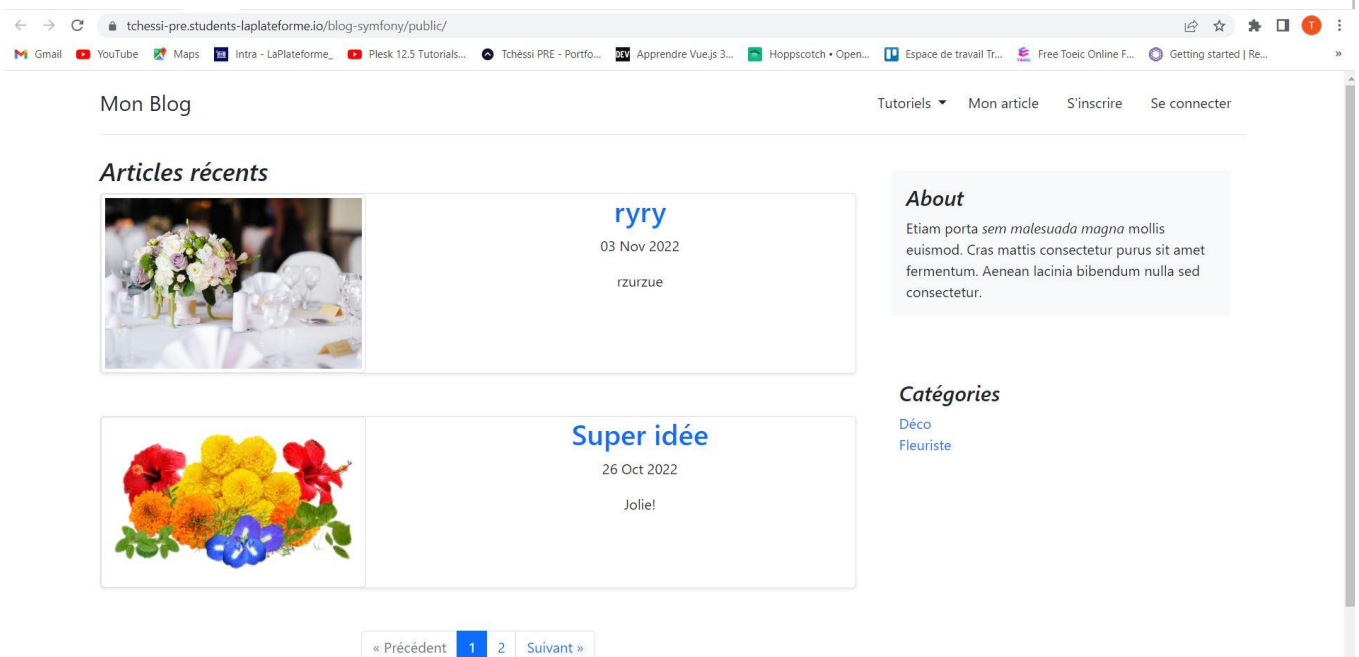
Chaque étape du déploiement a été effectuée avec soin pour éviter tout temps d'arrêt inutile et garantir la meilleure expérience possible pour les utilisateurs finaux. La moindre erreur dans ce processus peut entraîner des problèmes importants, il est donc crucial de faire preuve de rigueur et de précision.

2. Précisez les moyens utilisés :

Pour déployer une application Symfony avec API Platform sur Plesk, voici les outils et technologies que j'ai utilisés :

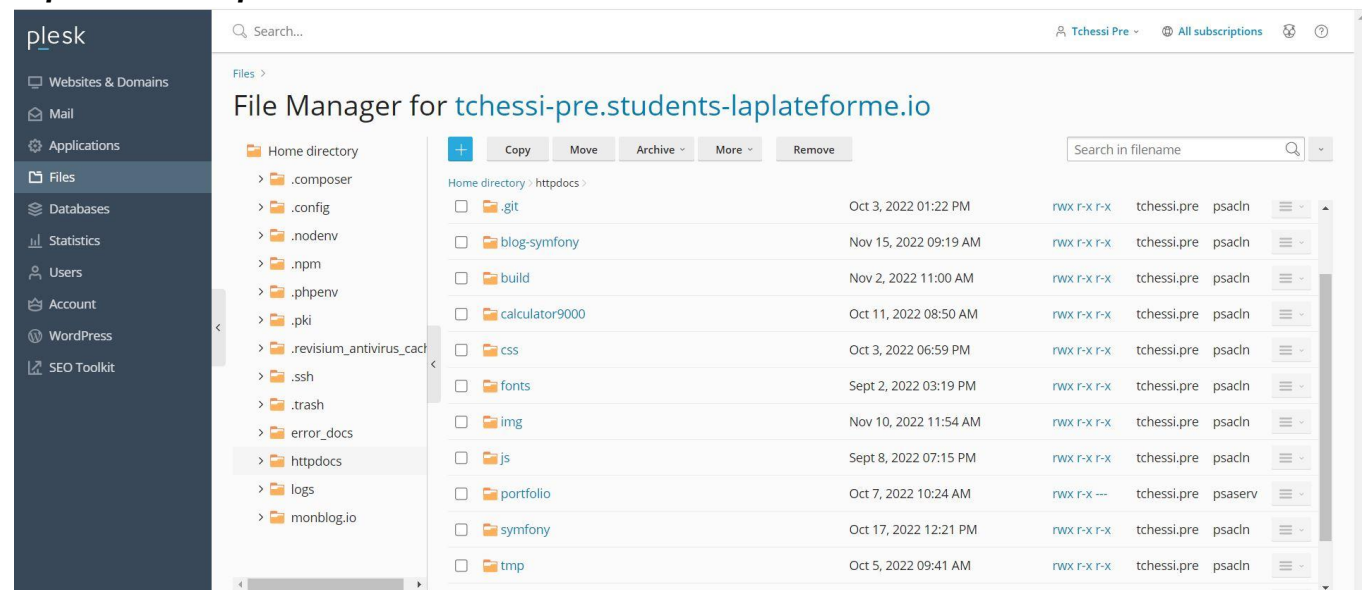
1. **Symfony** : Il s'agit du framework PHP que j'ai utilisé pour construire l'application de blog. Symfony fournit une structure et une méthodologie pour développer des applications web de qualité.
2. **API Platform** : Ce framework, construit sur Symfony, me permet de créer rapidement des APIs web puissantes. Il facilite la création d'APIs REST et GraphQL et offre des fonctionnalités telles que la validation automatique, la pagination, la génération de documentation et plus encore. Et aussi pour tester les APIs créées avec API Platform et vérifier que toutes les requêtes renvoient les bonnes réponses. SSL : Pour sécuriser les communications entre le serveur et les clients, j'ai utilisé un certificat SSL, que j'ai installé et configuré à travers Plesk.
- 3.
4. **MySQL** : J'ai utilisé MySQL comme système de gestion de base de données pour stocker les données de l'application.
5. **Plesk** : Plesk est l'outil de gestion de serveur web que j'ai utilisé pour gérer le déploiement de l'application. Il me permet de configurer les domaines, les bases de données, les certificats SSL et d'autres aspects du serveur web.
6. **FileZilla** : J'ai utilisé ce client FTP pour transférer les fichiers de l'application du serveur de développement au serveur de production.
7. **Composer** : C'est un outil de gestion des dépendances pour PHP. J'ai utilisé Composer pour installer et gérer les dépendances de l'application Symfony et de l'API Platform. Postman : J'ai utilisé Postman pour tester les APIs créées avec API Platform et vérifier que toutes les requêtes renvoient les bonnes réponses. SSL : Pour sécuriser les communications entre le serveur et les clients, j'ai utilisé un certificat SSL, que j'ai installé et configuré à travers Plesk.

