

## Remerciements

Je tiens à exprimer ma profonde gratitude envers **M. Wael Mejdoub**, le chef de l'entreprise **Aftercode**. Il m'a offert cette opportunité de stage, qui a été une étape importante dans mon parcours professionnel et personnel. Son partage de connaissances et son leadership ont grandement contribué à mon développement.

Je souhaite également remercier **Madame Eya Dhen** pour son encadrement, sa disponibilité et son aide précieuse tout au long de mon stage. Sa patience et son soutien ont été inestimables.

Un grand merci à Madame **Mahassen Khemiri** pour ses précieuses corrections et sa disponibilité constante lors de la préparation de ce rapport.

Je voudrais également remercier le **Orange Tech Club** qui a organisé le hackathon '**Hack4Good**'. C'est lors de cet événement que j'ai eu la chance de gagner cette offre de stage chez Aftercode, après avoir été classé premier avec mon équipe. Cette victoire a été un moment marquant et a ouvert la porte à cette expérience enrichissante.

Un grand merci à ma famille pour leur soutien constant tout au long de ce stage. Leur encouragement a été une source d'inspiration et de motivation pour moi.

Enfin, je voudrais exprimer ma reconnaissance à l'équipe de design de **Aftercode**. Travailler avec eux a été une expérience d'apprentissage précieuse et j'ai beaucoup apprécié leur professionnalisme et leur créativité.

En conclusion, ce stage a été une expérience incroyablement enrichissante et je suis reconnaissant pour toutes les opportunités et le soutien que j'ai reçus.

## Table des matières

Remerciements .....	1
Introduction générale.....	1
Chapitre I. Cadre général de projet .....	2
1. Présentation de l'organisme d'accueil .....	2
1.1. Présentation de Aftercode.....	2
1.2. Organigramme de Aftercode .....	2
2. Cadre et Objectif de stage .....	3
3. Étude de l'existant.....	3
4. Méthodologie agile.....	4
5. Product Backlog .....	5
5.1. Mise en œuvre d'un contrôle d'accès basé sur les rôles.....	5
5.2. Meilleures pratiques de sécurité .....	6
5.3. Développer un module pour ajouter et suivre les ingrédients .....	7
5.4. Mise en œuvre de mises à jour d'inventaire en temps réel .....	8
5.5. Construire une interface de suivi des commandes pour le caissier .....	8
6. Conclusion.....	9
Chapitre II. Étude Conceptuelle .....	10
Introduction .....	10
1. Conception des interfaces.....	10
2. Conception UML.....	11
Conclusion.....	16
Chapitre III. Réalisation.....	17
Introduction .....	17
1. Environnement matériel .....	17
2. Les choix techniques .....	17
2.1. Visual Studio Code.....	18

2.2.	GitHub et Git.....	18
2.3.	Djnago .....	19
2.4.	Next.js .....	19
2.5.	Shadcn .....	20
2.6.	Framer Motion.....	20
2.7.	Tailwind CSS.....	21
2.8.	GraphQL.....	21
2.9.	Vercel.....	22
2.10.	Axios .....	22
2.11.	Redux .....	23
2.12.	Slack .....	23
2.13.	PostgreSQL .....	24
2.14.	Diagrams.net .....	24
2.15.	Figma.....	25
2.16.	Redux DevTools .....	25
2.17.	Neon .....	26
3.	Les interfaces réalisées.....	27
	Conclusion.....	38
	Conclusion générale .....	39
	Webographie.....	40

## Liste des figures

Figure 1: Logo de l'organisme d'accueil Aftercode.....	2
Figure 2: Organigramme de Aftercode.....	3
Figure 3: Pages d'Authentification - Design Figma.....	10
Figure 4: Pages Cassier - Design Figma. ....	11
Figure 5: Digramme de cas d'utilisation – GestionCassier.....	12
Figure 6: Digramme de cas d'utilisation - GestionAdmin.....	13
Figure 7: Diagramme de séquence pour l'authentification. ....	14
Figure 8: Digramme de class.....	15
Figure 10: Logo Visual Studio Code.....	18
Figure 11: Logo Git.....	18
Figure 12: Logo GitHub.....	18
Figure 13: Logo Django. ....	19
Figure 14: Logo Next.js. ....	19
Figure 15: Logo Shadcn. ....	20
Figure 16: Logo Framer Motion.....	20
Figure 17: Logo Tailwind CSS.....	21
Figure 18: Logo GraphQL.....	21
Figure 19: Logo Vercel.....	22
Figure 20: Logo Axios.....	22
Figure 21: Logo Redux. ....	23
Figure 22: Logo Slack.....	23
Figure 23: Logo PostgreSQL. ....	24
Figure 24: Logo Diagrams.net. ....	24
Figure 25: Logo Figma.....	25
Figure 26: Logo Redux DevTools. ....	25
Figure 27: Logo Neon .....	26
Figure 28: Environnement de travail backend avec Django et VS Code.....	27
Figure 29: Environnement de travail frontend avec Next.js et VS Code. ....	28
Figure 30: Panel Admin Django.....	29
Figure 31: Interface du Playground de l'API GraphQL.....	29
Figure 32: Fusion des dépôts sur GitHub 1.....	30
Figure 33: Fusion des dépôts sur GitHub2.....	30

Figure 34: Courriel de Vercel confirmant la mise à jour du projet 'vitvent-frontend' .....	31
Figure 35: Tous les contrôles ont réussi sur Vercel et GitHub CI/CD.....	31
Figure 36: Page de Connexion. ....	32
Figure 37: Page de Création de Compte.....	32
Figure 38: Page d'enregistrement du Restaurant - Étape 1. ....	33
Figure 39: Page d'enregistrement du Restaurant - Étape 2. ....	33
Figure 40: Page d'enregistrement du Restaurant - Étape 3. ....	34
Figure 41: Page d'enregistrement du Restaurant - Étape 4. ....	34
Figure 42: Interface de la console Google Cloud.....	35
Figure 43: Page de caisse. ....	35
Figure 44: Onglet 'Détails du produit'.....	36
Figure 45: Utilisation de Redux DevTools lors du développement. ....	36
Figure 46: Onglet 'Paiement de l'ordre'.....	37
Figure 47: Vue du dépôt Front-End sur GitHub.....	37
Figure 48: Table de produit dans la base de données. ....	38

## Introduction générale

Ce stage a été effectué chez **Aftercode**, une entreprise reconnue pour son expertise dans le développement de solutions technologiques innovantes, dans le cadre du développement d'une plateforme innovante destinée à révolutionner la gestion des restaurants. Cette plateforme, nommée **Vitvent**, a été conçue pour répondre à un besoin réel des restaurateurs qui cherchent à optimiser leur service, fidéliser leur clientèle et augmenter leur chiffre d'affaires.

Vitvent se présente comme une solution complète qui permet aux restaurateurs de gérer leurs commandes, que ce soit en restaurant ou par livraison, et de suivre la gestion de stock et de management de tout le restaurant. De plus, la plateforme offre la possibilité de créer des comptes à des rôles différents, comme caissier, chef, livreur, etc., offrant ainsi une flexibilité et une personnalisation sans précédent.

Le projet Vitvent a été une occasion pour mettre en pratique les compétences acquises à l'Institut Supérieur d'Etudes Technologiques de Nabeul, notamment en matière Python avancé et Modélisation UML. Cela a permis non seulement d'appliquer les connaissances théoriques acquises, mais aussi d'acquérir une expérience pratique précieuse dans le développement d'applications web.

Le rapport de stage est organisé en trois chapitres principaux.

Le premier chapitre présente l'entreprise Aftercode, le contexte du projet Vitvent, les besoins et les attentes des clients potentiels de la plateforme, ainsi que la méthodologie Agile utilisée pour gérer le projet.

Le deuxième chapitre présente la conception de la plateforme Vitvent.

Le dernier chapitre détaille les principales fonctionnalités développées pour la plateforme Vitvent, ainsi que les choix techniques.

En somme, ce rapport détaille mon expérience de stage chez Aftercode, où j'ai contribué au développement de la plateforme Vitvent. Je vous invite à découvrir ce travail passionnant dans les chapitres qui suivent.

# Chapitre I. Cadre général de projet

## 1. Présentation de l'organisme d'accueil

### 1.1. Présentation de Aftercode

Aftercode est une agence informatique tunisienne créée en 2019, spécialisée dans le développement sur mesure des applications Web et mobile. Elle est reconnue pour son expertise et son approche innovante, appliquant la méthodologie Agile pour un suivi quotidien de l'avancement des projets. Aftercode aide ses clients à réussir en créant des identités de marque et des expériences numériques. L'agence est située à Dar Chaabane, Nabeul, en Tunisie.



*Figure 1: Logo de l'organisme d'accueil Aftercode.*

### 1.2. Organigramme de Aftercode

La structure administrative de l'entreprise est bien organisée et est basée sur le partage de l'information faisant de Aftercode un environnement de travail collaboratif et efficace. Cette approche favorise la transparence, l'innovation et la responsabilité, ce qui contribue à la qualité des services fournis par Aftercode.



