

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de la Manouba

Institut supérieur des Arts Multimédias



Projet de fin d'année

Développement d'une application web de gestion de tâches collaborative

Réalisé par

Aya Ghattas

Ines Khelil

Encadré par

Dr Sana BEN ABDALLAH– ISAMM

Mme Amel HAMOUDI

Année universitaire 2023-2024

Remerciements

Nous souhaitons exprimer notre profonde gratitude envers toutes les personnes qui ont, de près ou de loin, contribué à la réalisation de ce projet de fin d'année et à l'élaboration de ce rapport. Nos remerciements les plus sincères vont à l'ensemble de nos enseignants pour la qualité de leur enseignement, leurs précieux conseils, et l'intérêt constant qu'ils portent à chacun de leurs étudiants.

Nous tenons à manifester notre reconnaissance la plus profonde à **Dr Sana Ben Abdallah**, notre encadrante à l'ISAMM. Son aide précieuse, sa patience, ses conseils éclairés et l'attention qu'elle nous a octroyé ont été inestimables tout au long de ce projet. Nous sommes fières et reconnaissantes qu'elle ait accepté de nous encadrer pour ce projet de fin d'année.

Nos sincères remerciements vont également à **Madame Amel Hamoudi** pour son soutien indéfectible durant toute la réalisation de ce projet. Nous apprécions profondément le temps et les efforts qu'elle y a consacrés, et nous lui en sommes extrêmement reconnaissantes.

Enfin, nous adressons nos plus vives reconnaissances aux membres du jury pour avoir accepté d'évaluer notre travail.

Table des matières

Introduction générale.....	1
Chapitre 1 : Contexte et objectif du projet.....	2
Introduction.....	2
I.1 Contexte général	2
I.1.1 Cadre du projet	2
I.1.2 Objectifs du projet	2
I.2 Etude de l'existant.....	3
I.2.1 Application 1 : Trello.....	3
I.2.2 Application 2 : Jira	4
I.2.3 Application 3 : Microsoft To Do	6
I.2.4 Tableau comparatif.....	7
I.3 Choix de la méthodologie de travail et approche méthodologique.....	8
Conclusion.....	9
Chapitre 2 : Analyse et spécification des besoins.....	10
Introduction.....	10
I. Spécification des besoins fonctionnels	10
I.1 Spécification informelle.....	10
I.1.1 Identification des acteurs.....	10
I.1.2 Identification des besoins fonctionnels	10
I.2 Spécification semi-formelle.....	12
I.2.1 Diagramme de cas d'utilisation général.....	13
I.2.2 Diagramme de classe général.....	14
I.2.3 Backlog du produit.....	14
I.2.4 Pilotage du projet avec Agile Scrum.....	16
II. Spécification des besoins non fonctionnels	17
Conclusion :.....	17
Chapitre 3 : Initialisation du projet et réalisation	18
Introduction.....	18
I.1 Environnement matériel et logiciel	18
I.2 Identité de l'application	20
I.2.1 Logo	20

I.2.2	Charte graphique.....	20
I.3	Dockerisation.....	20
I.4	Interfaces.....	22
I.5	Test logiciel.....	28
I.5.1	Test Backend	28
I.5.2	Test Frontend	29
	Conclusion.....	30
	Conclusion générale et perspectives.....	31
	Bibliographie.....	32

Liste des figures

Figure 1 : Interface de la plateforme trello	3
Figure 2 : Interface de la plateforme jira	5
Figure 3 : Interface de la plateforme Microsoft to do	6
Figure 4 : Demarche Devops.....	9
Figure 5 : Diagramme de cas d'utilisation général	13
Figure 6 : Diagramme de classe general	14
Figure 7 : Logo de tâchety	20
Figure 8 : Fichier docker compose	21
Figure 9 : Execution du fichier docker compose	22
Figure 10 : Dashboard traefik	22
Figure 11 : Interface d'inscription.....	23
Figure 12 : Interface d'authentification.....	23
Figure 13 : Interface du tableau de board.....	24
Figure 14 : Interface des projets	24
Figure 15 : Interface d'ajout d'un projet	25
Figure 16 : Interface de modification d'un projet.....	25
Figure 17 : Interface d'ajout de membres et de colonnes.....	26
Figure 18 : Interface d'ajoute d'une tache.....	27
Figure 19 : Interface de modification d'une tache	27
Figure 20 : Notifications	28
Figure 21 : Postman : Test de récupération de la liste des projets	28
Figure 22 : Postman : Test de création d'un projet	29
Figure 23 : Cypress : validation du test de l'interface d'inscription	29
Figure 24 : Cypress : validation du test de l'interface d'authentification.....	30
Figure 25 : Cypress : Validation du test d'ajout d'un projet.....	30

Liste des tableaux

Tableau 1 : Tableau comparatif	8
Tableau 2 : Backlog de l'application	15
Tableau 3 : Décomposition du projet en sprint	17
Tableau 4 : Outils logiciel.....	18
Tableau 5 : Outils matériels	19
Tableau 6 : Principales couleurs.....	20

Introduction générale

L'évolution rapide des technologies de l'information et de la communication a radicalement transformé notre manière de gérer nos activités quotidiennes, tant sur le plan personnel que professionnel.

Dans le contexte actuel, où la collaboration et la gestion efficace du temps sont des facteurs déterminants pour la réussite des projets, les outils de gestion de tâches collaboratives se révèlent indispensables. Ces outils permettent aux équipes de coordonner leurs efforts, de suivre l'avancement des projets et d'atteindre leurs objectifs de manière plus efficace et structurée.

Ce rapport se propose d'explorer en détail le paysage des outils de gestion de tâches collaboratives. Nous en analyserons les avantages et les inconvénients, ainsi que leur pertinence dans divers contextes d'utilisation. Cette analyse nous permettra de définir nos besoins fonctionnels et non fonctionnels, étape essentielle avant de procéder au développement de notre propre application de gestion de tâches collaborative. Nous adopterons une méthodologie de travail et une approche pour garantir que la solution finale réponde pleinement aux attentes et aux exigences des utilisateurs.

Chapitre 1 : Contexte et objectif du projet

Introduction

Dans ce chapitre d'introduction, nous allons commencer par contextualiser notre projet. Ensuite, nous allons nous focaliser sur l'évaluation des solutions existantes et procéder à une analyse ainsi qu'une étude comparative. Enfin, nous allons présenter notre choix de méthodologie et approche pour le développement de notre application web.

I.1 Contexte général

I.1.1 Cadre du projet

Le travail présenté s'inscrit dans le cadre du projet de fin d'année pour la deuxième année cycle ingénieur à l'Institut Supérieur des Arts Multimédia de la Manouba.

I.1.2 Objectifs du projet

Dans un environnement où la gestion des tâches en équipe présente des défis significatifs (coordination des membres, suivi de l'avancement, etc.), le développement d'une application web collaborative de gestion des tâches offre des solutions efficaces et une plus grande flexibilité pour les utilisateurs. Notre application vise à permettre aux équipes de travailler ensemble de manière plus flexible, peu importe leur emplacement, en organisant leurs tâches de manière optimale. Cependant, même avec des outils collaboratifs disponibles, il peut être difficile de trouver une plateforme intuitive et complète qui répond à tous les besoins de gestion de projets. De plus, la complexité de certains outils peut décourager les utilisateurs moins expérimentés. Dans notre application, nous visons à résoudre ces problèmes en :

- Proposant une interface utilisateur simple et intuitive pour faciliter l'adoption par tous les membres de l'équipe.
- Offrant des fonctionnalités complètes pour la gestion des tâches, y compris des rappels, des échéances, et des suivis de progression.

I.2 Etude de l'existant

Afin de faciliter la gestion des tâches collaboratives et de la rendre accessible à tous, des applications ont été développées pour offrir des outils modernes et conviviaux. Ces applications visent à simplifier la coordination des équipes, le suivi de l'avancement des projets et la communication entre les membres, quel que soit leur emplacement géographique. Dans cette section, nous présentons quelques exemples d'outils de gestion des tâches collaboratifs pour mieux définir les fonctionnalités à intégrer dans notre application.

I.2.1 Application 1 : Trello

Trello est un outil de gestion de travail dans lequel les équipes peuvent créer des plans, collaborer sur des projets, organiser les flux de travail et suivre l'avancement de façon visuelle, productive et enrichissante. [1]

La figure 1 montre l'interface de la plateforme Trello.

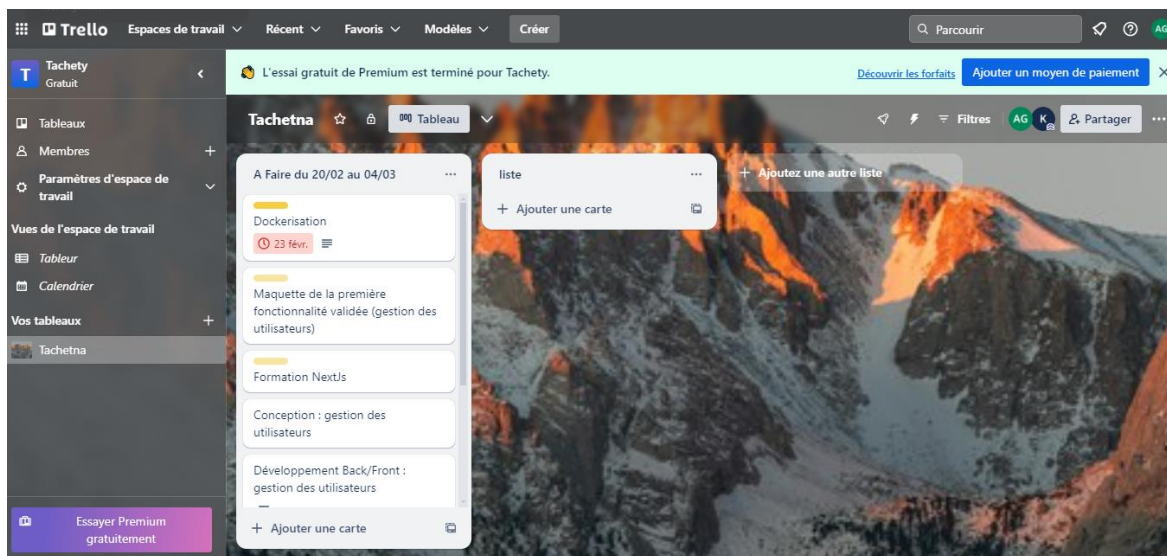


FIGURE 1 : INTERFACE DE LA PLATEFORME TRELLO

Fonctionnalités :

- **Tableaux, Listes et Cartes** : Structure de base pour organiser et gérer les tâches.
- **Power-Ups** : Intégrations avec des applications tierces pour des fonctionnalités supplémentaires.

- **Automatisation avec Butler** : Automatisation des tâches répétitives avec des règles et commandes.

Avantages :

- **Simplicité d'utilisation** : Interface intuitive et visuelle, facile à comprendre.
- **Collaboratif** : Permet une communication et une collaboration en temps réel entre les membres de l'équipe, avec des fonctionnalités de commentaires et d'attribution de tâches.
- **Accessibilité** : Disponible sur plusieurs plateformes (web, mobile) et synchronisé en temps réel, permettant un accès facile aux informations depuis n'importe où.

Inconvénients :

- **Limité pour grands projets** : Moins efficace pour les projets très complexes.
- **Dépendance aux Power-Ups** : Certaines fonctionnalités nécessitent des modules payants.
- **Absence de système de notifications.**

I.2.2 Application 2 : Jira

Jira est une plateforme multifonction qui vise à faciliter la gestion de projet en aidant à suivre les tâches, identifier les blocages et partager des informations entre les membres d'une équipe. [2]

La figure 2 montre l'interface de la plateforme Jira.