Voici le rendu:



En utilisant ces outils et technologies, j'ai été capable de déployer efficacement l'application Symfony avec API Platform sur un serveur géré par Plesk.

Capture d'écran plesk:



3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet dans le cadre de ma formation.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **La plateforme.**

Chantier, atelier, service ▶ **Dans le cadre de ma formation : Concepteur développeur d'application web (création d'un blog en symfony php)**

Période d'exercice ▶ Du : **24 Septembre 2022** au : **28 Octobre 2022**

Activité-type 3

Exemple n°5 - Préparer et exécuter les plans de tests d'une application

2. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions

Voici les tâches que j'ai réalisées :

1. **Identification des scénarios de test** : Pour chaque fonctionnalité de l'application de chat, j'ai identifié plusieurs scénarios de test. Par exemple, pour la fonctionnalité d'envoi de messages, les scénarios de test pourraient inclure l'envoi d'un message à un autre utilisateur, l'envoi d'un message à un groupe, l'envoi d'un message contenant des emojis, etc.
2. **Création du Plan de Test** : J'ai ensuite créé un plan de test détaillé. Pour chaque scénario de test, j'ai spécifié l'ordre d'exécution, les étapes à suivre, les données d'entrée à utiliser (par exemple, le texte du message à envoyer), les conditions préalables (par exemple, les deux utilisateurs doivent être amis), et le résultat attendu (par exemple, le message apparaît dans la fenêtre de chat du destinataire).
3. **Préparation de l'Environnement de Test** : J'ai mis en place un environnement de test spécifique, distinct de l'environnement de production, pour exécuter les tests. J'ai créé plusieurs comptes d'utilisateurs pour les tests, et j'ai préparé des scripts pour réinitialiser l'environnement de test à un état connu avant chaque série de tests.
4. **Exécution des Tests** : J'ai exécuté les tests selon le plan de test. Pour chaque test, j'ai suivi les étapes spécifiées, en utilisant les données d'entrée indiquées, et j'ai vérifié que le résultat correspondait au résultat attendu. Si un test échouait, je notais tous les détails pertinents, comme le message d'erreur affiché ou l'état de l'application après l'échec du test.
5. **Rapport de Test** : J'ai compilé les résultats de tous les tests dans un rapport de test. Ce rapport indiquait clairement le nombre de tests qui avaient réussi et ceux qui avaient échoué, et fournissait des détails sur les échecs de test, y compris les étapes de reproduction, les captures d'écran, les logs, etc.
6. **Gestion des Bugs** : Pour chaque échec de test, j'ai créé un ticket de bug dans notre système de suivi des bugs. J'ai fourni une description détaillée du bug, les étapes pour le reproduire, la gravité du bug, et toute autre information qui pourrait aider les développeurs à le résoudre.

Capture d'écran: annexe -3

2. Précisez les moyens utilisés :

Voici les outils et technologies que j'ai utilisés pour préparer et exécuter les plans de tests pour l'application de chat :

1. **React Native** : C'est le framework que j'ai utilisé pour développer l'application de chat. React Native offre une grande flexibilité pour tester l'application à la fois sur Android et iOS.
2. **Jest** : Jest est un framework de tests JavaScript largement utilisé, parfait pour tester les

applications React et React Native. Il me permet d'écrire des tests unitaires et des tests de composants pour mon application.

3. **Postman** : J'ai utilisé Postman pour tester les APIs de l'application. Avec Postman, je peux envoyer des requêtes HTTP à mes APIs et vérifier que les réponses sont correctes.
Git : Git est un système de contrôle de version que j'ai utilisé pour gérer le code source de l'application. Avec Git, je peux facilement créer des branches pour isoler le développement de nouvelles fonctionnalités et le travail de correction de bugs.
4. **Trello** est un outil de gestion de projet et de suivi des bugs que j'ai utilisé pour organiser mon travail et suivre les problèmes identifiés lors des tests.

En combinant ces outils, j'ai pu mettre en place un processus de test complet qui couvre à la fois les tests unitaires, les tests de composants, les tests d'intégration

Exemple de Tests: Annexe 3

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur le test de l'application.

4. Contexte

Nom de l'entreprise, organisme ou association ▶ **La plateforme**

Chantier, atelier, service ▶ **Dans le cadre de ma formation : Concepteur développeur d'application web et en entreprise (Application mobile de Chat)**

Période d'exercice ▶ Du : *Janvier 2023* au : *Mai 2023*

5. Informations complémentaires (facultatif)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Développeur web	OpenClassroom	Juin 2022 - Avril 2022

Déclaration sur l'honneur

Je soussigné(e) **Tchèssi PRE** ,
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 **Marseille** le **29 Mai 2023**

pour faire valoir ce que de droit.