# « More Than Code » ORGANIGRAMME AFTERCODE

PDG/FONDATEUR

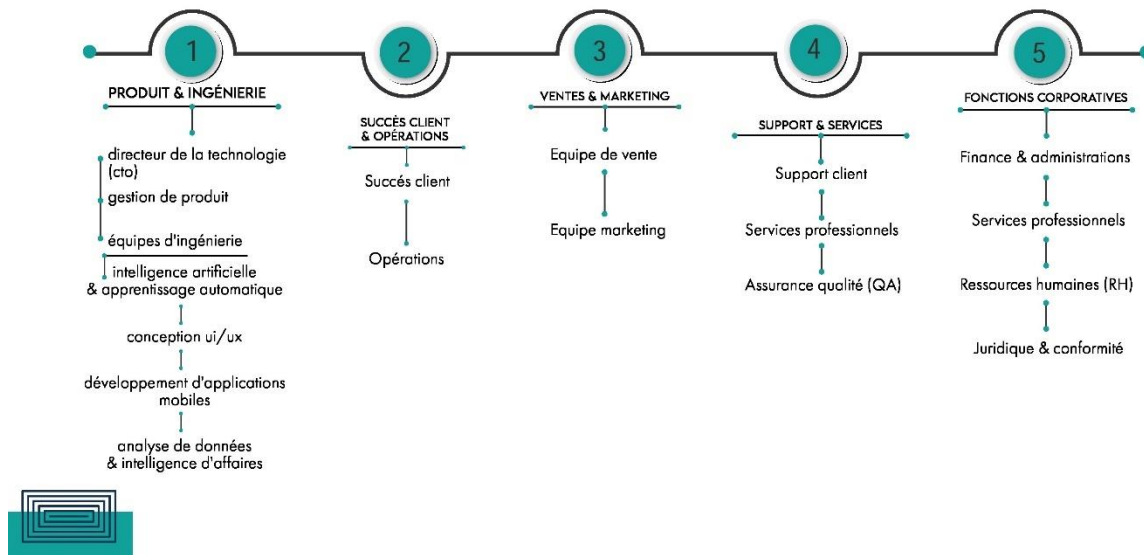


Figure 2: Organigramme de Aftercode

## 2. Cadre et Objectif de stage

Le stage de perfectionnement chez Aftercode a été réalisé dans le cadre du développement de la plateforme Vitvent. L'objectif principal de ce stage était d'appliquer et de perfectionner les compétences acquises à l'Institut Supérieur d'Etudes Technologiques de Nabeul, notamment en Python avancé et Modélisation UML, dans un contexte professionnel réel.

## 3. Étude de l'existant

Aftercode, en tant qu'entreprise de développement web, a identifié un besoin réel chez les restaurateurs pour une solution plus optimisée et flexible. Les défis auxquels sont confrontés les restaurateurs comprennent la gestion efficace des commandes, le suivi précis des stocks, la coordination du personnel et la personnalisation des rôles au sein du restaurant.

La plateforme Vitvent a été conçue pour répondre à ces défis. Elle offre une solution complète qui permet aux restaurateurs de gérer leurs commandes, que ce soit en restaurant ou par livraison, et de suivre la gestion de stock et le management de tout le restaurant.



De plus, la plateforme offre la possibilité de créer des comptes à des rôles différents, comme caissier, chef, livreur, etc., offrant ainsi une flexibilité et une personnalisation sans précédent.

En conclusion, l'étude de l'existant a permis de mettre en évidence les besoins réels des restaurateurs et de concevoir une solution adaptée pour y répondre.

Dans les chapitres suivants, nous détaillerons les différentes étapes de la conception et développement de la plateforme Vitvent.

#### 4. Méthodologie agile

La méthodologie Agile est une approche de gestion de projet qui privilégie l'itération et l'adaptabilité. Elle est particulièrement utilisée dans le développement de logiciels. Chez Aftercode, nous avons adopté cette méthodologie pour plusieurs raisons :

- **Flexibilité** : Agile permet des changements et des ajustements tout au long du projet. Cela signifie que si les besoins du client évoluent, le projet peut s'adapter pour répondre à ces nouveaux besoins.
- **Livraison continue** : Avec Agile, nous avons un produit fonctionnel que nous pouvons présenter au client. Cela permet une livraison continue et une amélioration constante du produit.
- **Collaboration** : Agile encourage une collaboration étroite entre toutes les parties prenantes du projet, y compris les développeurs, les chefs de projet, et le client. Cela favorise une meilleure communication et une meilleure compréhension des objectifs du projet.
- **Satisfaction du client** : En fin de compte, l'objectif de la méthodologie Agile est d'améliorer la satisfaction du client. En livrant régulièrement des versions fonctionnelles du produit et en répondant rapidement aux besoins changeants du client, nous pouvons assurer que le client est toujours satisfait du travail en cours.

## 5. Product Backlog

Dans le cadre du développement de la plateforme Vitvent par Aftercode, un élément clé de notre processus Agile a été l'utilisation du Product Backlog. Le Product Backlog est essentiellement une liste ordonnée de tout ce qui est nécessaire pour le projet, et il est la source unique de toutes les exigences pour tout travail à effectuer sur le projet.

Pour Vitvent, le Product Backlog a été un outil précieux pour aider l'équipe à comprendre les priorités et à planifier efficacement le travail. Il a permis de définir clairement les fonctionnalités à développer, les bugs à corriger et les améliorations à apporter. Chaque élément du backlog, appelé "User Story", a été clairement défini avec des critères d'acceptation pour garantir que tous les membres de l'équipe comprennent ce qui est attendu.

### 5.1. Mise en œuvre d'un contrôle d'accès basé sur les rôles

En tant qu'administrateur système, je souhaite définir des rôles et des permissions associées pour les membres du personnel afin de garantir un accès sécurisé et spécifique aux fonctionnalités.

Critères d'acceptation :

- L'administrateur peut accéder à la section "Gestion des rôles" pour créer, modifier ou supprimer des rôles.
- Chaque rôle dispose d'autorisations prédéfinies correspondant aux responsabilités du membre du personnel concerné.
- Les modifications apportées aux rôles sont appliquées en temps réel, se reflétant immédiatement dans l'accès des utilisateurs.

En tant que serveur, je souhaite n'avoir accès qu'aux fonctionnalités pertinentes pour la prise et la gestion des commandes des clients.

Critères d'acceptation :

- Les permissions du rôle de serveur permettent d'accéder à la gestion des commandes, à l'attribution des tables et aux fonctionnalités d'interaction avec les clients.
- Le serveur n'a pas accès aux modules sensibles tels que les paramètres financiers ou d'administration.

## 5.2. Meilleures pratiques de sécurité

En tant qu'utilisateur conscient de la sécurité, je souhaite que l'application utilise des méthodes de cryptage avancées pour sécuriser mon mot de passe et mes informations personnelles.

Critères d'acceptation :

- Les mots de passe sont sécurisés, hachés et stockés en utilisant des algorithmes de cryptage standard de l'industrie.
- Les informations personnelles sont cryptées en transit et au repos pour prévenir l'accès non autorisé.
- Des audits de sécurité réguliers sont effectués pour identifier et traiter les vulnérabilités potentielles.

En tant qu'administrateur, je souhaite avoir accès à des journaux d'audit complets pour surveiller les activités des utilisateurs et maintenir un système sécurisé.

Critères d'acceptation :

- L'application enregistre les actions critiques des utilisateurs, y compris les tentatives de connexion, les changements de rôle et les réinitialisations de mot de passe.
- Les administrateurs peuvent examiner et exporter les journaux pour des fins d'audit.
- Les journaux sont stockés de manière sécurisée et protégés contre la falsification.

### 5.3. Développer un module pour ajouter et suivre les ingrédients

En tant que gérant de restaurant, je souhaite ajouter de nouveaux ingrédients et fournitures à l'inventaire, assurant un suivi précis des niveaux de stock.

Critères d'acceptation :

- Le système fournit une interface conviviale pour ajouter de nouveaux articles à l'inventaire.
- Les informations requises comprennent le nom de l'article, la catégorie, la quantité de stock initiale et les détails du fournisseur.
- Les articles sont classés logiquement pour une gestion facile.

En tant que chef, je souhaite pouvoir mettre à jour les détails des ingrédients existants, tels que la quantité et les informations sur le fournisseur, afin de maintenir un inventaire précis.

Critères d'acceptation :

- Le système permet aux utilisateurs autorisés de modifier les détails des articles existants.
- Les modifications apportées aux détails des articles, y compris les ajustements de quantité, sont enregistrées à des fins d'audit.
- Seuls les utilisateurs disposant des permissions appropriées peuvent modifier les entrées d'inventaire

#### 5.4. Mise en œuvre de mises à jour d'inventaire en temps réel

En tant que caissier, je souhaite que l'inventaire soit automatiquement mis à jour en temps réel lors du traitement des commandes clients pour éviter la survente ou les écarts de stock.