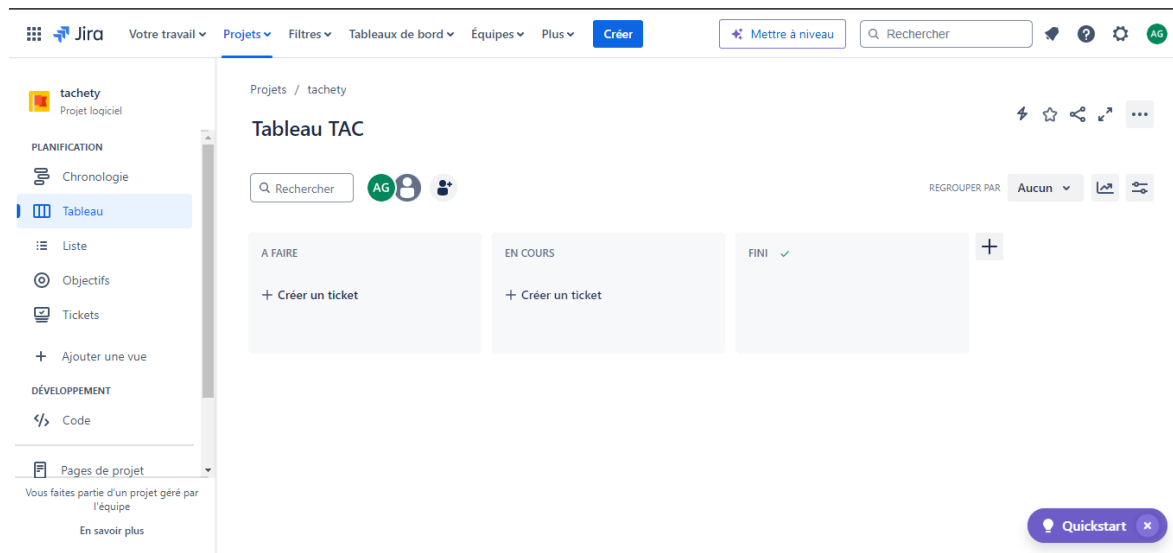


FIGURE 2 : INTERFACE DE LA PLATEFORME JIRA

Fonctionnalités :

- **Tableaux Scrum et Kanban** : Pour la gestion agile des projets.
- **Rapports et tableaux de bord** : Visualisation des performances et des métriques.
- **Intégrations** : Connectivité avec de nombreuses autres applications et outils.

Avantages :

- **Personnalisable** : S'adapte aux besoins spécifiques des équipes.
- **Gestion Agile** : Parfait pour les méthodologies Scrum et Kanban.
- **Rapports détaillés** : Outils puissants de reporting et d'analyse.

Inconvénients :

- **Complexité** : Peut être difficile à maîtriser pour les nouveaux utilisateurs.
- **Coût élevé** : Les frais peuvent s'accumuler avec les utilisateurs et les plugins.
- **Configuration initiale** : Mise en place et personnalisation peuvent être chronophages.
- **Interface encombrée** : Peut sembler surchargée pour certains utilisateurs.

I.2.3 Application 3 : Microsoft To Do

Microsoft To Do est une application de gestion des tâches développée par Microsoft. Elle permet aux utilisateurs de créer, organiser et suivre des listes de tâches quotidiennes et des projets à court terme, avec des fonctionnalités de synchronisation sur plusieurs appareils.

La figure 3 ci-dessous montre l'interface de la plateforme Microsoft To Do.

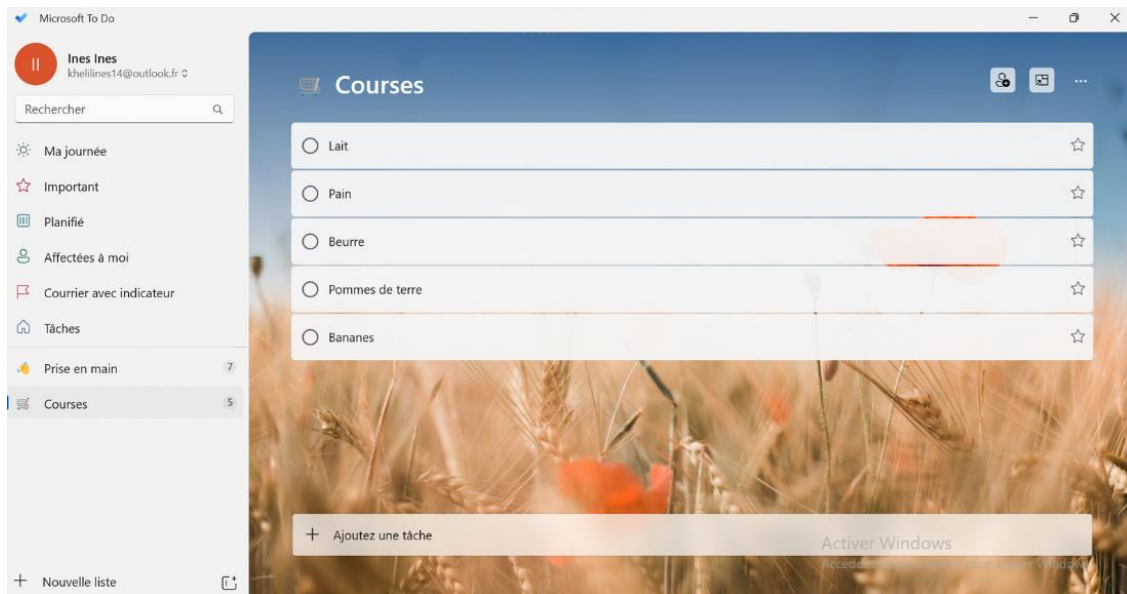


FIGURE 3 : INTERFACE DE LA PLATEFORME MICROSOFT TO DO

Fonctionnalités :

- **Listes de tâches** : Créer et organiser des listes de tâches.
- **Rappels et échéances** : Ajouter des rappels et des dates d'échéance pour chaque tâche.
- **Synchronisation multiplateforme** : Accès aux tâches sur différents appareils.
- **Partage de listes** : Partager des listes de tâches avec d'autres utilisateurs.

Avantages :

- **Intégration avec Microsoft 365** : Synchronisation avec Outlook et autres applications Microsoft.
- **Interface simple** : Facile à utiliser et à naviguer.

- **Gratuit** : Disponible gratuitement.
- **Synchronisation automatique** : Mise à jour en temps réel sur tous les appareils.

Inconvénients :

- **Fonctionnalités limitées** : Moins de fonctionnalités avancées comparé à d'autres outils.
- **Personnalisation restreinte** : Options de personnalisation limitées.
- **Pas de support pour projets complexes.**

I.2.4 Tableau comparatif

Grace à une analyse des fonctionnalités de Jira, Trello et Microsoft To Do, nous avons pu faire une comparaison entre ces différentes applications présentée par le Tableau 1 et identifier les principales fonctionnalités qui doivent exister dans notre application à savoir : La gestion des projets, gestion des tâches, gestion des notifications etc...

TABLEAU 1 : TABLEAU COMPARATIF

Tableau comparatif			
Outils Fonctionnalités	Trello	Jira	Microsoft ToDo
Gestion de projet	✓	✓	✓
Gestion des tâches	✓	✓	✓
Tableaux Kanban	✓	✓	✗
Gestion des utilisateurs	✓ 	✓	✓ 
Personnalisation des tableaux	✓	✓	✗ 
Prix	Gratuit avec des options payantes	Gratuit avec des options payantes	Gratuit avec des options payantes
Intégration Git	✓ 	✓	✗
Rapports et analytics	✓ Avec Power-ups	✗	✗
Collaboration en équipe	✓	✓	✓  Microsoft 365
Notifications	✓ 	✓	✓

I.3 Choix de la méthodologie de travail et approche méthodologique

Notre choix s'est porté sur le framework Scrum utilisé dans la gestion de projet Agile aidant les équipes à mieux structurer et gérer leur travail.

En complément, nous adopterons une approche DevOps. Voir la démarche dans la figure 4.

Cette méthode nous permettra d'améliorer l'efficacité grâce à l'automatisation des processus, et garantir des livraisons continues et fiables de notre application.

En combinant Scrum et DevOps, nous pourrions mieux gérer notre temps et assurer une qualité élevée à chaque étape du développement.

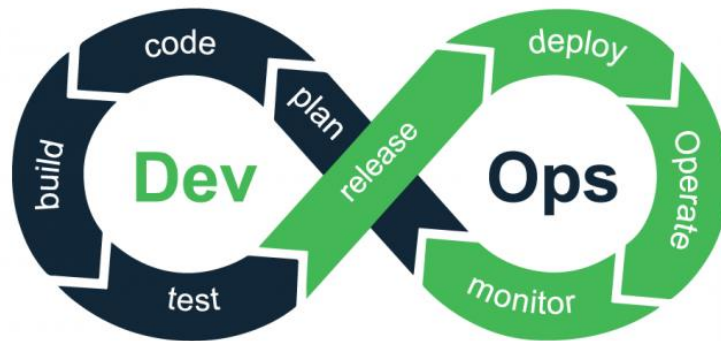


FIGURE 4 : DEMARCHE DEVOPS

Conclusion

Dans ce chapitre, nous avons introduit le contexte global du projet et examiné en détail les solutions existantes ainsi que les critiques qui leur sont associées et étalé la méthodologie que nous allons suivre tout au long du projet. Le prochain chapitre sera consacré à l'analyse et à la spécification des besoins.

Chapitre 2 : Analyse et spécification des besoins

Introduction

Cette phase représente la première phase du cycle de développement d'un logiciel. Dans ce chapitre, nous commençons en premier lieu par une spécification des besoins fonctionnels à travers l'introduction des acteurs et les diagrammes de cas d'utilisation relatifs à ces acteurs, passant ensuite à la spécification des besoins non fonctionnels.

I. Spécification des besoins fonctionnels

I.1 Spécification informelle

Dans cette partie nous allons identifier les acteurs et leurs rôles ainsi que les besoins fonctionnels et non fonctionnels qu'on cherche à adapter dans notre application.

I.1.1 Identification des acteurs

Dans cette partie nous allons identifier les acteurs du système :

- **User** : C'est tout type d'utilisateurs de l'applications.
- **Manager** : C'est celui qui a créé le projet.
- **Member** : C'est un membre de l'équipe.

Maintenant nous allons préciser les besoins fonctionnels et non fonctionnels de ces acteurs.

I.1.2 Identification des besoins fonctionnels

L'application doit permettre à l'**Utilisateur** de :

- **S'inscrire** : l'utilisateur peut créer un compte avec ses informations personnelles.
- **S'authentifier** : L'utilisateur saisit son login et son mot de passe pour accéder à son espace personnel.

- **Gérer son profil personnel** : L'utilisateur peut accéder à ses données personnelles et les modifier, ajouter une photo de profil.
- **Consulter ses projets** : L'utilisateur peut consulter les projets dans lesquels il est membre ou manager.
- **Chercher un projet** : L'utilisateur peut chercher un projet selon son nom.
- **Ajouter un projet** : L'utilisateur peut créer un nouveau projet et en devient automatiquement le manager.
- **Consulter calendrier** : L'utilisateur peut consulter le calendrier pour voir les tâches qui lui sont assignées, triées par date.
- **Recevoir des notifications** : L'utilisateur peut recevoir des notifications quand il est ajouté à un projet ou quand une tâche lui a été assignée.
- **Consulter les statistiques de ses projets** : l'utilisateur peut consulter les statistiques tels que le nombre total de projets et tâches qui lui sont assignées etc...