Signature :

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

Annexe 1: Rendus de mon application de Chat

Annexe 2: Rendus du panel admin

Maquettes d'interface : Les maquettes de l'interface utilisateur de l'application de chat, montrant comment les utilisateurs peuvent interagir avec l'application.

<https://www.figma.com/file/pCwIYbexxiYcB8oRVCCwf5/Untitled?node-id=0%3A1&t=QOBwCJjSdCwnd9wU-0>

Code source : Des extraits de code source qui montrent comment j'ai implémenté des fonctionnalités clés, comme la gestion des messages, la connexion des utilisateurs

Lien github: <https://github.com/Tchessi/app-mobile-chat/tree/tch-v-3>

Documentation :

Postman:

<https://documenter.getpostman.com/view/17540434/2s8Z72VX2N>

Maquette: figma

<https://www.figma.com/file/L9vcJYA2TyRMPZKXOhoWYB/V-2?type=design&t=95Gdq8kEAzqg2ukc-0>

Trello

<https://trello.com/invite/tchessisamirismailsalim/ATTI9472e90e9d0b4861b4b1a72e24a8ecd3FCA49511>

Autres:

<https://reactnative.dev/>

<https://www.npmjs.com/>

<https://sequelize.org/>

<https://mui.com/>

Débogage: <https://stackoverflow.com/>

<https://www.google.fr/>

Documents illustrant la pratique professionnelle

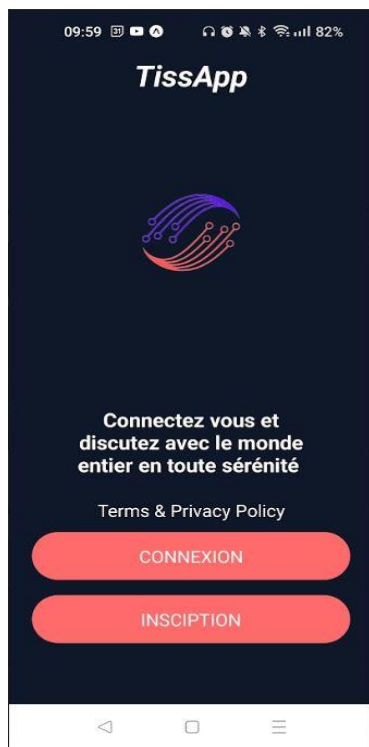
(facultatif)

Intitulé
Annexe 3: Test unitaires

ANNEXES

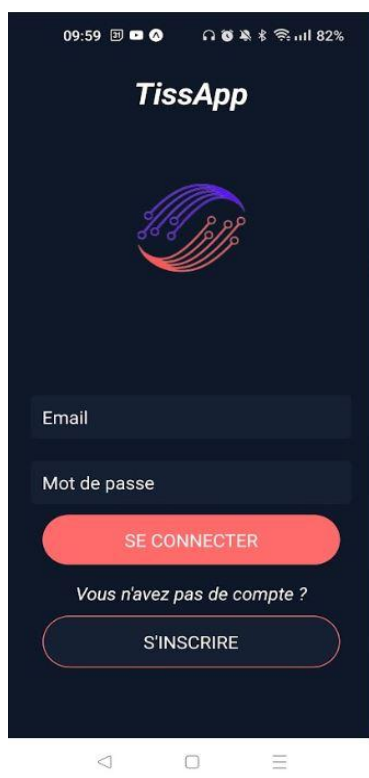
(Si le RC le prévoit)

Annexe 1: Rendu de mon application de Chat



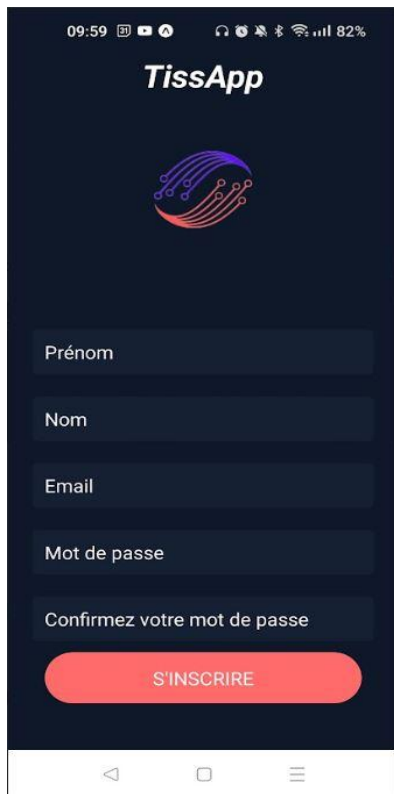
Page d'accueil :

La page d'accueil de l'application offre une introduction élégante à l'application de chat, mettant en avant ses caractéristiques principales et fournissant des informations sur comment et pourquoi l'utiliser. Elle comporte également des boutons pour se connecter et s'inscrire, permettant aux utilisateurs de commencer à utiliser l'application rapidement.



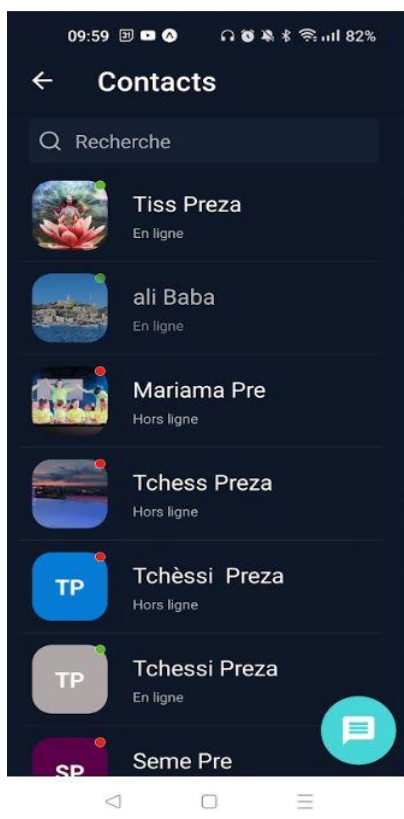
Page de connexion :

Cette page fournit un formulaire simple et sécurisé pour que les utilisateurs puissent se connecter à leurs comptes. Les utilisateurs entrent leur email et leur mot de passe, et l'authentification est gérée en arrière-plan avec des mesures de sécurité appropriées.