Critères d'acceptation :

- Les niveaux de stock sont automatiquement ajustés lorsqu'une commande est traitée, en tenant compte des quantités vendues.
- Les mises à jour en temps réel sont reflétées dans le tableau de bord de l'inventaire.

#### 5.5. Construire une interface de suivi des commandes pour le caissier

En tant que caissier, je souhaite une interface de prise de commandes intuitive pour saisir efficacement les commandes sur place, assurant l'exactitude et une expérience client positive.

Critères d'acceptation :

- L'interface de prise de commandes affiche le menu avec des articles catégorisés pour une sélection rapide.
- Le personnel de service peut facilement ajouter des articles à la commande, spécifier les quantités et attribuer des commandes aux tables.
- L'interface permet l'ajout de demandes spéciales ou de modifications.

En tant que caissier, je souhaite un processus simplifié pour saisir les commandes à emporter et à livrer, y compris les détails de l'adresse et l'estimation du temps de livraison.

Critères d'acceptation :

- L'interface de commande comprend des options pour spécifier si la commande est à emporter ou à livrer.
- Les caissiers peuvent entrer les détails de l'adresse du client pour une livraison précise.
- Les estimations du temps de livraison sont calculées en fonction de la charge actuelle de la cuisine et de la distance de livraison.

## 6. Conclusion

En conclure que le stage de perfectionnement chez Aftercode a été une opportunité pour mettre en pratique les compétences acquises à l'Institut Supérieur d'Etudes Technologiques de Nabeul dans le développement de la plateforme Vitvent. Cette plateforme offre une solution complète et innovante pour les restaurateurs, leur permettant de gérer efficacement leurs commandes, leurs stocks et leur personnel. Le processus de développement a suivi la méthodologie Agile, avec l'utilisation du Product Backlog comme outil de gestion des exigences et des priorités. Dans les chapitres suivants, nous présenterons les différentes phases de la conception et de la réalisation de la plateforme Vitvent.

## Chapitre II. Étude Conceptuelle

### Introduction

Dans ce chapitre, nous allons aborder l'étude conceptuelle, une phase essentielle avant le début du développement. Cette étape permet de comprendre la logique métier à développer et de présenter clairement les besoins du client ainsi que les principales fonctionnalités du système.

#### 1. Conception des interfaces

Dans le cadre de la conception de notre plateforme Vitvent, nous avons conçu plusieurs interfaces utilisateur. Ces interfaces ont été créées sur Figma par notre talentueuse équipe de design chez Aftercode. Elles sont conçues pour être intuitives et faciles à utiliser, permettant ainsi à l'administrateur, au caissier et au chef de se concentrer sur leurs rôles respectifs.

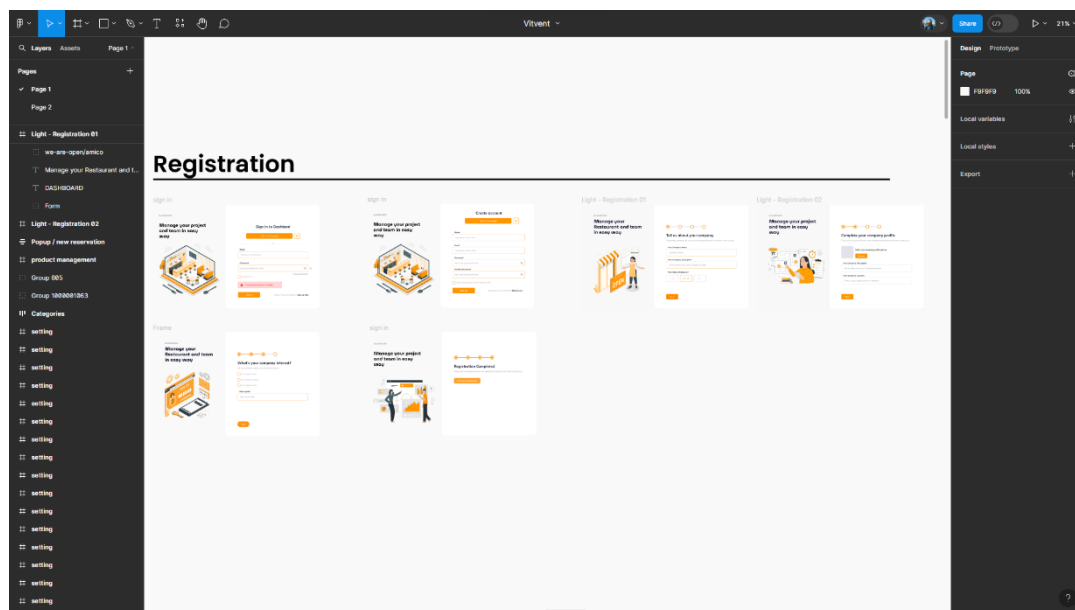


Figure 3: Pages d'Authentification - Design Figma.

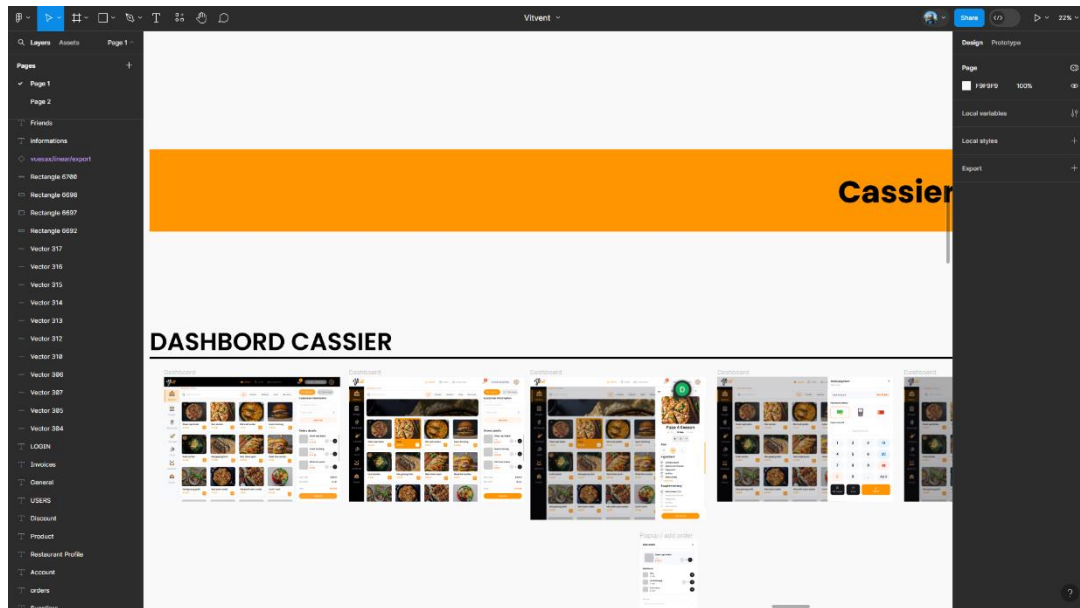


Figure 4: Pages Cassier - Design Figma.

## 2. Conception UML

Grâce à l'outil [app.diagrams.net](https://app.diagrams.net), plusieurs schémas ont été créés pour illustrer ces concepts. Un diagramme de séquence a été élaboré pour l'authentification, permettant de visualiser les interactions entre les différents acteurs au cours de ce processus. Un diagramme de cas d'utilisation a également été créé pour résumer les fonctionnalités de cassier et de l'admin. Enfin, un diagramme de classes a été conçu pour représenter la structure du système.