L'application doit permettre au **Manager** de :

- **Gérer les colonnes** : Seul le manager du projet peut ajouter un supprimer les colonnes.
- **Supprimer des membres** : Le manager peut supprimer des membres du projet.
- **Supprimer un projet** : Seul le manager du projet est capable de le supprimer
- **Modifier un projet** : Seul le manager du projet qui peut modifier les informations de ce dernier.

L'application doit permettre au **Membre** de :

- **Gérer les tâches** :
 - **Ajouter, modifier, supprimer tâche** : le membre peut gérer uniquement les tâches dont il est responsable.
 - **Basculer tâche** : le membre peut faire le drag and drop pour modifier l'état d'une tâche.
- **Importer un fichier** : Le membre peut ajouter un fichier joint aux tâches dont il est responsable.

- **Ajouter un membre** : Le membre peut inviter un collègue à rejoindre le projet.

I.2 Spécification semi-formelle

La figure 5 illustre le diagramme de cas d'utilisation global de notre application.

I.2.1 Diagramme de cas d'utilisation général

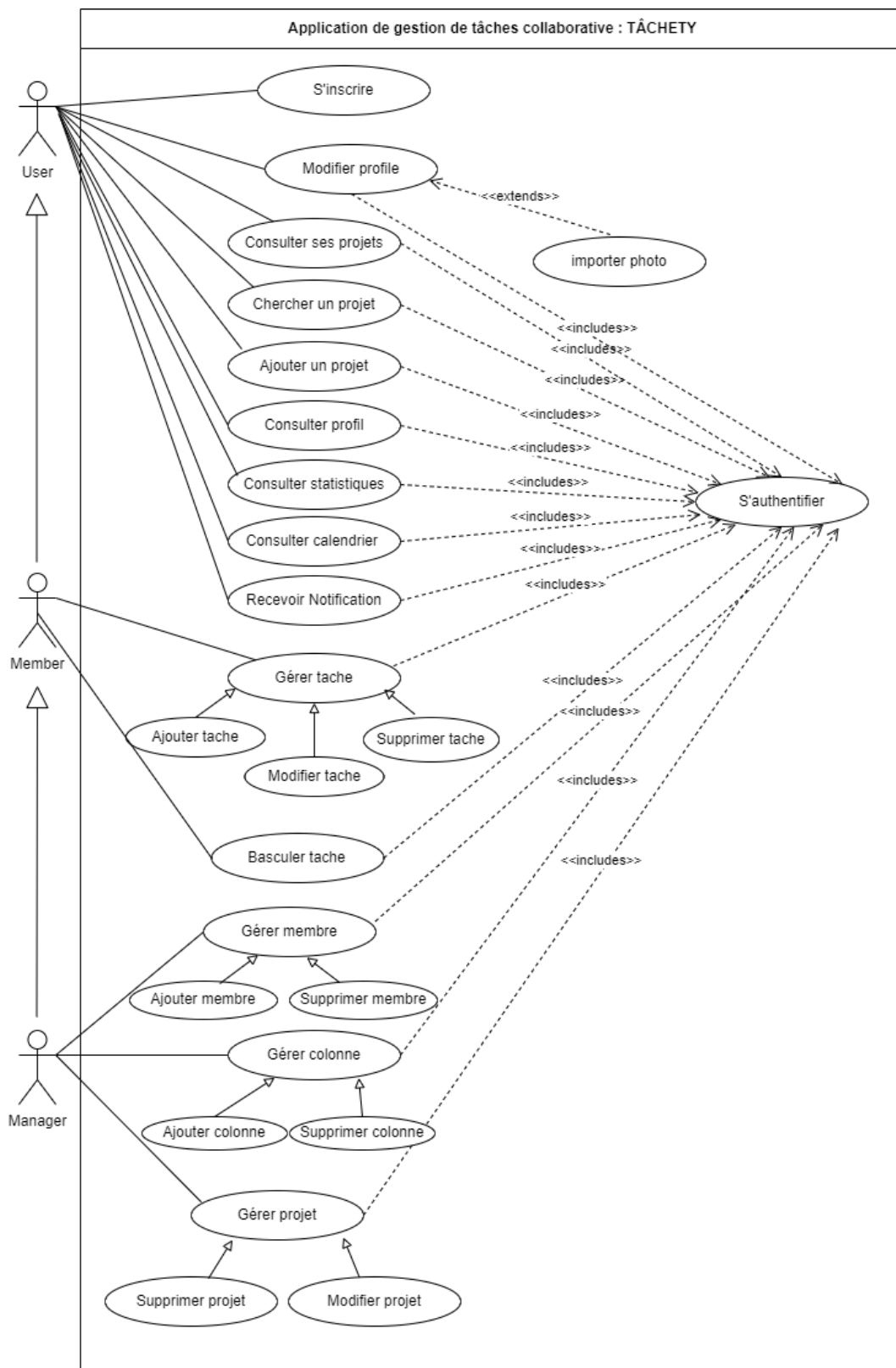


FIGURE 5 : DIAGRAMME DE CAS D'UTILISATION GENERAL

I.2.2 Diagramme de classe général

La figure 6 illustre le diagramme classe général de notre application.

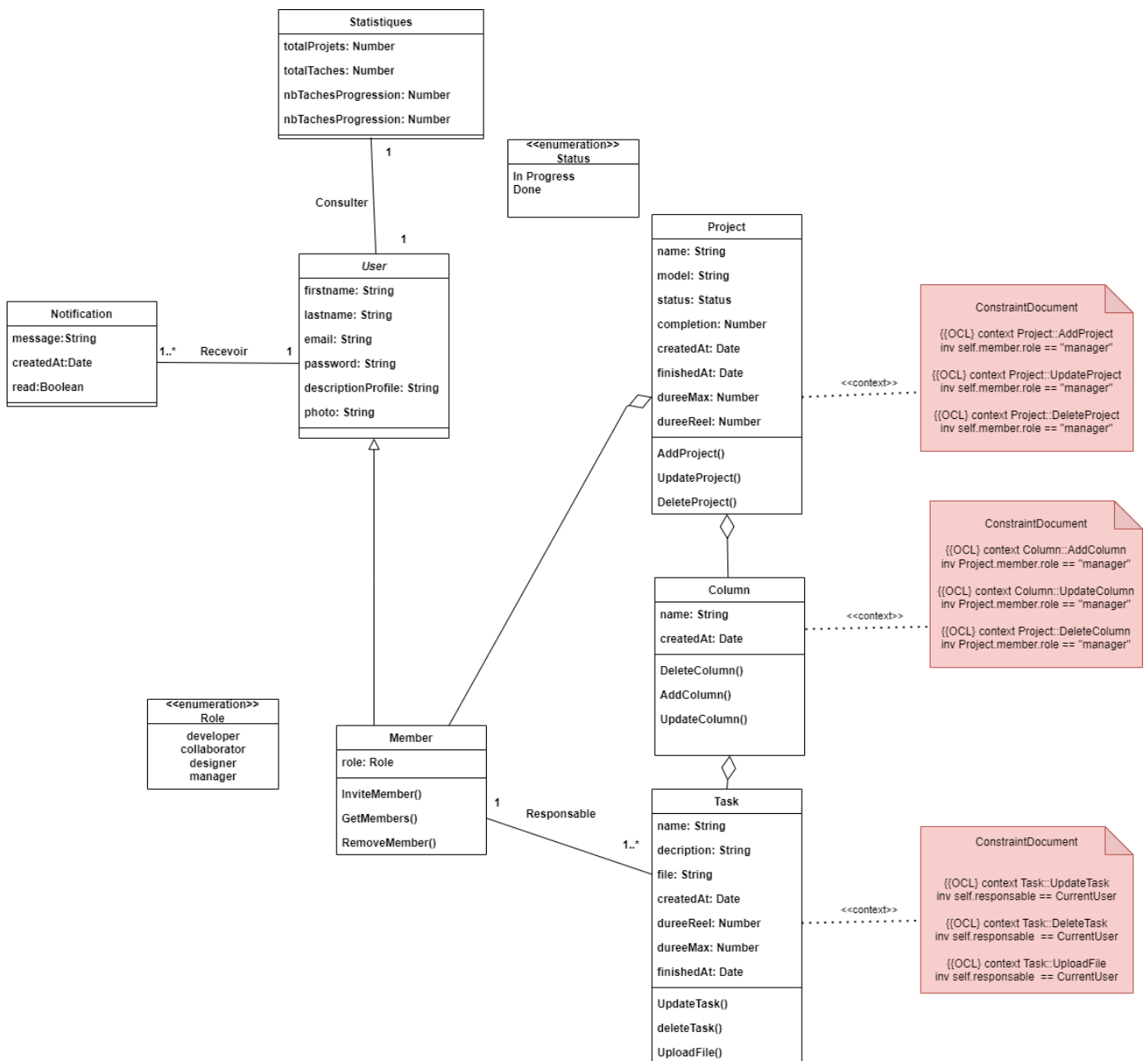


FIGURE 6 : DIAGRAMME DE CLASSE GENERAL

I.2.3 Backlog du produit

Un backlog produit est une liste hiérarchisée de tâches destinées à l'équipe de développement. Il est créé à partir de la feuille de route produit et de ses exigences.

[4]

Notre Backlog, présenté dans le tableau 2, comprend les champs suivants :

Rang : C'est un identifiant unique de chaque user story.

Thème : Représente le contexte dans lequel existe l'user story.

User story : Décrit un besoin fonctionnel désiré par le client.

Priorité : Représenté par la méthode de priorisation "MoSCoW" dans laquelle chaque user story est priorisée comme l'un des suivants :

- **Must** : La fonctionnalité doit être implémentée.
- **Should** : La fonctionnalité devrait être implémentée dans la mesure du possible.
- **Could** : La fonctionnalité pourrait être implémentée si elle n'a pas d'impact sur les autres tâches.
- **Won't** : La fonctionnalité ne sera pas implémentée cette fois mais sera faite plus tard.

La priorité d'une story peut être changée en cours de réalisation du projet.

TABLEAU 2 : BACKLOG DE L'APPLICATION

Rang	Thème	Element du backlog	Priorité
1	Dockerisation de l'application	Dockeriser l'application dans un environnement isolé et portable.	Must
2	Gestion des utilisateurs	Créer un compte, m'authentifier et gérer les membres de mon équipe si je suis un manager.	Must
3	Gestion des projets	Créer, modifier et supprimer un espace de travail et je peux attribuer des tâches à des membres du projet.	Must
4	Gestion des tâches	Ajouter, modifier, supprimer et déplacer des tâches dans différentes colonnes, je peux attribuer des dates d'échéance, avec possibilité de joindre des fichiers.	Must
5	Gestion des colonnes	Créer, modifier et supprimer les colonnes du projet.	Must

6	Fonction de recherche	Chercher facilement des projets ou des membres.	Must
7	Visualisation statistique et	Générer des rapports et des graphiques de progression du projet et de mon avancement.	Must
8	Synchronization en temps réel	Voir la mise à jour des données sans avoir recours au rafraichissement manuel.	Must
9	Gestion des accès	Gérer les accès des utilisateurs	Must
10	Gestion des membres	Ajouter, supprimer et attribuer des tâches aux membres du projet.	Must
11	Rappels et notifications	Recevoir des rappels automatiques pour l'ajout à une tâche ou un projet.	Should
12	Calendrier des tâches	Consulter le calendrier pour visualiser mes tâches.	Should
13	Gestion du profil utilisateur	Modifier les informations de mon profil (nom, prénom, photo de profil ...).	Should
14	Intégration d'autres outils	Utiliser des outils tiers tels que Google Calendar, Teams, etc..	Won't
15	Messagerie	Envoyer et recevoir des messages des différents utilisateurs de l'application.	Won't

I.2.4 Pilotage du projet avec Agile Scrum

Pour notre projet nous avons choisi de développer trois releases illustrées par le tableau 3.