Page d'inscription :

La page d'inscription fournit un formulaire d'inscription pour les nouveaux utilisateurs. Elle comprend des champs pour les informations nécessaires comme le nom, prénom, l'email et le mot de passe. Des contrôles de validation sont en place pour s'assurer que les mots de passe sont forts et que les adresses email sont valides.



Page des contacts :

La page des contacts montre une liste des contacts de l'utilisateur. Les utilisateurs peuvent rechercher des contacts spécifiques, cliquer sur un contact pour voir leur profil. Chat contact a un badge qui montre si l'utilisateur est en ligne ou hors ligne. Celui-ci contient un bouton qui redirige vers la page de chat général.



Page des discussions :

La page des discussions montre une liste de toutes les discussions auxquelles l'utilisateur participe. Les utilisateurs peuvent cliquer sur une discussion pour l'ouvrir dans la page de chat. Cette page permet aux utilisateurs de gérer efficacement leurs conversations et de passer facilement d'une discussion à l'autre. Les utilisateurs peuvent cliquer sur les photos de profil qui redirigent vers la page de profil de l'utilisateur associé.



Page de profil :

La page de profil permet aux utilisateurs de voir et de modifier leurs informations de profil. Ils peuvent changer leur photo de profil, mettre à jour leur nom et prénom mais pas leur email. Cette page est conçue pour être intuitive et facile à utiliser.

Elle contient un bouton modifier qui permet de modifier et d'enregistrer la mise à jour et aussi un bouton de déconnexion



Page de chat :

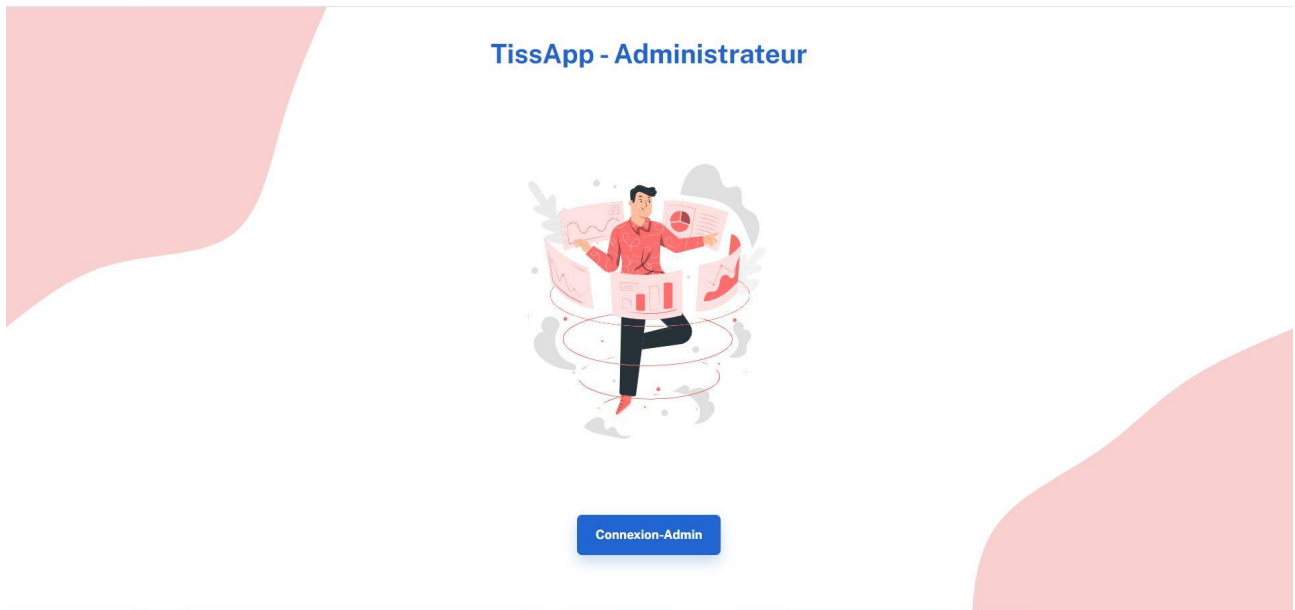
La page de chat est l'endroit où les utilisateurs peuvent envoyer et recevoir des messages. Les messages sont affichés dans une interface conviviale qui permet de voir clairement la conversation. Les utilisateurs peuvent envoyer des messages texte et éventuellement des emojis, des images et d'autres types de messages. L'utilisateur peut afficher une image en appuyant sur l'image. Un utilisateur peut supprimer son message avec un appui long sur le message à supprimer.

Dans l'ensemble, l'application de chat est conçue pour offrir une expérience utilisateur fluide et agréable. Chaque page a été soigneusement conçue avec une attention particulière portée à l'ergonomie, la convivialité et la cohérence du design.

Annexe 2: Rendu du panel admin

Page d'accueil :

La page d'accueil du Panel-Admin est simple et claire, offrant un accès facile à la page de connexion via un bouton dédié. Cela permet aux administrateurs d'accéder rapidement à l'outil de gestion.



Page de connexion :

La page de connexion offre un formulaire sécurisé pour que les administrateurs puissent se connecter à leurs comptes. L'authentification se fait à travers l'entrée d'un identifiant (email) et d'un mot de passe, avec des mesures de sécurité appropriées en place pour protéger l'accès.