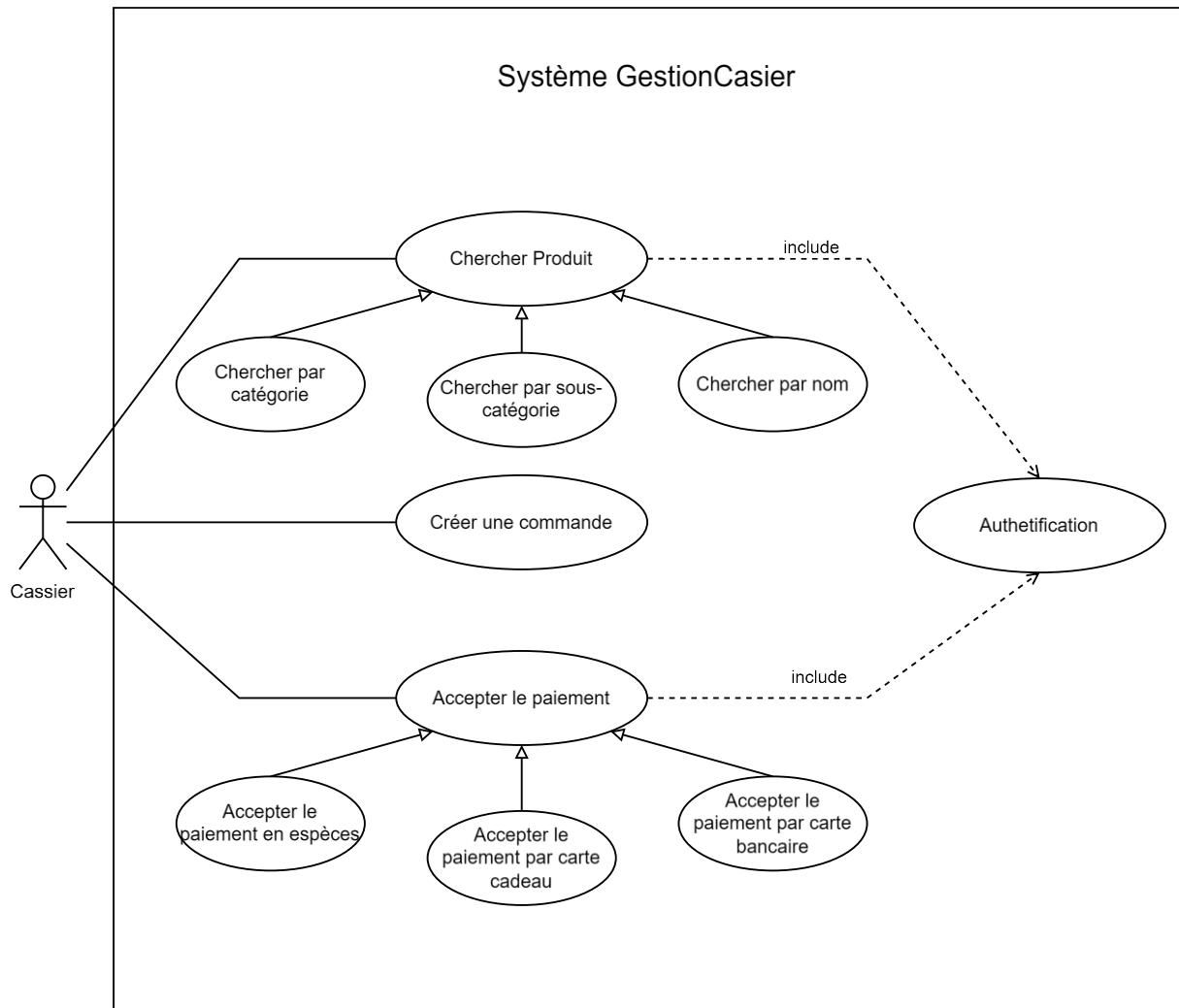


Figure 5: Digramme de cas d'utilisation – GestionCassier.

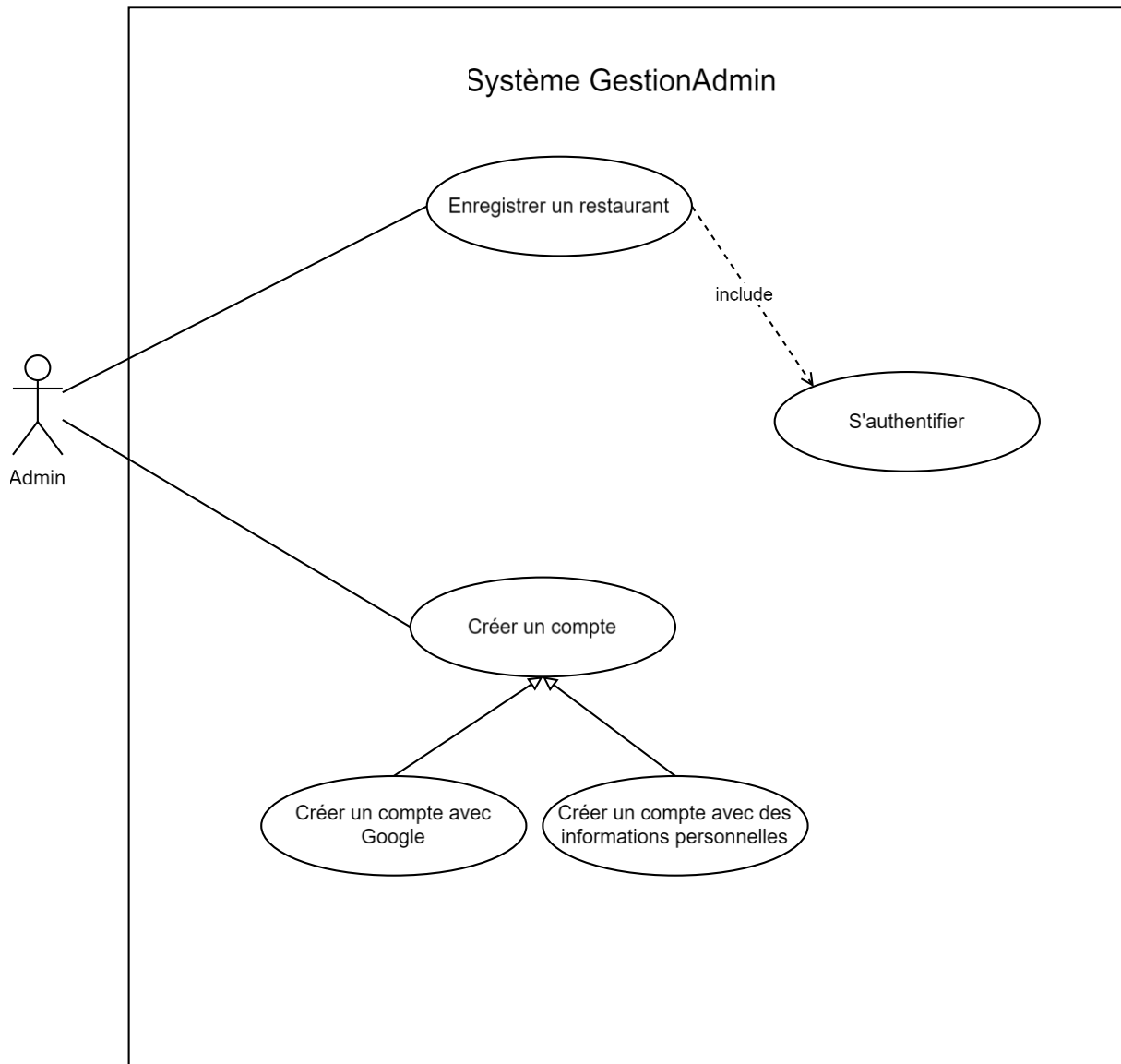


Figure 6: Diagramme de cas d'utilisation - GestionAdmin

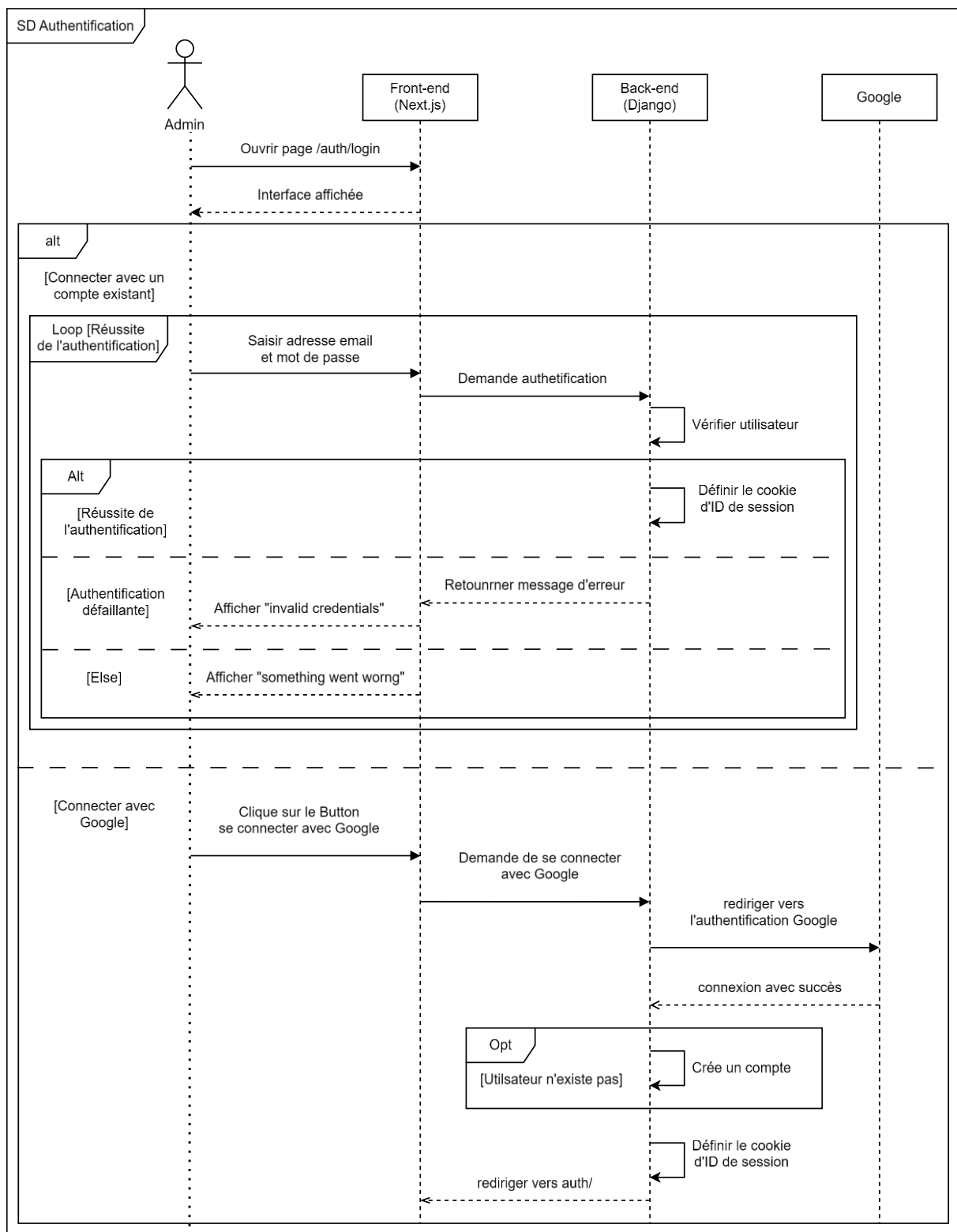


Figure 7: Diagramme de séquence pour l'authentification.