TABLEAU 3 : DECOMPOSITION DU PROJET EN SPRINT

Release	Sprint	Tâche
Release 1	Sprint 1	- Dockerisation de l'application - Gestion des utilisateurs
	Sprint 2	- Gestion des projets - Gestion des tâches
	Sprint 3	- Gestion des colonnes - Gestion des membres
Release 2	Sprint 4	- Visualisation et statistique - Gestion du profil utilisateur
	Sprint 5	- Rappels et notifications - Gestion des accès
	Sprint 6	- Calendrier des tâches - Fonction de recherche

II. Spécification des besoins non fonctionnels

Les besoins non fonctionnels sont des exigences qui spécifient les critères de qualité d'un système.

- **Sécurité** : Attribution des rôles ainsi que des informations d'accès à chaque groupe d'utilisateurs pour protéger l'accès aux différents espaces.
- **Ergonomie** : La convivialité, la compréhensibilité des interfaces graphiques, le choix des couleurs etc...
- **Disponibilité** : l'application doit être accessible à tout moment pendant tous les jours de la semaine.

Conclusion :

Ce chapitre a détaillé l'analyse et la spécification des besoins de notre application, en identifiant les acteurs, leurs rôles, ainsi que les besoins fonctionnels et non fonctionnels. Nous avons également planifié le projet en utilisant la méthodologie Agile Scrum, en décomposant le développement en sprints pour une livraison progressive et structurée des fonctionnalités.

Chapitre 3 : Initialisation du projet et réalisation

Introduction

Ce chapitre est consacré à l'initialisation du projet, où nous commençons par présenter les Frameworks de développement utilisé. Ensuite, nous exposons les outils de conception et de développement qui seront employés pour la réalisation de l'application web. Enfin, nous illustrons ces éléments par quelques captures d'écran de l'avancement de l'application ainsi que les tests réalisés.

I.1 Environnement matériel et logiciel

Dans la partie suivante nous présentons les ressources matériels et logiciels (de développement, et de conception) que nous avons utilisé pour le développement de l'application web, dans les deux tableaux illustrés ci-dessous.

Logiciels et Frameworks :

TABLEAU 4 : OUTILS LOGICIEL

Outil	Description
	C'est une plateforme open-source de développement javascript. Il fonctionne comme un environnement d'exécution qui peut être utilisé pour écrire des services API (Application Programming Interface). [5]
	Postman est un environnement de développement des API. Il facilite le test des API et des services web.
	MongoDB Atlas est la seule plateforme de données applicatives multi-cloud qui accélère et simplifie la création avec des données. [6]
	C'est un framework de développement web qui s'appuie sur la bibliothèque React.

	Cypress est un framework d'automatisation de tests basé sur du JavaScript. C'est un outil open-source qui permet d'établir les tests bout à bout (end-to-end).
	GitHub est un service d'hébergement Open-Source, permettant aux programmeurs et aux développeurs de partager le code informatique de leurs projets afin de travailler dessus de façon collaborative. [7]
	Le Sous-système Windows pour Linux (WSL) est une fonctionnalité du système d'exploitation Windows qui permet d'exécuter un système de fichiers Linux, ainsi que des outils en ligne de commande et des applications Linux, directement sur Windows. [8]
	Docker est une technologie de conteneurisation qui permet la création et l'utilisation de conteneurs Linux. [9]
	Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.
	Un logiciel de dessin graphique multiplateforme gratuit et open source développée en HTML5 et JavaScript.
	C'est un logiciel de prototypage d'interface et de conception de sites web ou d'applications mobile.

Outils matériels :

TABLEAU 5 : OUTILS MATERIELS

Fournisseur	Processeur	RAM	Disque Dur	Système d'exploitation
Lenovo	Intel(R) Core (TM) i7-6600U CPU @ 2.60GHz 2.81 GHz	8 GO	SSD	Windows 10 Professional 64Bits
Lenovo	Intel® Core™ i7-1165G7 @ 2.80GHz 2.80GHz	8 GO	SSD	Windows 11 Professional 64Bits

I.2 Identité de l'application

Dans cette partie nous allons présenter l'identité visuelle de notre application.

I.2.1 Logo

La figure 7 du logo, illustre une pile de cartes, symbolise l'organisation et la gestion des tâches au sein de l'équipe. Chaque carte, représentative d'une tâche, est attribuée à un responsable, garantissant une collaboration fluide et une meilleure productivité.

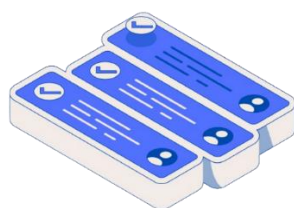





FIGURE 7 : LOGO DE TACHETY

I.2.2 Charte graphique

Le Tableau 6 présente les principales couleurs et leur présentation en code hexadécimal (HEX) et dans la présentation du système Rouge Vert Bleu (RGB).

TABLEAU 6 : PRINCIPALES COULEURS

Couleur	Hexadécimale	RGB
	#1f2941	rgb(31,41,65)
	#e9edf7	rgb(233,237,247)
	#1d4ed8	rgba(29,78,216,255)

Pour la typographie nous utilisons la police d'écriture **Satoshi**.

I.3 Dockerisation

Pour la partie dockerisation de l'application, nous avons utilisé le docker compose ci-dessous, composé de 4 services : server, client, un reverse proxy traefik et une base de données mongoDB.

```

1  version: '3.9'
2  services:
3    server:
4      image: node:alpine
5      working_dir: /app/server
6      command: sh -c "npm install && npm uninstall bcrypt && npm install bcrypt && npm run dev"
7      volumes:
8        - ./server:/app/server
9      ports:
10       - "5000:5000"
11      labels:
12       - "traefik.http.routers.server.rule=Host('server.localhost')"
13      depends_on:
14       - mongodb
15       - traefik
16
17    client:
18      image: node:alpine
19      working_dir: /app/client
20      command: sh -c "npm install && npm run dev"
21      volumes:
22       - ./client:/app/client
23      ports:
24       - "3000:3000"
25      labels:
26       - "traefik.http.routers.client.rule=Host('client.localhost')"
27      depends_on:
28       - server
29       - traefik
30
31    traefik:
32      image: traefik:v3.0.0-rc3
33      command: --api.insecure=true --providers.docker
34      ports:
35       - "8080:8080"
36       - "80:80"
37      volumes:
38       - /var/run/docker.sock:/var/run/docker.sock
39
40
41    mongodb:
42      image: mongo:latest
43      volumes:
44       - mongodb:/data/db
45      ports:
46       - "27017:27017"
47      labels:
48       - "traefik.http.routers.mongodb.rule=Host('mongodb.localhost')"
49
50  volumes:
51    mongodb: {}
52    traefik: {}

```

FIGURE 8 : FICHIER DOCKER COMPOSE

La commande “sudo docker-compose up” pour exécuter le fichier est représenté par la figure 9.

```
[+] Running 4/0
✓Container gestion_tache_collab-mongodb-1 Created
✓Container gestion_tache_collab-traefik-1 Created
✓Container gestion_tache_collab-server-1 Created
✓Container gestion_tache_collab-client-1 Created
Attaching to client-1, mongodb-1, server-1, traefik-1
```

FIGURE 9 : EXECUTION DU FICHIER DOCKER COMPOSE

La figure 10 est un aperçu du dashboard de treafik de notre application :

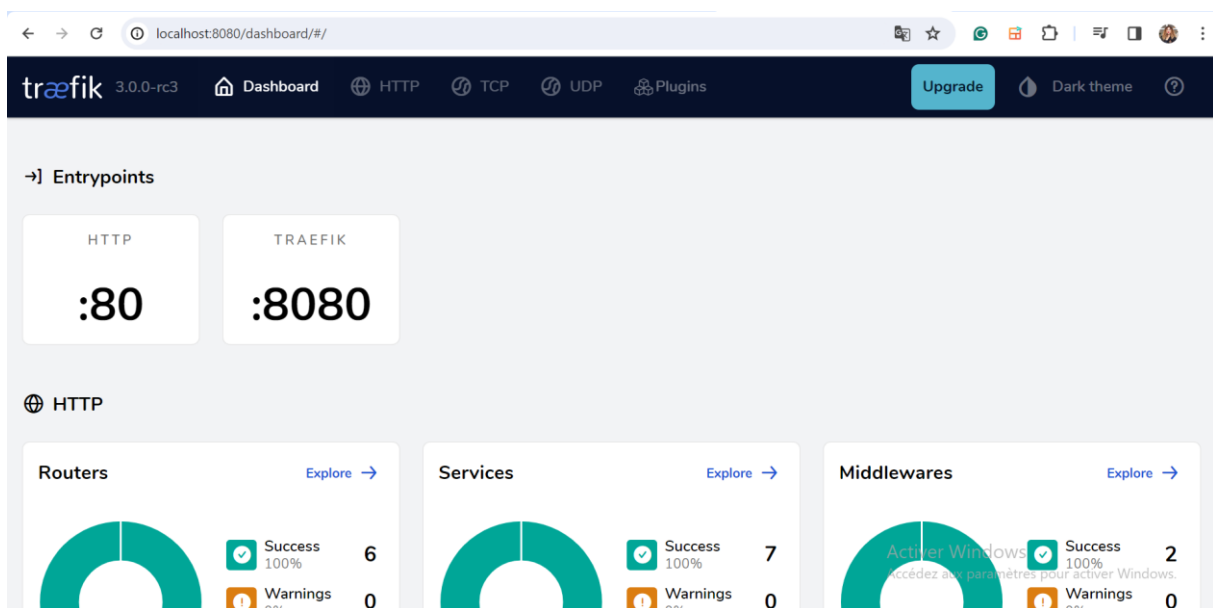


FIGURE 10 : DASHBOARD TRAEFIK

I.4 Interfaces

Nous allons présenter quelques scénarios de notre application.

• Scénario d'inscription

L'utilisateur peut s'inscrire en remplissant correctement tous les champs de l'interface Signup illustrée par la figure 11.