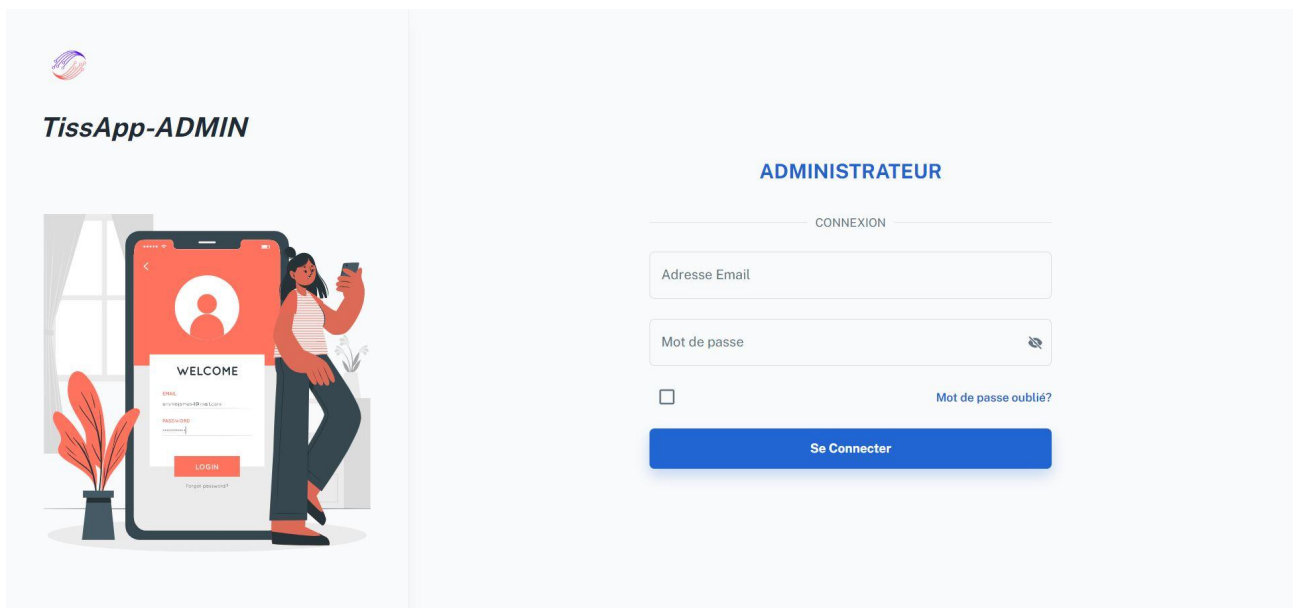
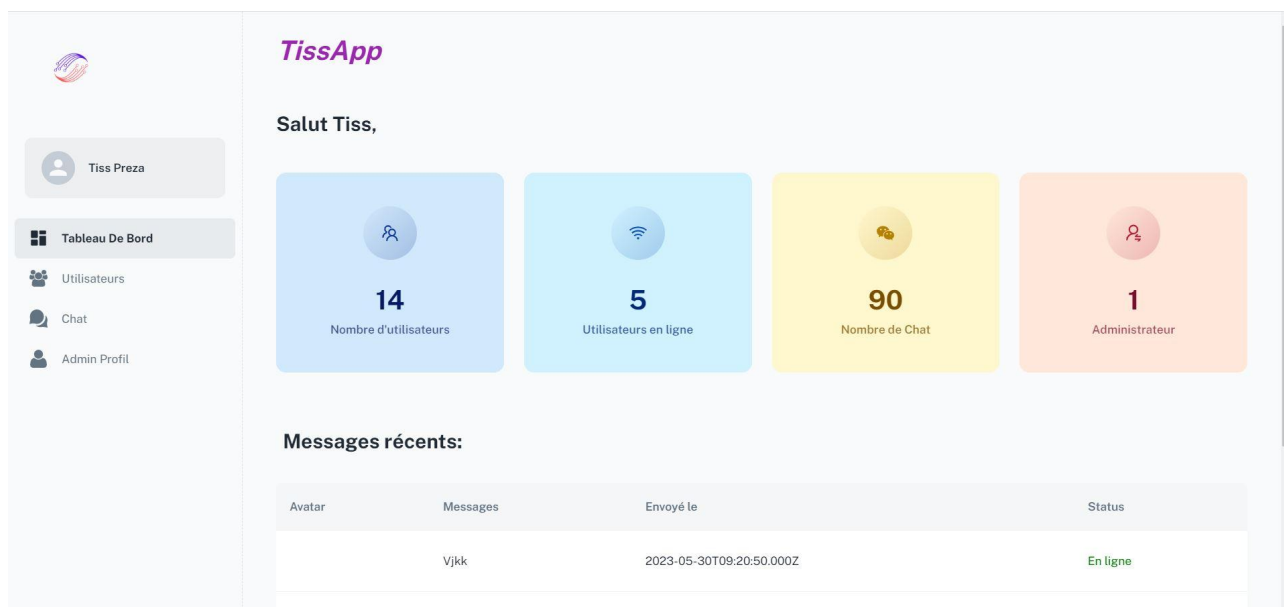


Tableau de bord :

Le tableau de bord est le cœur de l'interface d'administration. Il affiche des informations clés sur l'état de l'application, y compris le nombre total d'utilisateurs, le nombre de messages échangés, le nombre d'administrateurs, et le nombre de personnes connectées actuellement. Il présente également les quatre derniers messages envoyés, ce qui permet aux administrateurs de surveiller l'activité de l'application en temps réel.



Page des utilisateurs :

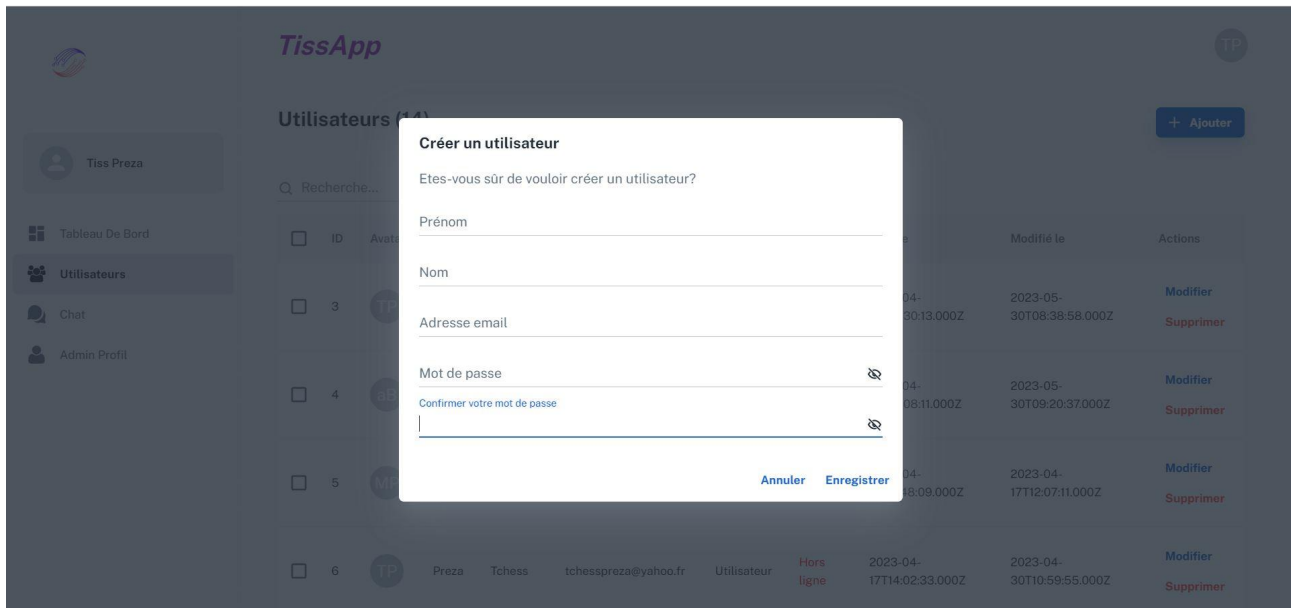
Cette page affiche une liste complète de tous les utilisateurs de l'application. Les administrateurs peuvent avoir des informations détaillées sur chaque utilisateur et peuvent avoir la possibilité de modérer les comptes si nécessaire. Elle contient une barre de recherche pour rechercher un utilisateur grâce à son nom et son prénom. Elle affiche dans un tableau le nom, prénom, email, le rôle, la date de création, la date de mise à jour et un bouton supprimer, modifier et aussi une pagination.