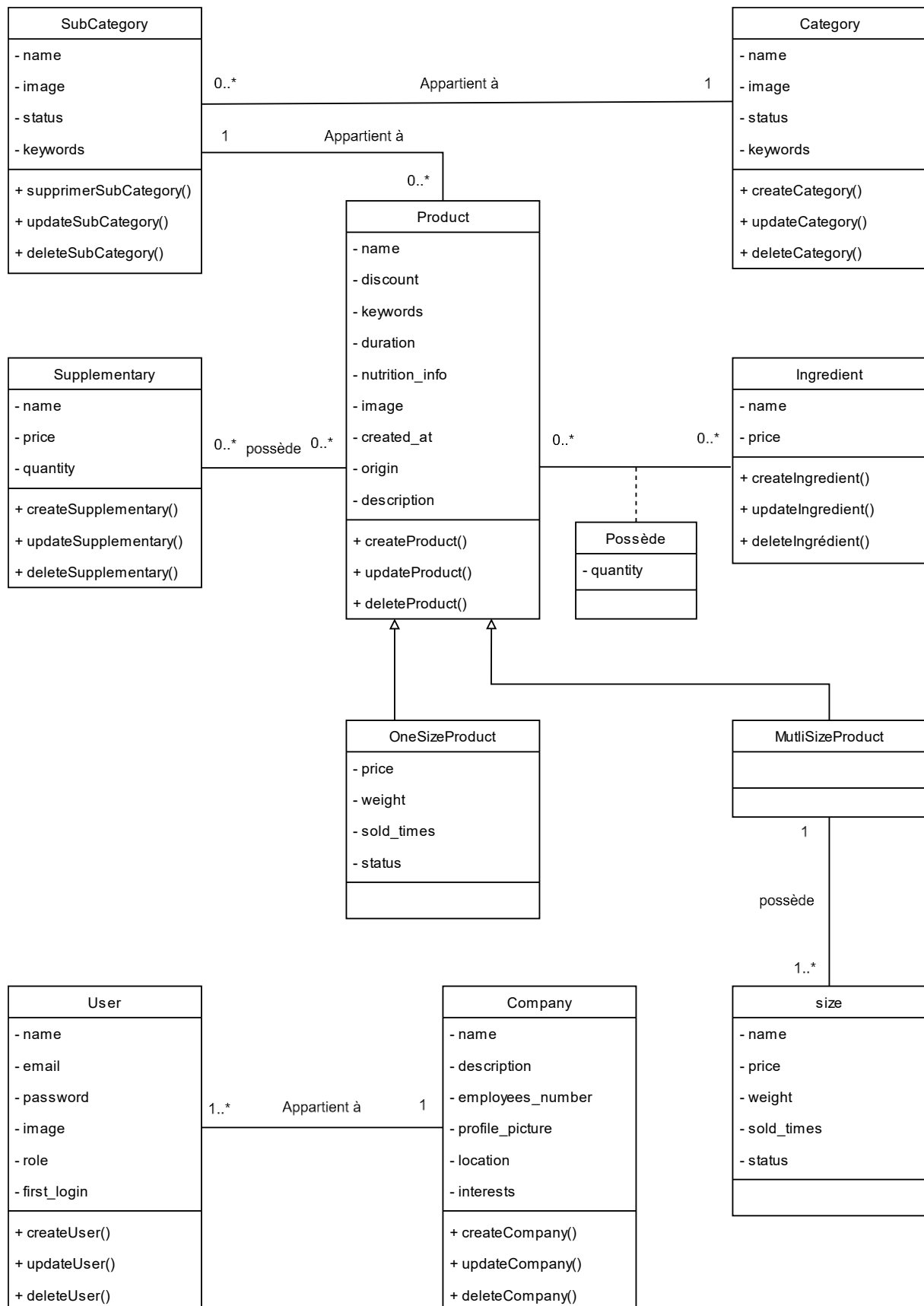


Figure 8: Diagramme de class

## Conclusion

En conclusion, ce chapitre a permis de conceptualiser les fonctionnalités de notre système à travers divers diagrammes. Ces illustrations ont aidé à comprendre les interactions, les fonctionnalités et la structure du système. Cette étape essentielle assure que nous répondons précisément aux besoins du client. Dans le prochain chapitre, nous commencerons le développement du système.

## Chapitre III. Réalisation

### Introduction

Après avoir achevé la phase de conception, nous entrons maintenant dans le vif du sujet avec la phase de réalisation. Dans ce chapitre, Nous commencerons par présenter notre environnement de développement, tant sur le plan matériel que logiciel. Par la suite, nous vous guiderons à travers les différentes interfaces de notre application web, en expliquant leur fonctionnement et leur rôle au sein du système.

#### 1. Environnement matériel

Au cours de ce stage, j'ai travaillé sur mon PC qui a les caractéristiques suivantes

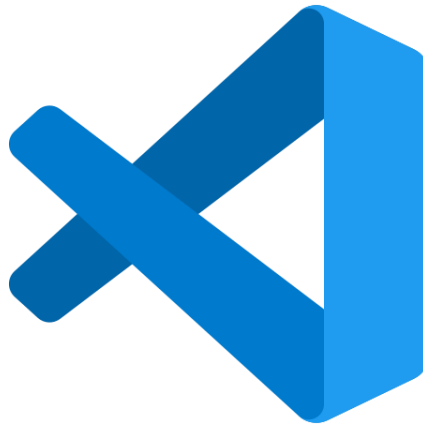
Caractère de l'équipement	Modèle de l'équipement
Marque	ASUS TUF F17
Écran (1)	17.3" Full HD, 144Hz
Écran (2)	DELL Full HD, 60Hz
Processeur	Intel Core i5-12500H (Up to 4.50 GHz Turbo max, 18Mo de mémoire cache, 12-Cores)
Système d'exploitation	Windows 11 Professionnel
Mémoire RAM	16 Go DDR4
Disque Dur	1 To SSD
Carte Graphique	NVIDIA GeForce RTX 3050 (4 Go de mémoire dédiée GDDR6)

#### 2. Les choix techniques

Pour le développement de la plateforme Vitvent, nous avons utilisé une combinaison d'outils et de technologies pour garantir une performance optimale, une facilité d'utilisation et une expérience utilisateur de qualité.

## 2.1. Visual Studio Code

Visual Studio Code (VS Code) est un choix optimal pour le développement de notre application. Il offre une prise en charge multilingue, une intégration Git transparente et une multitude d'extensions pour améliorer la productivité et la qualité du code. [1]



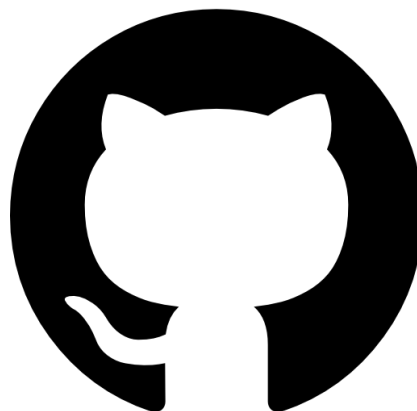
*Figure 9: Logo Visual Studio Code.*

## 2.2. GitHub et Git

GitHub et Git ont été choisis pour leur capacité à faciliter le travail collaboratif sur le code. Git est un système de contrôle de version distribué qui permet de suivre les modifications apportées au code au fil du temps, tandis que GitHub est une plateforme de développement qui héberge les dépôts Git et facilite le partage et la collaboration. Ces outils sont essentiels pour maintenir l'organisation et la cohérence dans un projet de développement logiciel. Ils permettent également une intégration continue et un déploiement continu, ce qui est crucial pour assurer la qualité du code et la rapidité de la livraison. [2][3]



*Figure 10: Logo Git.*



*Figure 11: Logo GitHub.*

### 2.3. Django

Django est un framework de développement web en Python qui suit le modèle de conception MVT (Modèle-Vue-Template). Il facilite la création d'applications web complexes et robustes. Django est connu pour sa rapidité de développement et sa sécurité intégrée. [4]