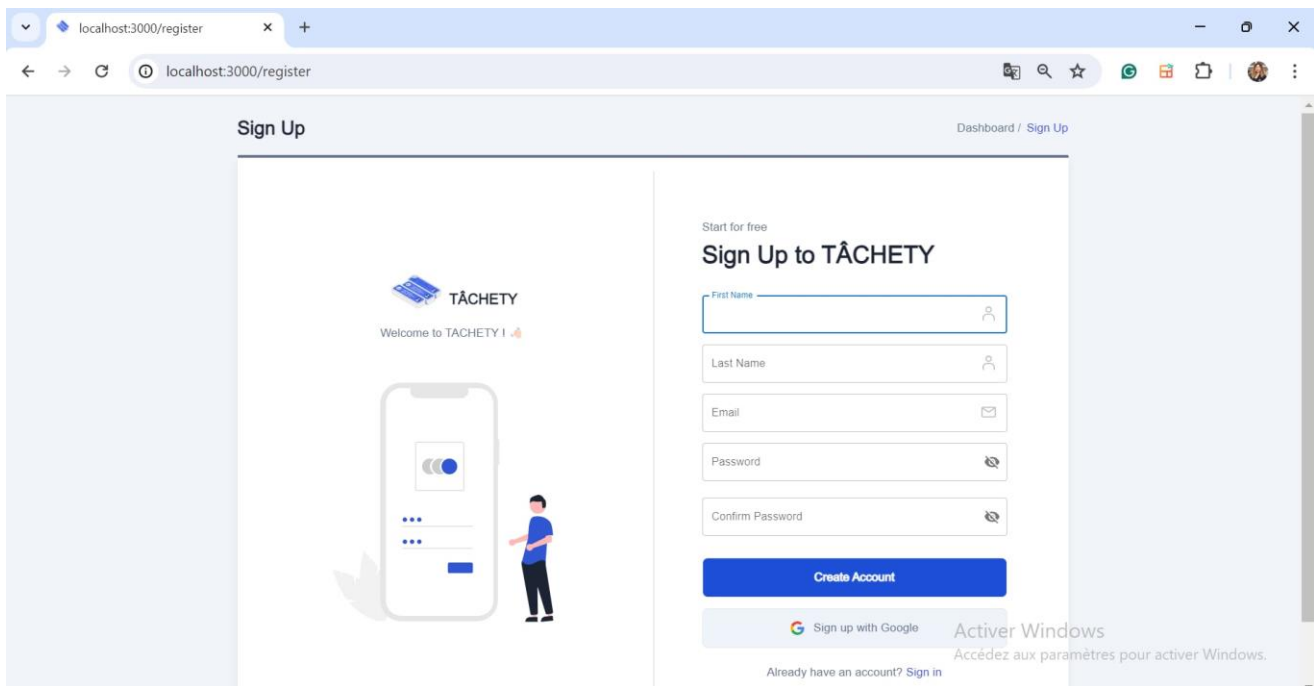


FIGURE 11 : INTERFACE D'INSCRIPTION

• Scénario d'authentification

L'utilisateur peut s'authentifier en saisissant correctement son email et mot de passe dans l'interface Login illustrée par la figure 12.

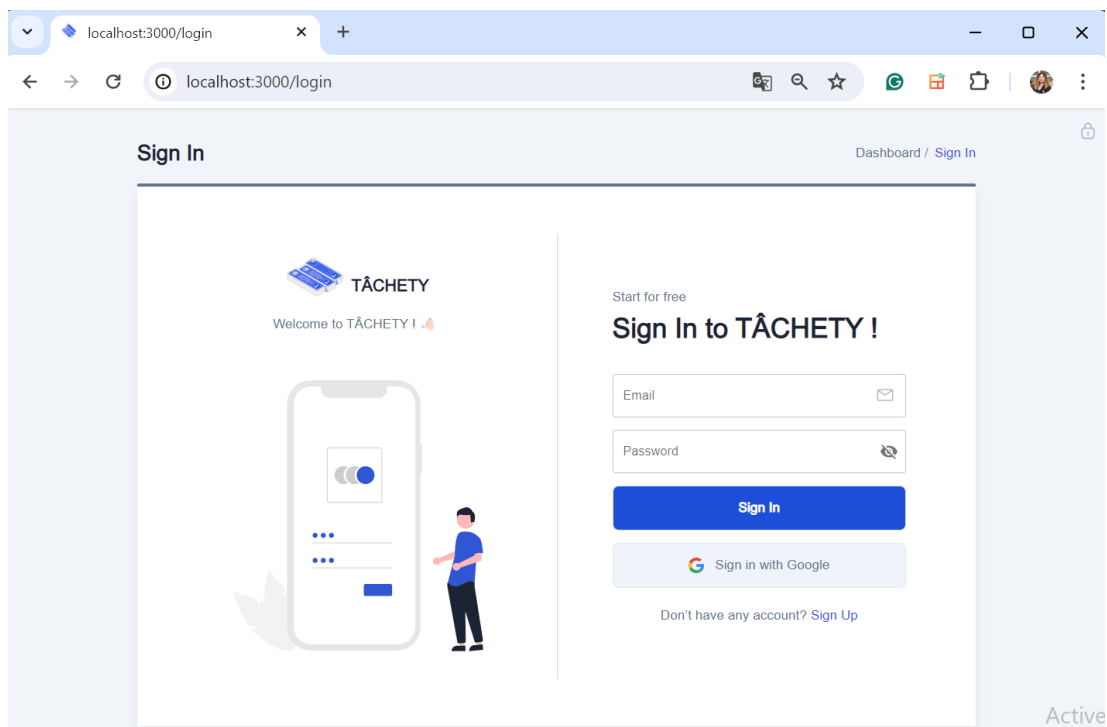


FIGURE 12 : INTERFACE D'AUTHENTIFICATION

L'utilisateur authentifié est redirigé vers l'interface Dashboard illustrée par la figure 13.

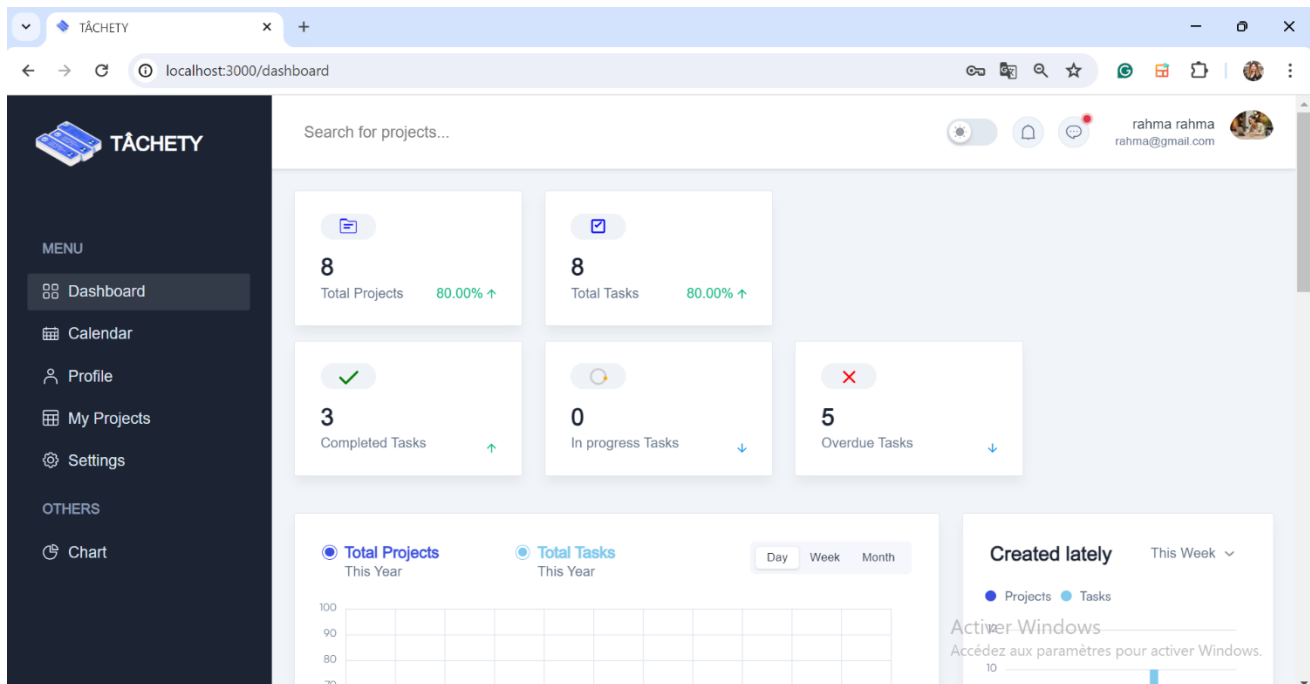


FIGURE 13 : INTERFACE DU TABLEAU DE BOARD

• Scénario d'ajout d'un projet

L'utilisateur peut accéder à ses projets dans l'interface My projects illustrée par la figure 13.

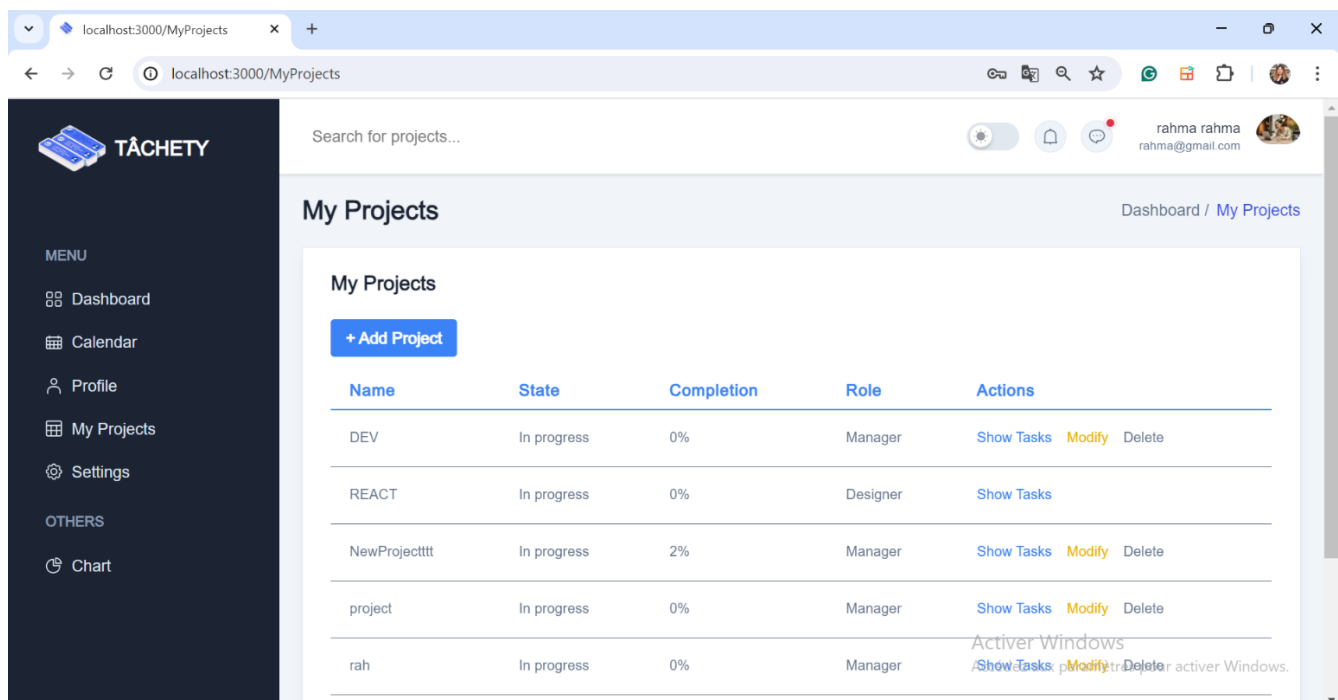


FIGURE 14 : INTERFACE DES PROJETS

L'utilisateur peut ajouter un projet sur la même l'interface My projects illustrée par la figure 14. Le rôle de Manager lui est automatiquement assigné.