The user management page for TissApp shows a list of 14 users. The table includes columns for ID, Avatar, Nom, Prénom, Email, Rôle, Status, Créé le, Modifié le, and Actions. The actions column contains links for 'Modifier' and 'Supprimer'.

ID	Avatar	Nom	Prénom	Email	Rôle	Status	Créé le	Modifié le	Actions
3		Preza	Tiss	tiss@test.fr	Admin	En ligne	2023-04-15T13:30:13.000Z	2023-05-30T08:38:58.000Z	Modifier Supprimer
4		Baba	ali	ali@test.fr	Utilisateur	En ligne	2023-04-15T14:08:11.000Z	2023-05-30T09:20:37.000Z	Modifier Supprimer
5		Pre	Mariama	mariama@test.fr	Utilisateur	Hors ligne	2023-04-17T11:48:09.000Z	2023-04-17T12:07:11.000Z	Modifier Supprimer
6		Preza	Tchess	tchesspreza@yahoo.fr	Utilisateur	Hors ligne	2023-04-17T14:02:33.000Z	2023-04-30T10:59:55.000Z	Modifier Supprimer

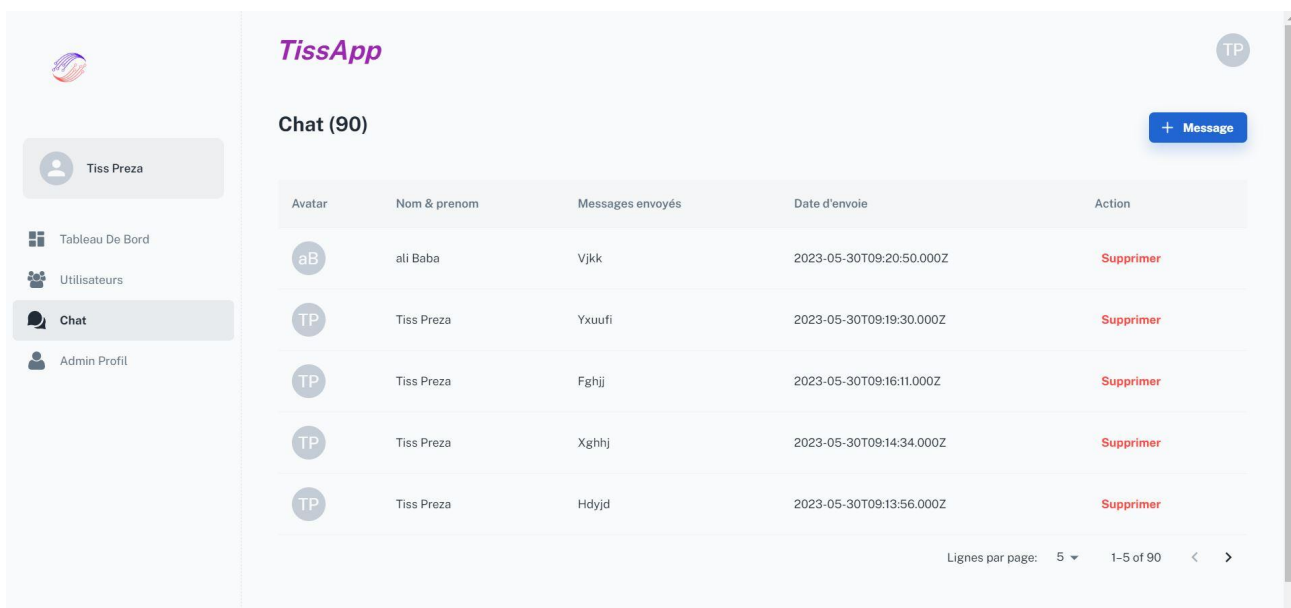
Modal d'inscription d'un nouvel utilisateur :

Sur la page qui affiche les utilisateurs, j'ai intégré un bouton pour ajouter un utilisateur qui ouvre un modal qui facilite l'inscription de nouveaux utilisateurs. En cliquant sur un bouton, les administrateurs peuvent faire apparaître ce modal qui contient un formulaire d'inscription. Ce formulaire comprend des champs pour les informations nécessaires comme le nom, prénom, l'email et le mot de passe. Une fois les informations saisies et validées, un nouvel utilisateur est créé dans la base de données. Cela offre aux administrateurs un moyen facile et rapide d'ajouter des nouveaux utilisateurs à l'application.