*Figure 12: Logo Django.*

### 2.4. Next.js

Next.js est un framework React qui permet le rendu côté serveur et la génération de sites statiques pour les applications web basées sur React. Il offre des fonctionnalités telles que le routage basé sur le système de fichiers et l'optimisation automatique. [5]

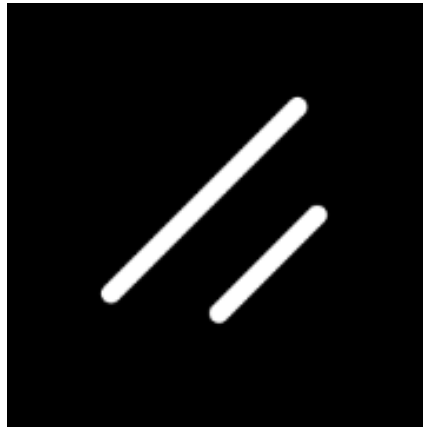


*Figure 13: Logo Next.js.*



## 2.5. Shadcn

Shadcn est une collection de composants React réutilisables, bien conçus et accessibles, pour construire des applications web modernes. Il n'est pas spécifiquement une bibliothèque de composants ou un framework UI, mais plutôt une compilation de composants réutilisables. Shadcn est construit sur Tailwind CSS et Radix UI, et est compatible avec Next.js, Gatsby, Remix, Astro, Laravel, et Vite. [6]



*Figure 14: Logo Shadcn.*

## 2.6. Framer Motion

Framer Motion est une bibliothèque de mouvement pour React qui alimente les animations et interactions dans Framer. Elle offre une syntaxe déclarative simple, gère des animations complexes, des gestes, des animations de défilement, des animations de sortie, et bien plus encore. [7]



*Figure 15: Logo Framer Motion.*

## 2.7. Tailwind CSS

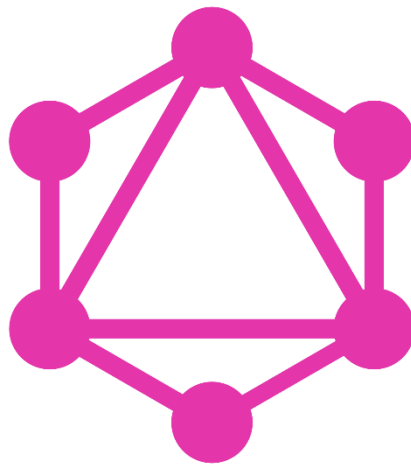
Tailwind CSS est un framework CSS open source qui fournit des classes utilitaires pour construire rapidement des interfaces utilisateur personnalisées. Contrairement à d'autres frameworks CSS, Tailwind ne fournit pas de classes prédéfinies pour des éléments spécifiques, mais permet de composer et d'agencer des styles directement dans votre balisage. [8]



*Figure 16: Logo Tailwind CSS.*

## 2.8. GraphQL

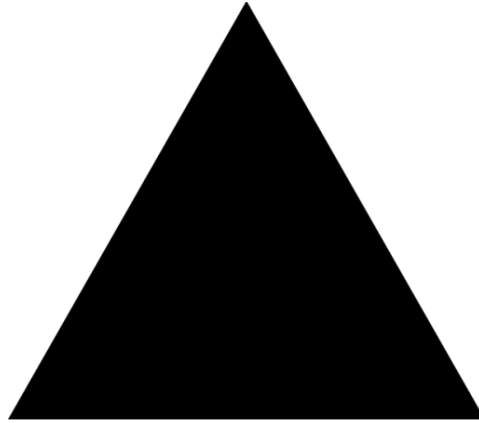
GraphQL est un langage de requête et d'exécution pour les API. Il permet aux clients de demander des données spécifiques à partir d'un serveur, contrairement aux API REST qui imposent des structures de données prédéfinies. [9]



*Figure 17: Logo GraphQL.*

## 2.9. Vercel

Vercel est une plateforme d'hébergement web qui offre une intégration transparente avec GitHub pour la mise en place de pipelines CI/CD. Elle permet un déploiement continu à partir de GitHub, facilitant ainsi le développement et le déploiement de prototypes. Vercel supporte plusieurs langages et frameworks, rendant le développement plus flexible. [10]



*Figure 18: Logo Vercel.*

## 2.10. Axios

Axios est une bibliothèque JavaScript populaire utilisée pour effectuer des requêtes HTTP. Elle offre une API facile à utiliser pour créer des requêtes HTTP vers des serveurs REST et effectuer des opérations CRUD. Axios est également capable de transformer les données de la requête et de la réponse, d'annuler les requêtes, et a un support automatique pour la protection contre les attaques XSRF. [11]



*Figure 19: Logo Axios.*

### 2.11. Redux

Redux est une bibliothèque JavaScript open-source pour la gestion et la centralisation de l'état de l'application. Il est le plus souvent utilisé avec des bibliothèques comme React ou Angular pour la construction d'interfaces utilisateur. Redux sert de magasin centralisé pour l'état qui doit être utilisé dans toute votre application, avec des règles garantissant que l'état ne peut être mis à jour que de manière prévisible. [12]



*Figure 20: Logo Redux.*

### 2.12. Slack

Slack est une plateforme de communication collaborative qui facilite la communication et la collaboration au sein des équipes. Il offre des fonctionnalités telles que la messagerie texte, le partage de fichiers et de médias, les appels vocaux et vidéo, et le chat de groupe pour la collaboration en équipe. [13]



*Figure 21: Logo Slack.*

### 2.13. PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle et objet open-source qui utilise et étend le SQL. Il est réputé pour sa robustesse, son intégrité des données, sa flexibilité et son extensibilité. [14]



*Figure 22: Logo PostgreSQL.*

### 2.14. Diagrams.net

App.diagrams.net est une application en ligne gratuite qui permet de créer des diagrammes et des organigrammes. Elle est intégrée à Google Drive, ce qui facilite la création et le partage de diagrammes. Elle supporte une variété de types de diagrammes, y compris les diagrammes de flux, UML, ERD, et les maquettes d'interface. [15]



*Figure 23: Logo Diagrams.net.*

### 2.15. Figma

Figma est une plateforme de conception d'interfaces utilisateur basée sur le Web qui met l'accent sur la collaboration en temps réel. Il est largement utilisé pour créer, partager et tester des designs pour divers produits numériques. [16]



Figure 24: Logo Figma.

### 2.16. Redux DevTools

Redux DevTools est un outil de développement pour Redux qui offre des fonctionnalités pour votre flux de travail de développement Redux. Il permet d'inspecter chaque état et payload d'action. Il peut être utilisé comme une extension de navigateur, une application autonome ou un composant React intégré dans l'application client. [17]



Figure 25: Logo Redux DevTools.

### 2.17. Neon

Neon est un service d'hébergement de bases de données Postgres sans serveur. Il utilise la technique “copy-on-write” pour offrir des fonctionnalités modernes telles que le branching, le checkpointing en ligne, et la restauration à un moment donné. Neon est conçu pour être évolutif, efficace en termes de coûts et facile à utiliser. [18]



*Figure 26: Logo Neon*

### 3. Les interfaces réalisées

Dans les pages suivantes, nous allons décrire en détail les interfaces réalisées et expliquerons la méthodologie de leur réalisation.

La capture d'écran ci-dessous illustre mon environnement de travail Django où j'ai créé un projet, installé les paquets nécessaires, développé mes points de terminaison API, mis en place un fichier ".env" pour la configuration du backend et la sécurisation de mes clés, et créé les modèles pour la base de données.

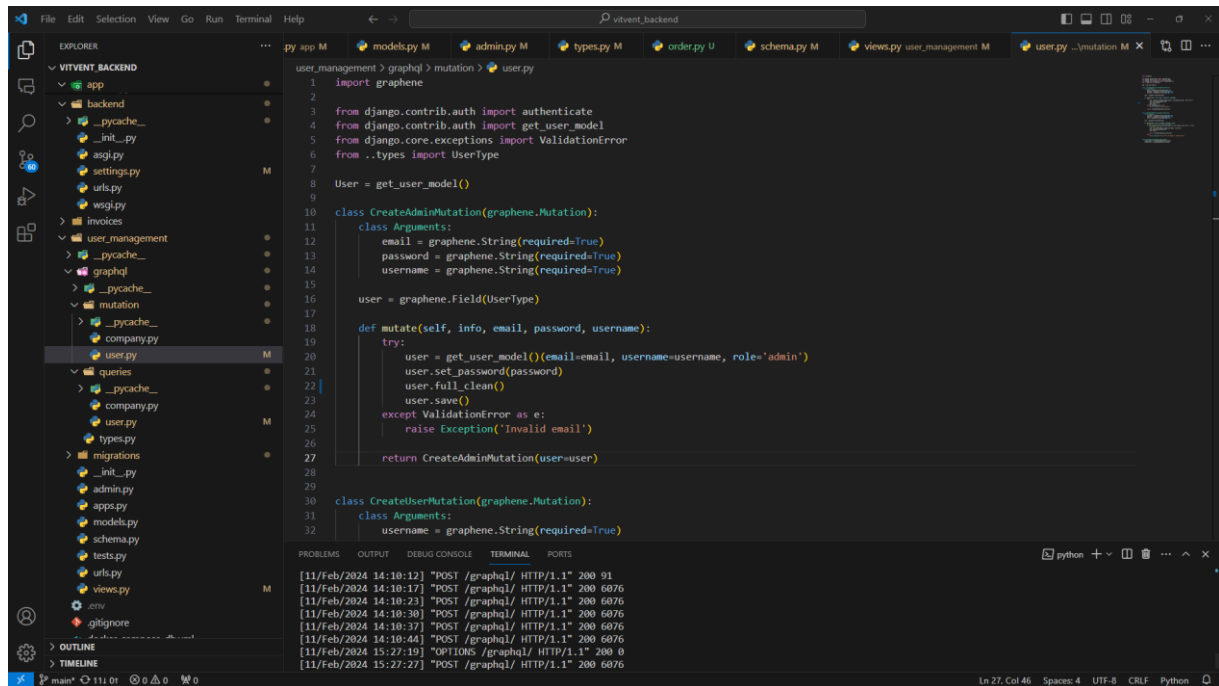


Figure 27: Environnement de travail backend avec Django et VS Code.