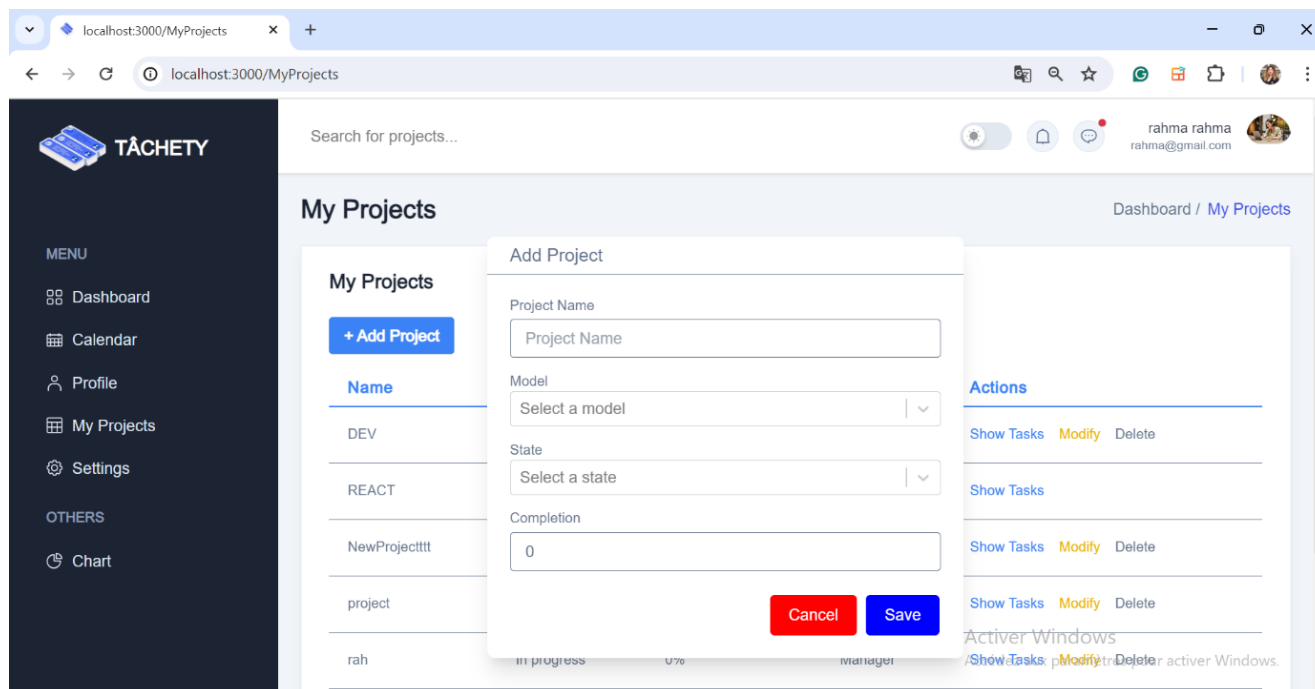


FIGURE 15 : INTERFACE D'AJOUT D'UN PROJET

- **Scénario de gestion de projets**

Le manager peut modifier ou supprimer un projet sur la même l'interface My projects illustrée par la figure 17.

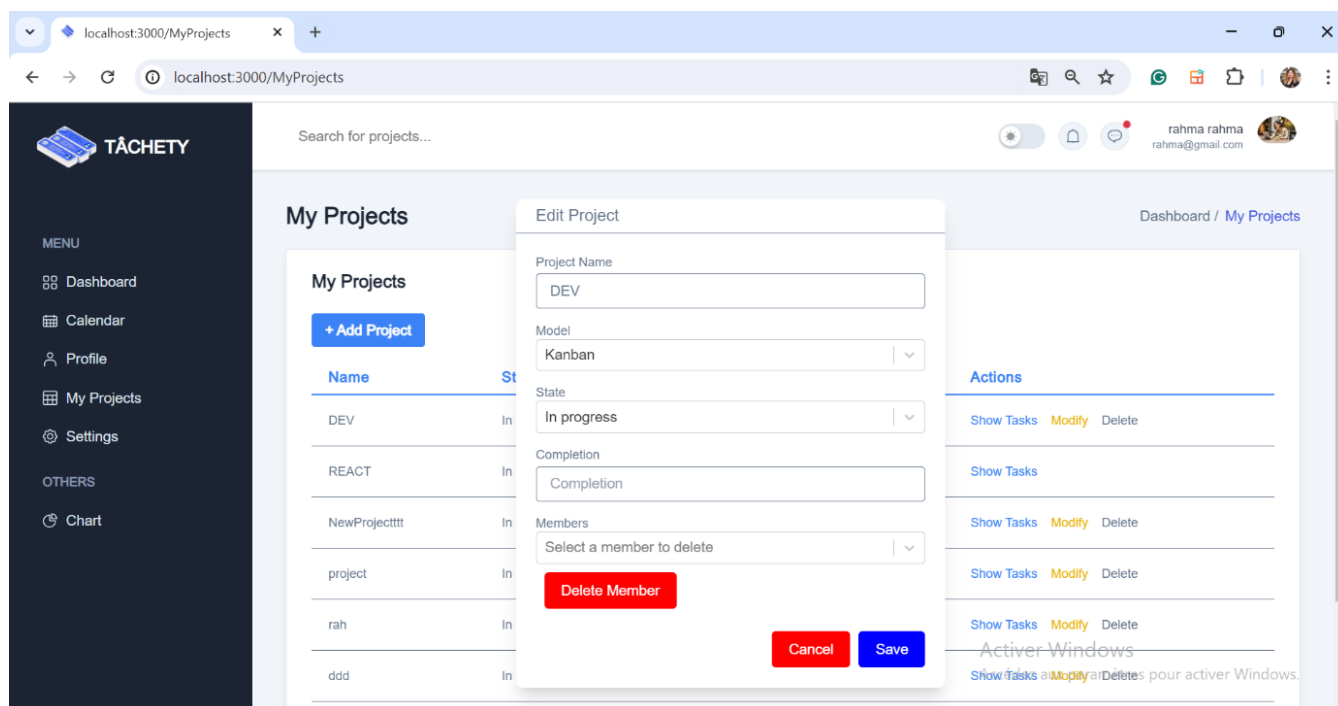


FIGURE 16 : INTERFACE DE MODIFICATION D'UN PROJET

Seul le manager peut ajouter un membre ou une colonne à un projet, illustrée par la figure 14.

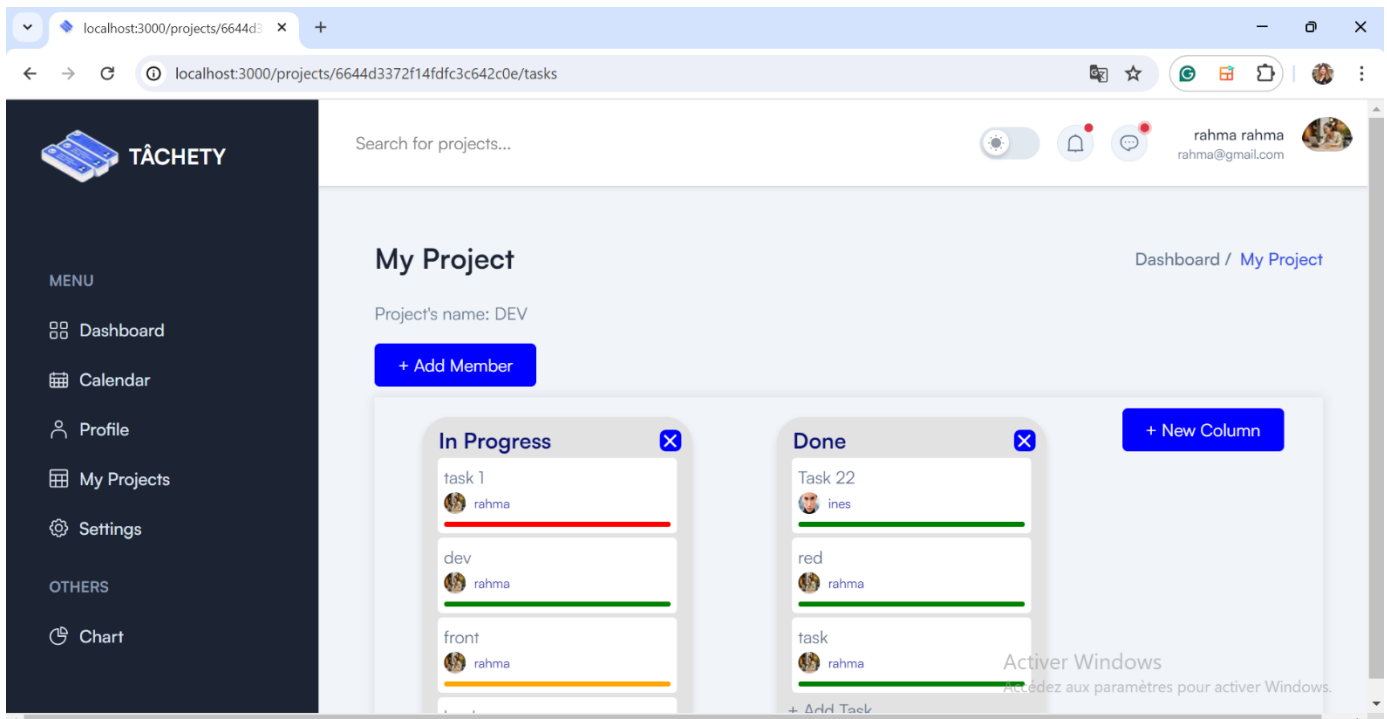


FIGURE 17 : INTERFACE D'AJOUT DE MEMBRES ET DE COLONNES

- **Scénario de gestion de tâches**

Un utilisateur peut gérer ses propres tâches (Ajouter, modifier, supprimer et glisser-déposer). Ces cas d'utilisations sont illustrés par les figures 18 et 19.