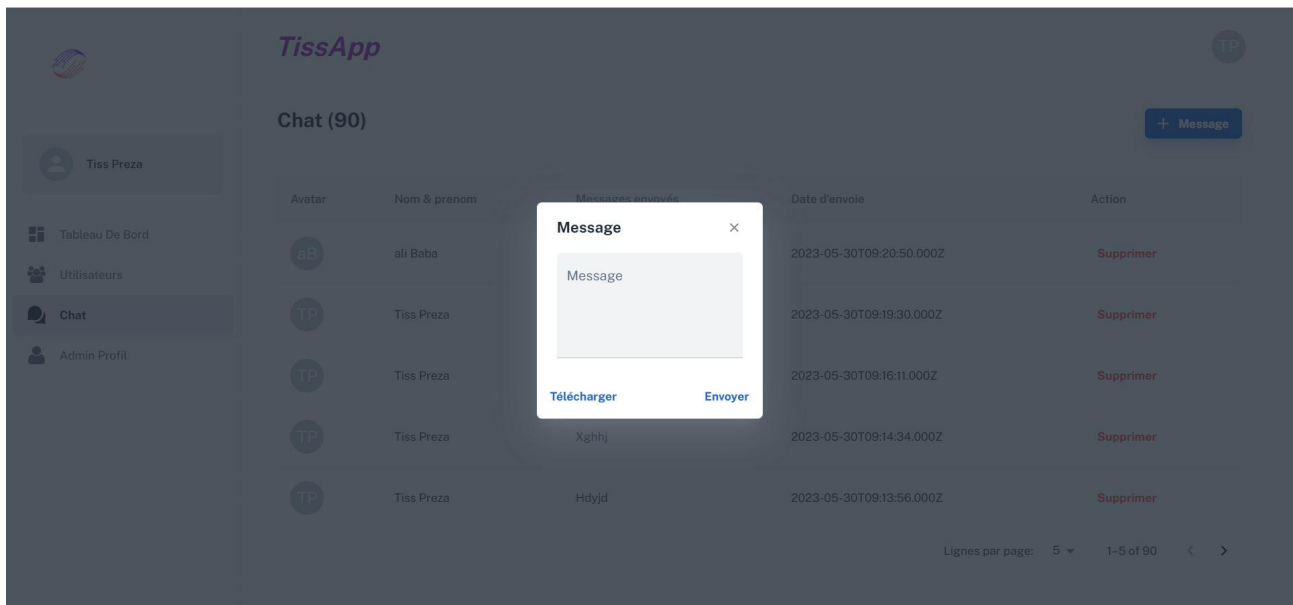
Page de chat :

La page de chat permet aux administrateurs de voir tous les messages envoyés à travers l'application. C'est un outil précieux pour surveiller les conversations et s'assurer qu'elles respectent les règles de l'application. Elle contient aussi une pagination.



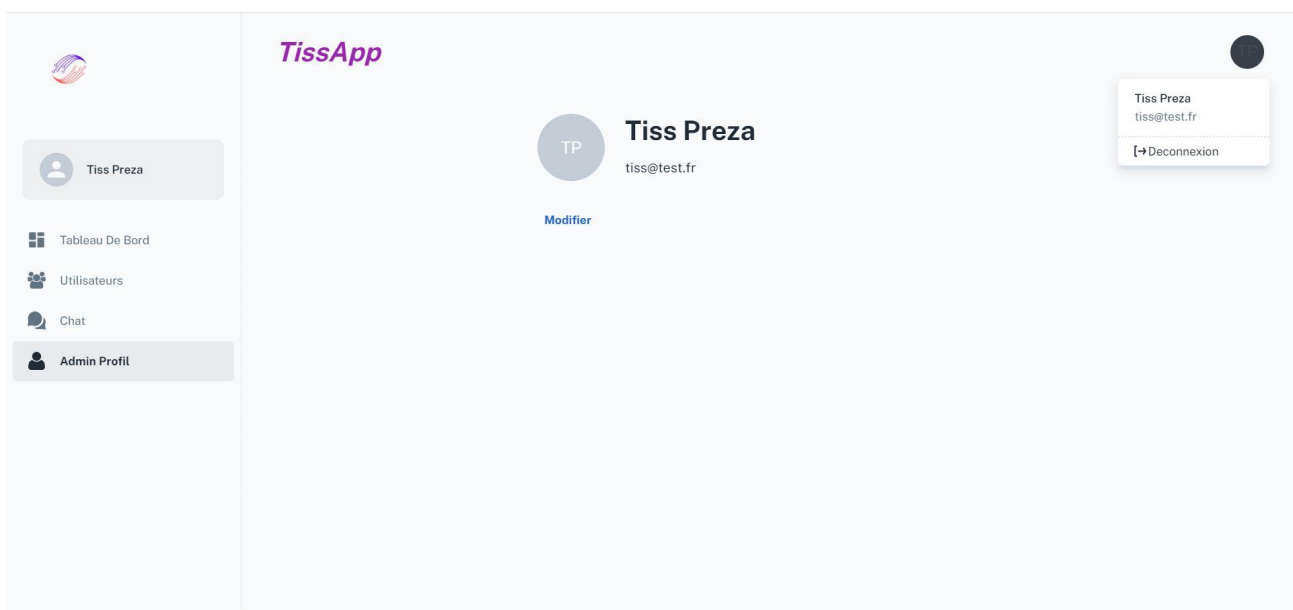
Modal d'envoi de message dans le chat général :

De plus, j'ai mis en place un modal permettant aux administrateurs d'envoyer un message dans le chat général. Ce modal contient un champ où les administrateurs peuvent saisir leur message, un bouton pour télécharger une image. Une fois le message rédigé, ils peuvent cliquer sur un bouton "Envoyer" pour publier le message. Ce message sera alors visible par tous les utilisateurs dans le chat général. Cela offre un moyen efficace pour les administrateurs de communiquer des informations importantes ou des annonces à tous les utilisateurs simultanément.



Page de profil :

La page de profil permet aux administrateurs de gérer leurs informations personnelles. Ils peuvent modifier leurs informations de profil, y compris leur nom et prénom et email



- **Test d'un service d'authentification:**

Dans le cadre du développement de mon application, j'ai également écrit des tests pour valider les mots de passe d'utilisateur. J'ai utilisé Jest, une bibliothèque de tests JavaScript populaire, pour écrire et exécuter les tests.

Le fichier de test pour l'authentification du mot de passe est `authentication.test.js`. Dans ce fichier, j'écris des tests pour chaque fonction dans le contrôleur utilisateur.