La capture d'écran ci-dessous illustre mon environnement de travail Next.js. Dans ce cadre, j'ai créé un projet, installé les paquets nécessaires et développé les pages conformément au design Figma. J'ai travaillé sur plusieurs fichiers simultanément, comme le montrent les multiples onglets ouverts dans mon éditeur de code. De plus, j'ai utilisé un serveur de développement local pour tester et visualiser mon travail en temps réel.

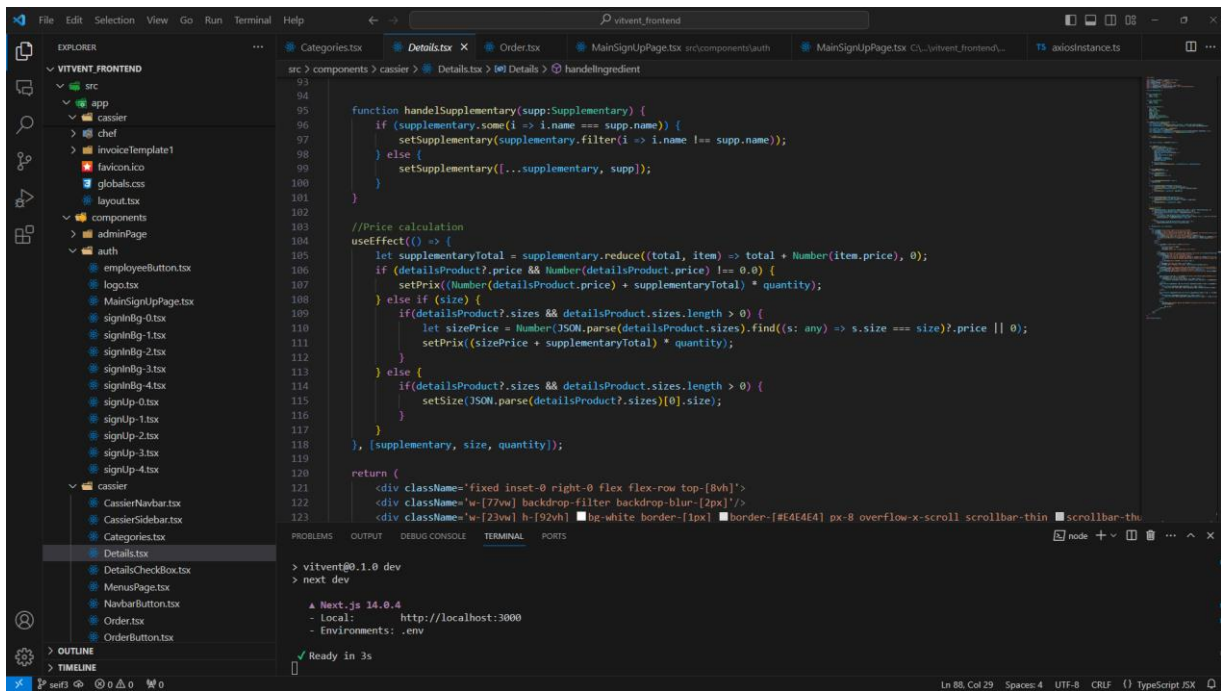


Figure 28: Environnement de travail frontend avec Next.js et VS Code.

La capture d'écran ci-dessous montre le panneau d'administration Django. Cet outil m'a aidé à gérer les données de l'application. J'ai pu voir les données en direct, ajouter de nouvelles informations, supprimer ce qui n'est plus nécessaire et changer les données existantes.

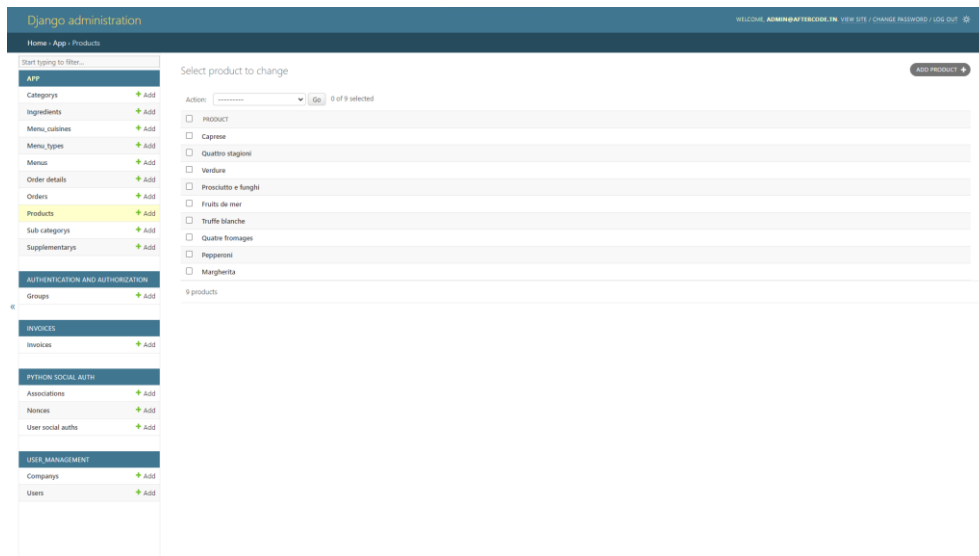


Figure 29: Panel Admin Django.

La capture d'écran ci-dessous montre l'interface du Playground de l'API GraphQL. C'est un outil que j'ai utilisé pour tester mes requêtes et mutations. Comme vous pouvez le voir, j'ai écrit une requête sur le côté gauche de l'écran. Après avoir exécuté la requête, les résultats sont affichés dans un format structuré sur le côté droit de l'écran. Cela m'a permis de vérifier et de valider les réponses de l'API en temps réel.

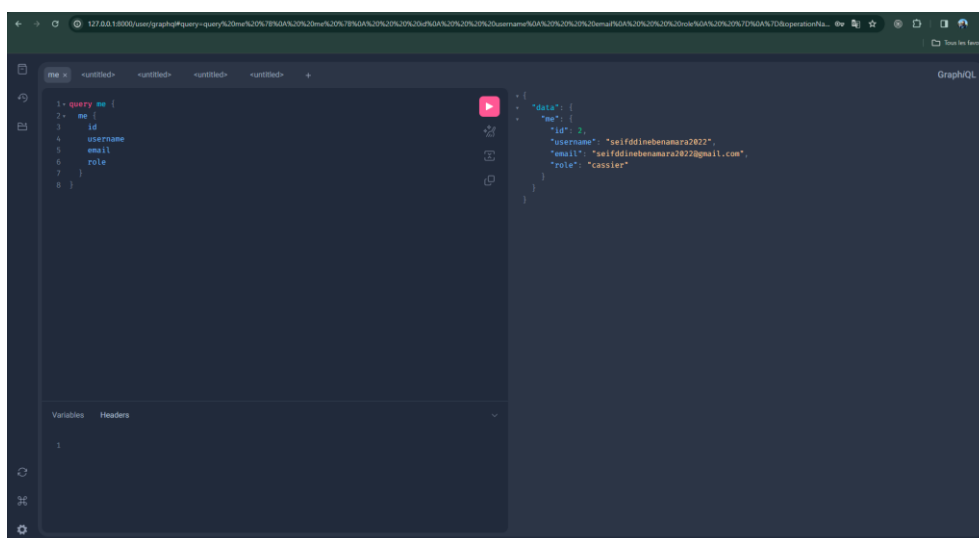


Figure 30: Interface du Playground de l'API GraphQL.

La capture d'écran ci-dessous illustre l'interface GitHub une fois qu'un conflit de fusion a été résolu. Ou j'attends que le processus d'intégration continue/déploiement continu (CI/CD) de Vercel effectue les vérifications nécessaires. Un signe d'avertissement jaune indique que des vérifications sont en cours.