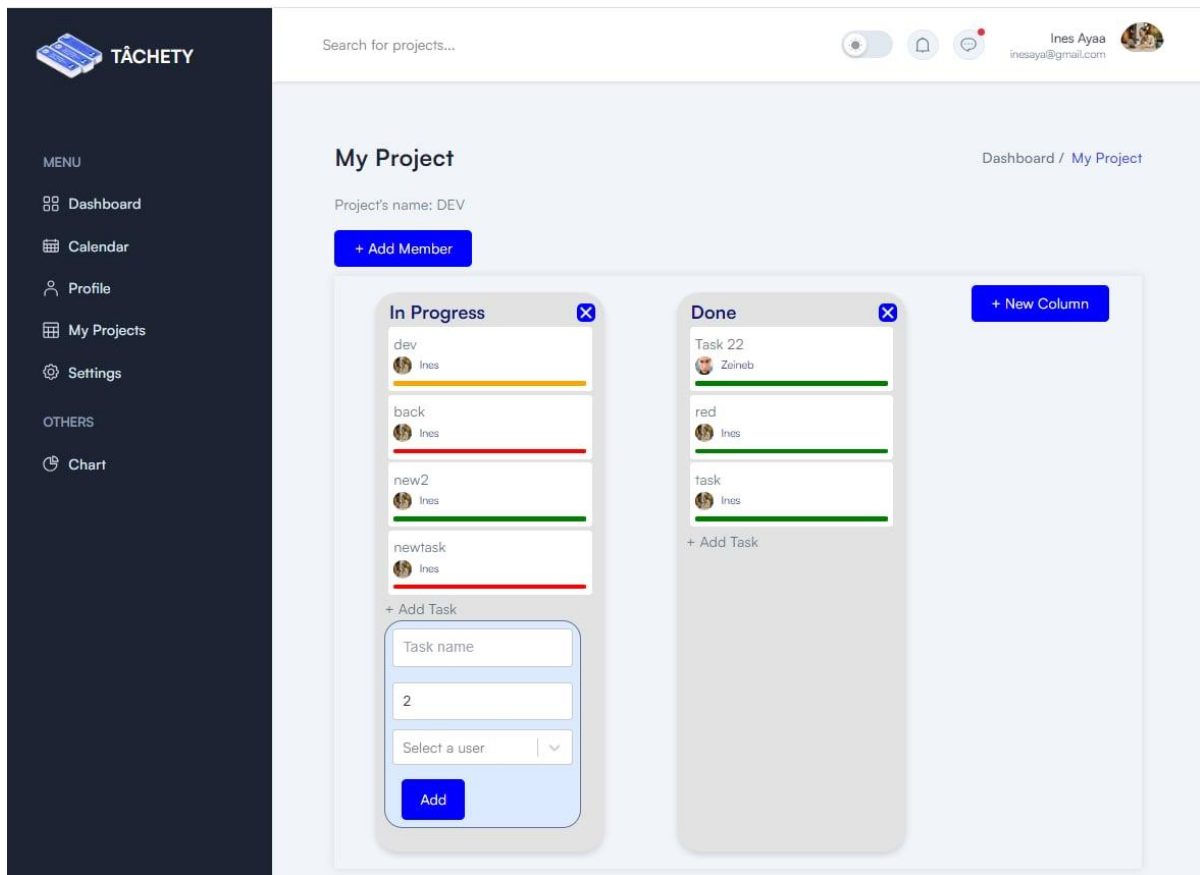


FIGURE 18 : INTERFACE D'AJOUTE D'UNE TACHE

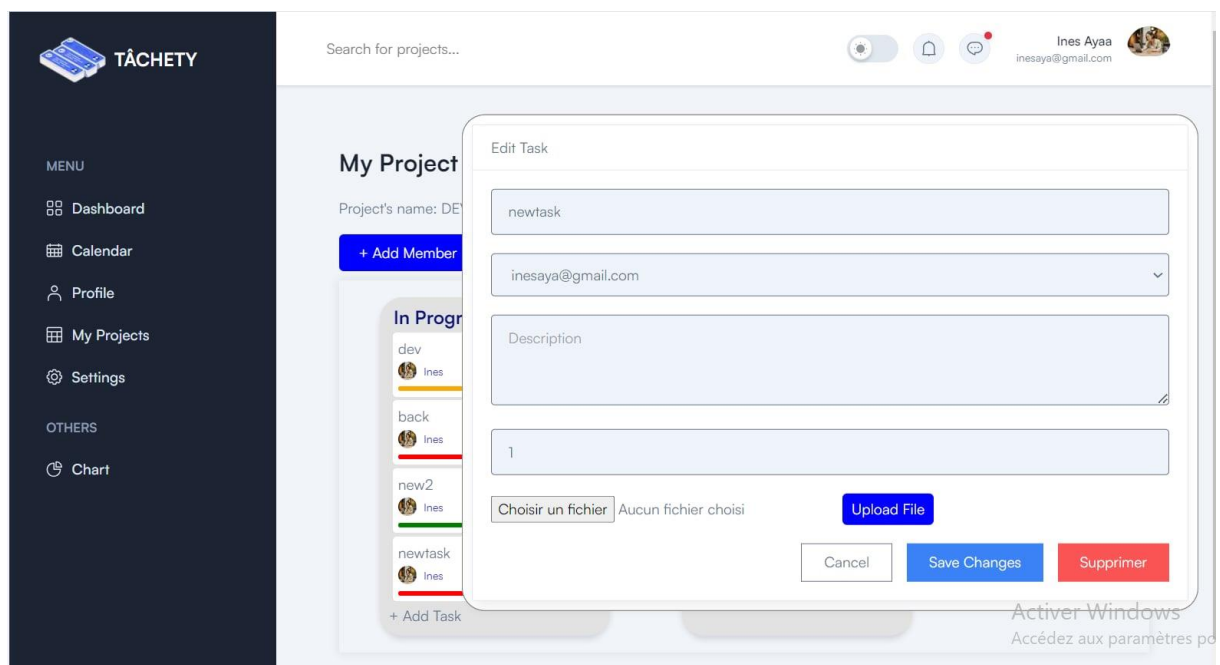


FIGURE 19 : INTERFACE DE MODIFICATION D'UNE TACHE

Une notification est envoyée à l'utilisateur dès qu'il est ajouté à un projet ou qu'une tâche lui a été assigné. La liste des notifications est illustrée par la figure 20.

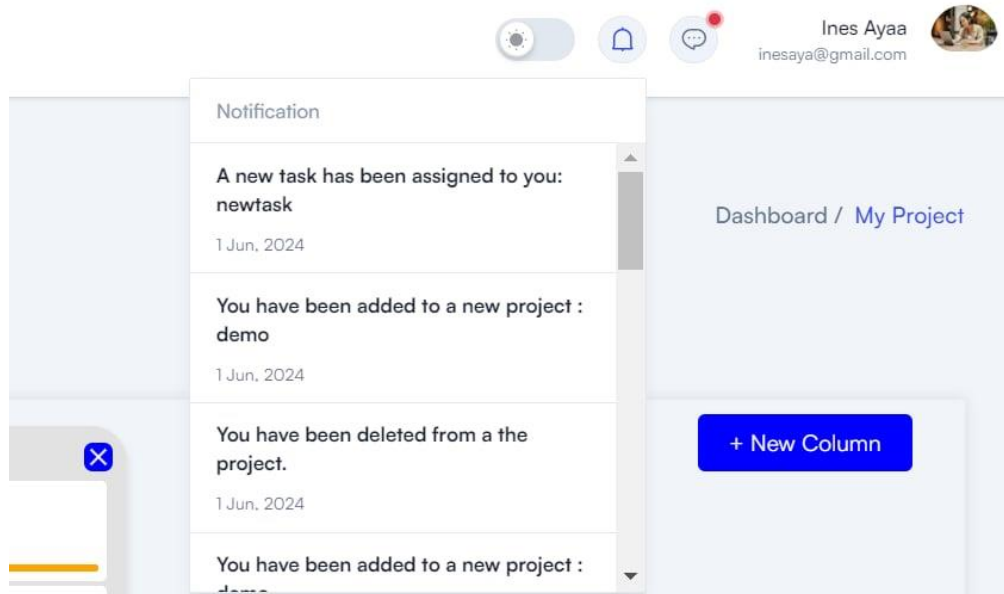


FIGURE 20 : NOTIFICATIONS

I.5 Test logiciel

I.5.1 Test Backend

Voici quelques exemples de tests réalisés sur le backend à l'aide du logiciel Postman :

- **Test de récupération de la liste des projets de l'utilisateur connecté**

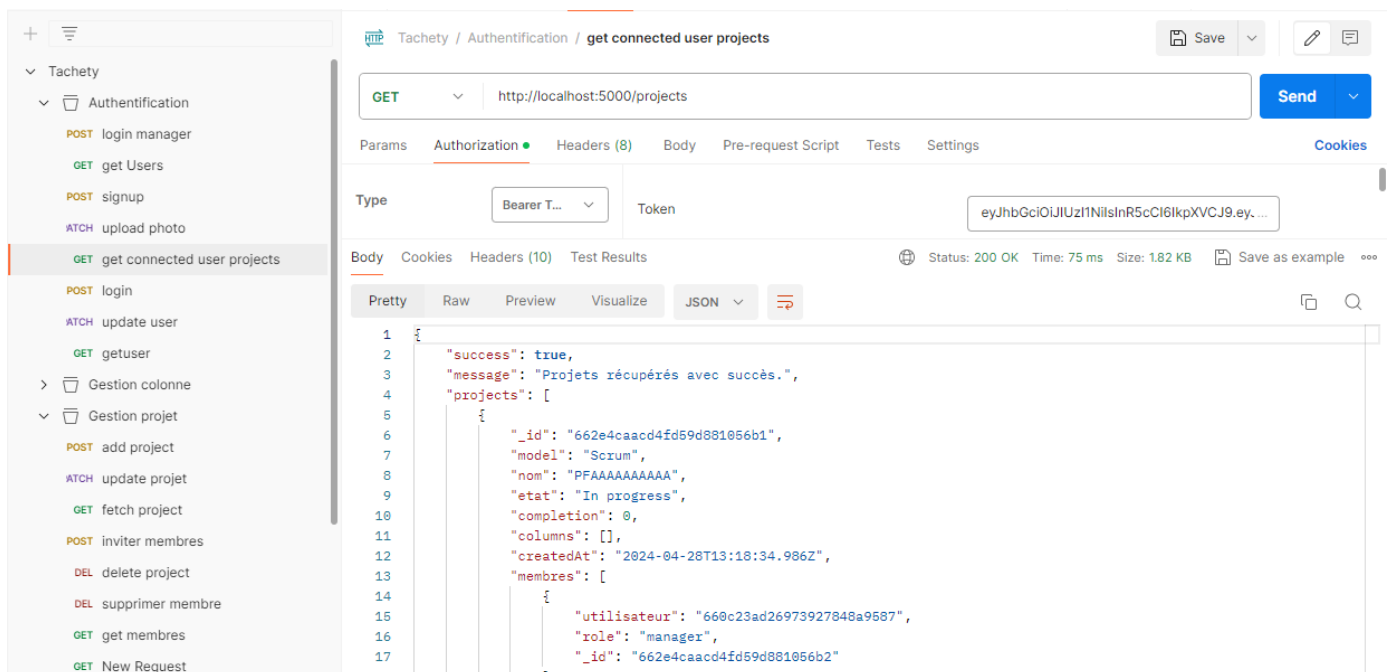


FIGURE 21 : POSTMAN : TEST DE RECUPERATION DE LA LISTE DES PROJETS

- **Test d'ajout d'un projet**

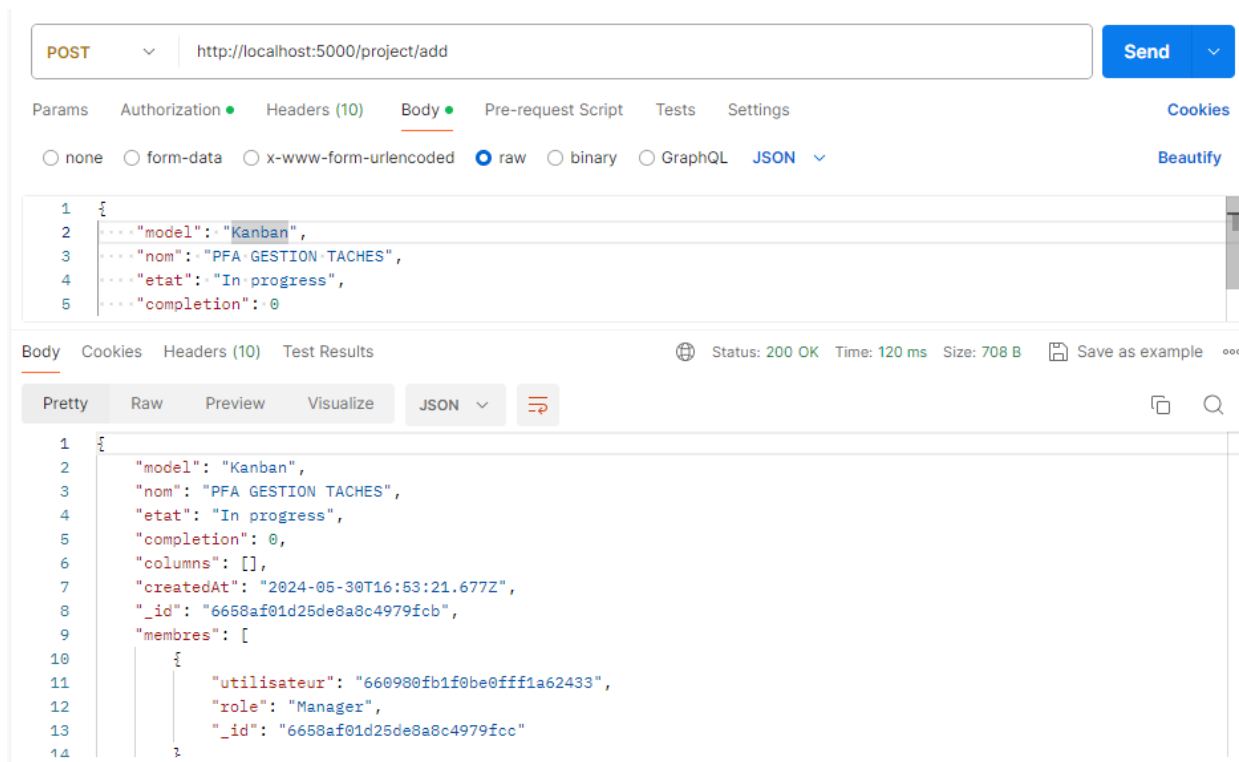


FIGURE 22 : POSTMAN : TEST DE CREATION D'UN PROJET

I.5.2 Test Frontend

Voici quelques exemples de tests réalisés sur le frontend à l'aide du logiciel Cypress :