Capture d'écran test `authentication.test.js`:

```
TIBABI RECORDS@LAPTOP-5TUN1NVG MINGW64 /d/LA PLATEFORME - Bachelor 2022/Myprojects/app-mobile-chat/back-end (tch-v-3)
$ npx jest
(node:14056) MaxListenersExceededWarning: Possible EventEmitter memory leak detected. 11 open listeners added to [_0x38099f]. Use emitter.setMaxListeners() to increase limit
(Use `node --trace-warnings ...` to show where the warning was created)
PASS tests/authentication.test.js
  ensurePasswordIsStrongEnough
    ✓ devrait lancer une exception si le mot de passe n'est pas assez fort (13 ms)
    ✓ ne devrait pas lancer une exception si le mot de passe est suffisamment fort (1 ms)
  addAuthenticationOn
    ✓ devrait renvoyer un message d'erreur si l'utilisateur n'est pas trouvé (2 ms)
    ✓ devrait renvoyer un message d'erreur si le mot de passe n'est pas correct (1 ms)
    ✓ devrait renvoyer l'utilisateur si l'email et le mot de passe sont corrects (1 ms)

Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        0.778 s
Ran all test suites.
```

- **Test du service de suppression de fichier**

Afin de maintenir l'intégrité des données et de l'espace de stockage dans mon application, j'ai implémenté un service de suppression de fichiers qui supprime les fichiers lorsque cela est nécessaire. Pour m'assurer que ce service fonctionne comme prévu, j'ai écrit des tests pour le service `file-removal.test.js`. Le fichier de test pour le service de suppression de fichiers est `fileRemoval.test.js`. Dans ce fichier, j'écris des tests pour la fonction `deleteFile`. Par exemple, voici le test pour la fonction `deleteFile`:

```
back-end > tests > file-removal.test.js > beforeEach() callback
1  const { deleteFile } = require('../services/file-removal');
2  const fs = require('fs');
3
4  // Mock fs module
5  jest.mock('fs', () => ({
6    unlink: jest.fn(),
7  }));
8
9  beforeEach(() => {
10   fs.unlink.mockReset();
11 });
12
13 describe('deleteFile', () => {
14   it('doit appeler fs.unlink avec le bon chemin', () => {
15     const imageUrl = 'http://example.com/public/image.jpg';
16     deleteFile(imageUrl);
17     expect(fs.unlink).toHaveBeenCalledWith('public/image.jpg', expect.any(Function));
18   });
19
20   it('ne doit pas appeler fs.unlink si imageUrl n\'est pas fourni', () => {
21     deleteFile(null);
22     expect(fs.unlink).not.toHaveBeenCalled();
23   });
24 });
25
```

Dans ces tests, je vérifie si la fonction `fs.unlink` du module `fs` de Node.js est appelée avec le bon chemin lorsqu'un URL d'image est fourni, et si elle n'est pas appelée lorsqu'aucun URL n'est fourni. J'utilise une fonction `mock` pour `fs.unlink` pour simuler l'effacement d'un fichier dans le système de fichiers. Lorsque j'exécute mes tests avec la commande `npm test`, tous les tests passent avec succès, comme le montre la capture d'écran suivante:

Résultat du test du service de suppression de fichiers `file-removal.test`:

```
TIBABI RECORDS@LAPTOP-5TUN1MVG MINGW64 /d/LA PLATEFORME - Bachelor 2022/Myprojects/app-mobile-chat/back-end (tch-v-3)
$ npm test tests/file-removal.test.js

> back-end@1.0.0 test
> jest tests/file-removal.test.js

PASS tests/file-removal.test.js
  deleteFile
    ✓ doit appeler fs.unlink avec le bon chemin (4 ms)
    ✓ ne doit pas appeler fs.unlink si imageUrl n'est pas fourni (1 ms)

Test Suites: 1 passed, 1 total
Tests: 2 passed, 2 total
Snapshots: 0 total
Time: 0.608 s, estimated 1 s
Ran all test suites matching /tests\\file-removal.test.js/i.
```

● Test du controller SignUp

```
1  const UserController = require('../controllers/user');
2  const User = require('../models').sequelize.models.User;
3  const jwt = require('jsonwebtoken');
4
5  jest.mock('../models', () => ({
6    sequelize: {
7      models: {
8        User: {
9          create: jest.fn(),
10        },
11      },
12    },
13  }));
14
15  jest.mock('jsonwebtoken');
16
17  beforeEach(() => {
18    jest.resetAllMocks();
19  });
20
21  describe('UserController', () => {
22    describe('signup', () => {
23      it('should create a new user and return a token', async () => {
24        const mockUser = { firstName: 'John', lastName: 'Doe', email: 'john@doe.com', password: 'password' };
25        const mockReq = { body: mockUser };
26        const mockRes = {
27          status: jest.fn(() => mockRes),
28          json: jest.fn(),
29        };
30        const mockNext = jest.fn();
31        const mockToken = 'mockToken';
32
33        User.create.mockResolvedValue({ ...mockUser, id: 1 });
34        jwt.sign.mockReturnValue(mockToken);
35
36        await UserController.signup(mockReq, mockRes, mockNext);
```

Dans cet exemple, on utilise `jest.mock` pour mocker les modules `User` et `jwt`. Ensuite, pour chaque test, on crée des mocks pour la requête, la réponse et la fonction `next`. Ces mocks sont utilisés pour simuler le comportement de `Express.js`. On utilise également `User.create.mockResolvedValue` et `User.create.mockRejectedValue` pour simuler le succès et l'échec de la création d'un utilisateur.

Un **"mock"** est une technique utilisée en programmation pour simuler le comportement d'un objet réel dans un contexte de test.

Capture d'écran du résultat du test UserSignUp.test.js

```
TIBABI RECORDS@LAPTOP-5TUN1NVG MINGW64 /d/LA PLATEFORME - Bachelor 2022/Myprojects/app-mobile-chat/back-end (tch-v-3)
$ npm test tests/User.SignUp.test.js

> back-end@1.0.0 test
> jest tests/User.SignUp.test.js

PASS tests/User.SignUp.test.js
  UserController
    signup
      ✓ should create a new user and return a token (4 ms)
      ✓ should return 401 if creating a user fails (2 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.022 s
Ran all test suites matching /tests\\User.SignUp.test.js/i.
```