- **Test de l'interface d'inscription**

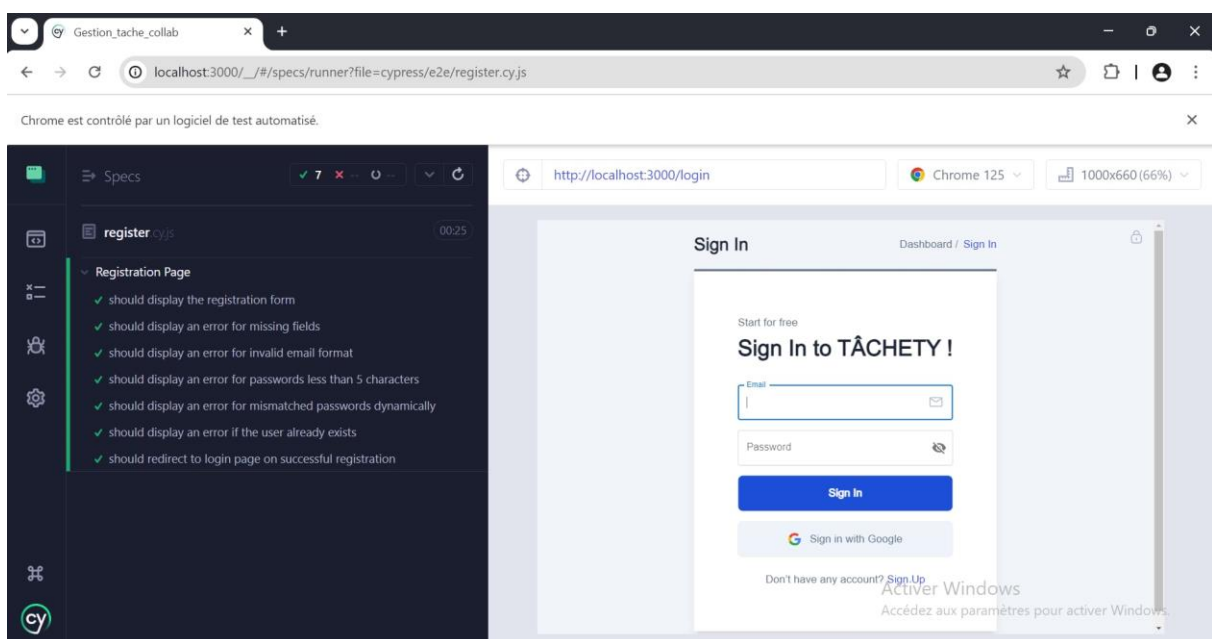


FIGURE 23 : CYPRESS : VALIDATION DU TEST DE L'INTERFACE D'INSCRIPTION

- **Test de l'interface d'authentification**

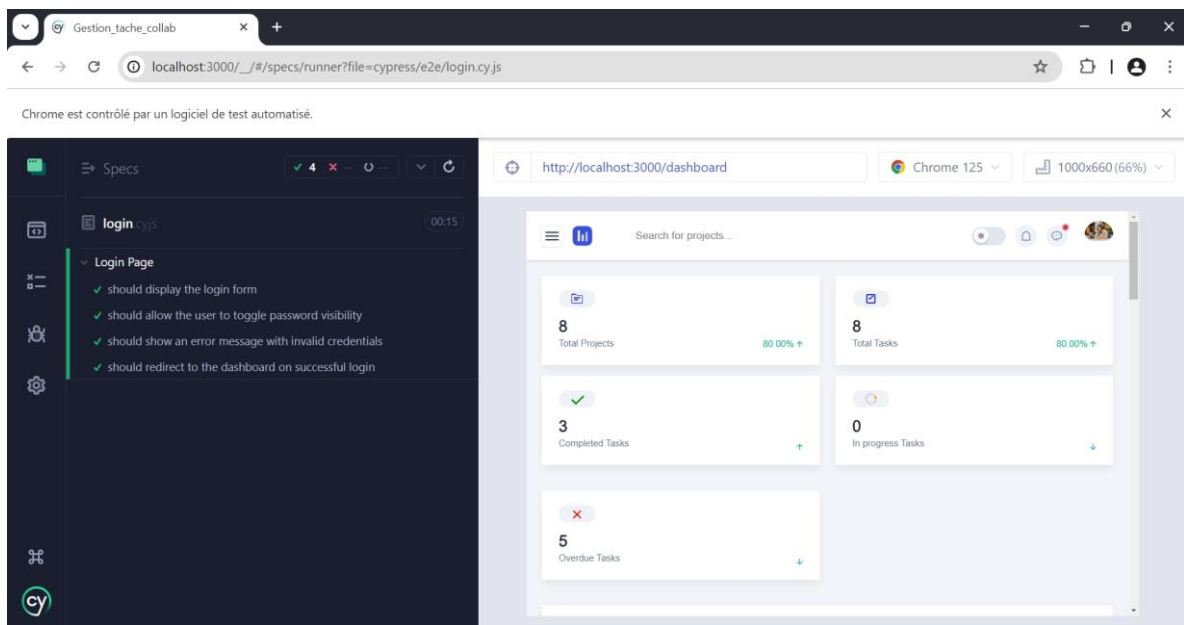


FIGURE 24 : CYPRESS : VALIDATION DU TEST DE L'INTERFACE D'AUTHENTIFICATION

- **Test de l'interface d'ajout d'un projet**

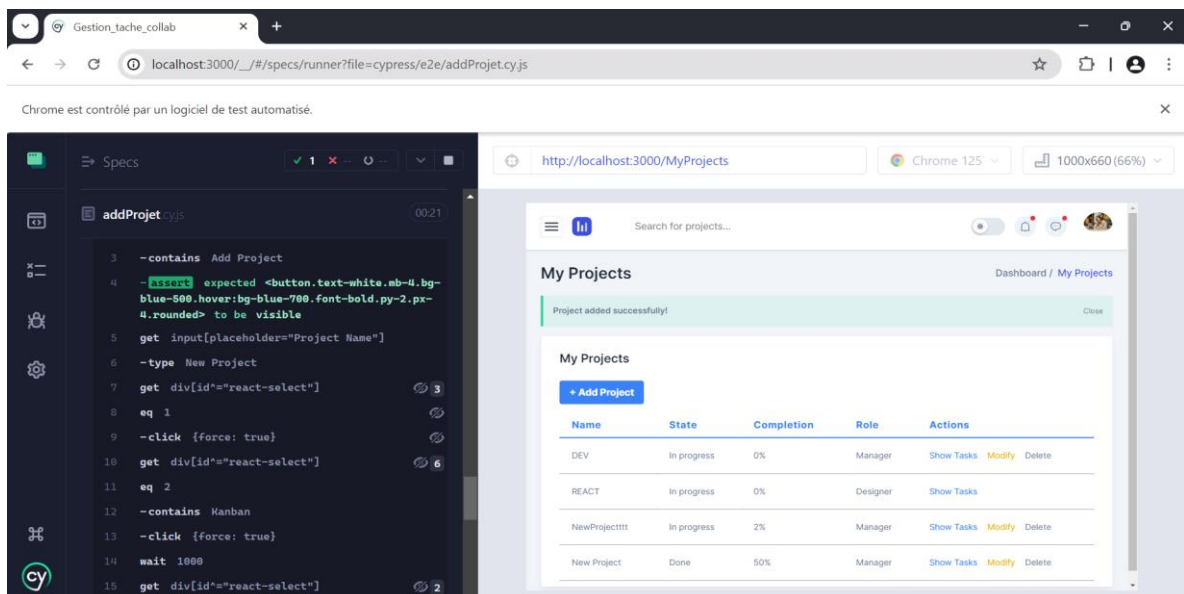


FIGURE 25 : CYPRESS : VALIDATION DU TEST D'AJOUT D'UN PROJET

Conclusion

Dans ce chapitre, nous avons présenté les outils de développement, la charte graphique, le processus de dockerisation de l'application ainsi que les différents scénarios en passant par les tests réalisés au cours du développement de la solution.

Conclusion générale et perspectives

Dans le cadre de notre projet de fin d'année, nous avons conçu et développé une application web de gestion des tâches collaborative en adoptant une approche DevOps. Le présent document est une présentation du travail réalisé, détaillant toutes les étapes que nous avons traversées pour parvenir au résultat attendu. Les fonctionnalités prévues de notre application web « Tâchety » ont été mises en œuvre avec succès.

Ce projet DevOps a non seulement renforcé nos compétences techniques, mais il a également aiguisé notre intérêt pour le développement et les tests logiciels. En travaillant sur ce projet, nous avons pu approfondir notre compréhension des outils tels que Node.js, Next.js, Postman, Cypress et Docker tout en appliquant des pratiques rigoureuses de développement et de test pour assurer la qualité et l'efficacité de la solution finale. Il nous a également aidés à comprendre les défis inhérents à la gestion du temps et des efforts, notamment en équilibre avec les cours et les examens.

Cette expérience nous a permis de mieux appréhender les défis et les opportunités du développement logiciel, consolidant ainsi notre passion pour ce domaine en constante évolution.

Bibliographie

- [1 «Trello,» [En ligne]. Available: <https://trello.com/fr/about>. [Accès le 30 05 2024].
]
- [2 «cadre emploi,» [En ligne]. Available: <https://www.cadremploi.fr/editorial/conseils/conseils-carriere/detail/article/jose-pas-demander-mais-quest-ce-que-jira.html>. [Accès le 30 05 2024].
]
- [3 «atlassian,» [En ligne]. Available:
] <https://www.atlassian.com/fr/devops#:~:text=DevOps%20est%20un%20ensemble%20de,%C3%A9quipes%20de%20d%C3%A9veloppement%20et%20informatiques..> [Accès le 30 05 2024].
- [4 «Atlassian,» [En ligne]. Available: <https://www.atlassian.com/fr/agile/scrum/backlogs>. [Accès le 1 05 2024].
]
- [5 «MDN web docs,» [En ligne]. Available: https://developer.mozilla.org/fr/docs/Learn/Server-side/Express_Nodejs/Introduction. [Accès le 05 05 2024].
]
- [6 «mongodb,» [En ligne]. Available: <https://www.mongodb.com/fr-fr/atlas>. [Accès le 10 05 2024].
]
- [7 «data scientest,» [En ligne]. Available: <https://datascientest.com/github-tout-savoir>. [Accès le 10 05 2024].
]
- [8 «learn microsoft,» [En ligne]. Available: <https://learn.microsoft.com/fr-fr/windows/wsl/faq>. [Accès le 10 05 2024].
]
- [9 «redhat,» [En ligne]. Available: <https://www.redhat.com/fr/topics/containers/what-is-docker>. [Accès le 10 05 2024].
]
- [1 [En ligne]. Available: <https://www.getapp.fr/software/91591/zenefits>.
0]
- [1 [En ligne]. Available: <https://www.getapp.fr/software/101473/namely>.
1]
- [1 [En ligne]. Available: <https://www.getapp.fr/software/91247/cezanne-hr>.
2]