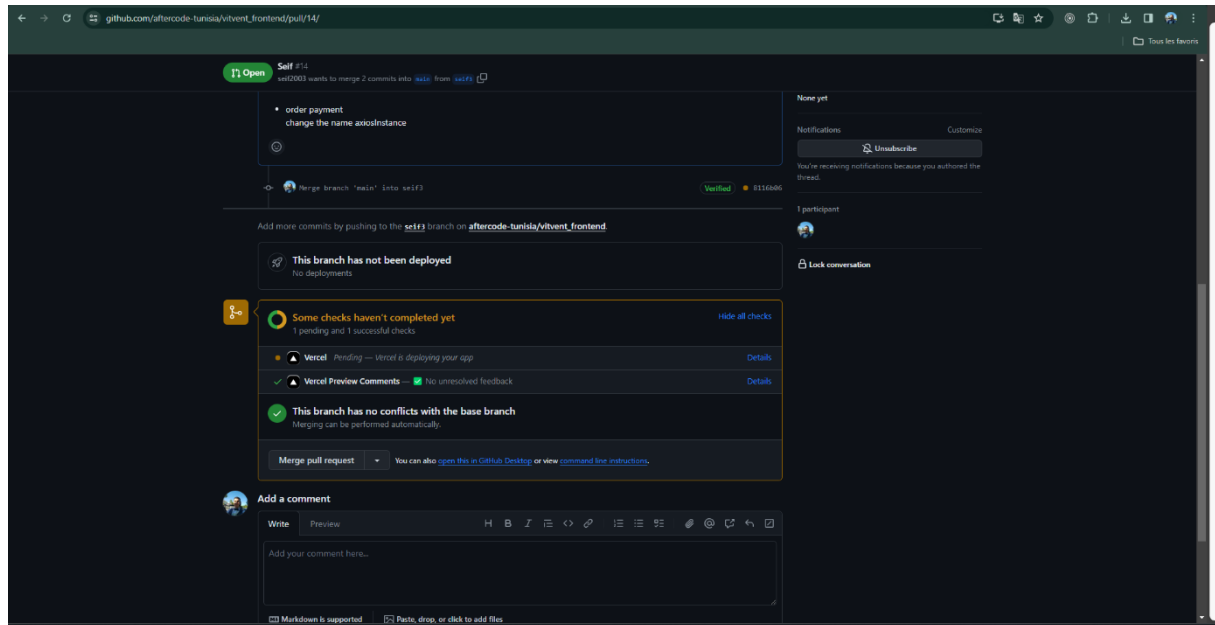


Figure 31: Fusion des dépôts sur GitHub 1.

La capture d'écran ci-dessous montre que la demande de fusion a été fusionnée et fermée avec succès sur GitHub. Cela signifie que les modifications que j'ai proposées ont été acceptées et intégrées dans la branche principale du projet.

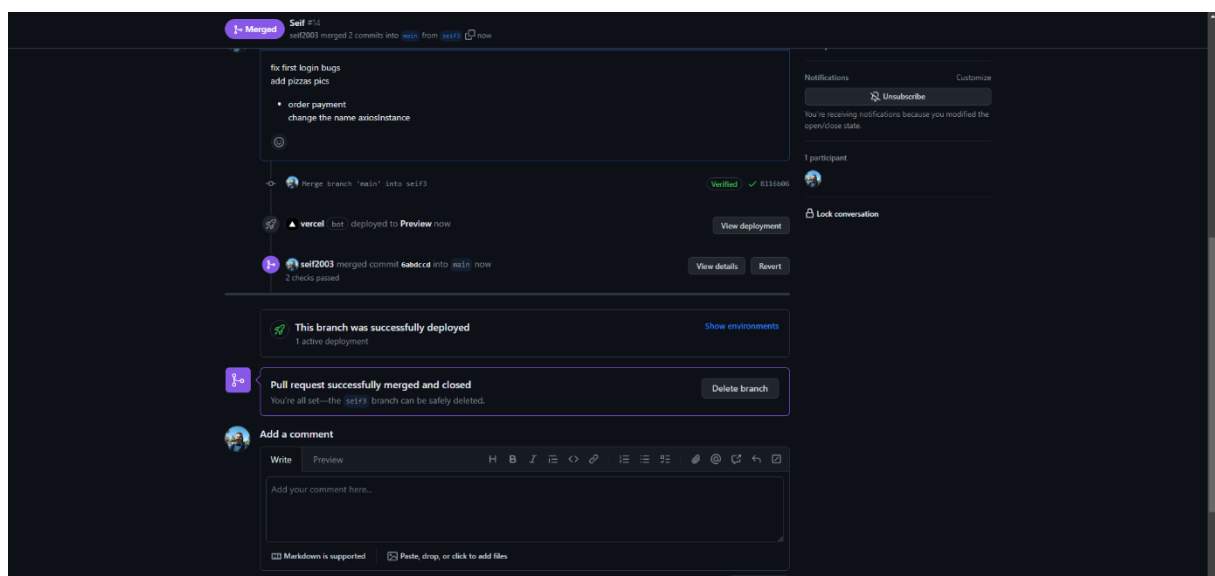


Figure 32: Fusion des dépôts sur GitHub 2.

Gmail

Rechercher dans les messages

Nouveau message

Boîte de réception 602

Messagerie suivies

En attente

Messages envoyés

Brouillons 43

Plus

Libellés +

< [aftercode-tunisia/vitvent\_frontend] Seif (PR #14) >

Boîte de réception x

vercel[bot] <notifications@github.com> 13:55 (il y a 7 heures)

À aftercode-tunisia/vitvent\_frontend, moi, Author ▾

The latest updates on your projects. Learn more about [Vercel for Git](#).

Name	Status	Preview	Updated (UTC)
vitvent-frontent	Ready (Inspect)	<a href="#">Visit Preview</a>	Feb 11, 2024 0:55am

—

Reply to this email directly, [view it on GitHub](#), or [unsubscribe](#).  
You are receiving this because you authored the thread.

Répondre Répondre à tous Transférer

La capture d'écran ci-dessous montre un signe vert sur le dépôt GitHub, indiquant que tous les contrôles ont été passés avec succès.



Voici des captures d'écran des interfaces que j'ai créées en utilisant Next.js, Framer Motion et Shaden tout en suivant le design figma

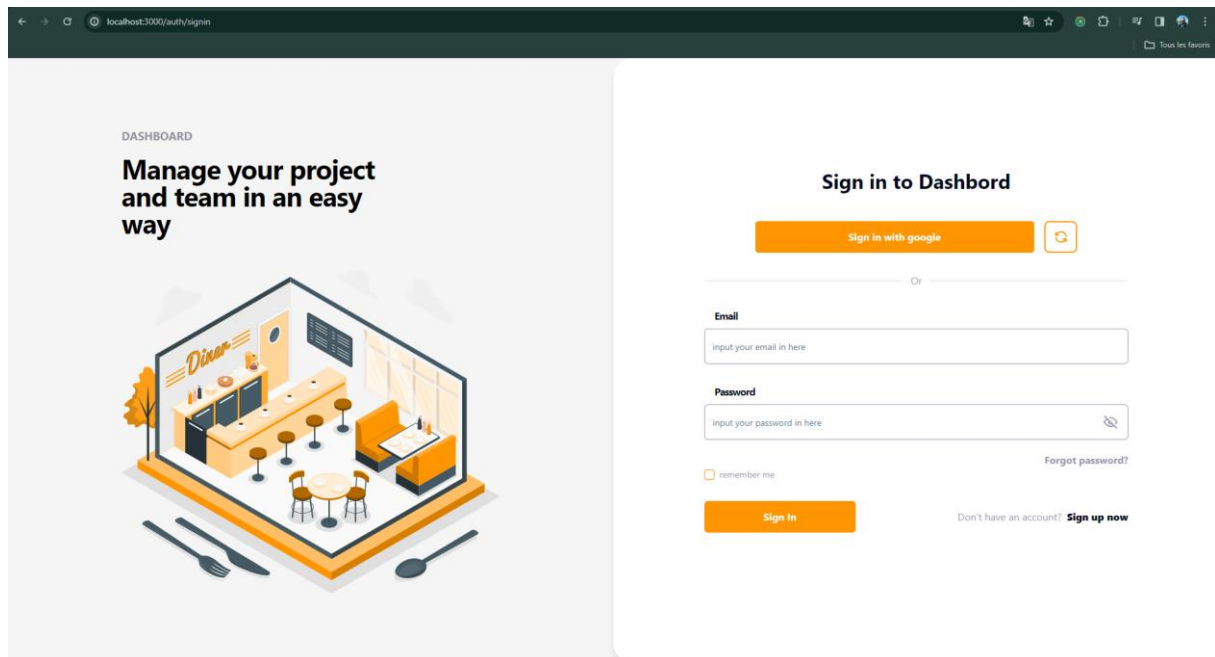


Figure 35: Page de Connexion.

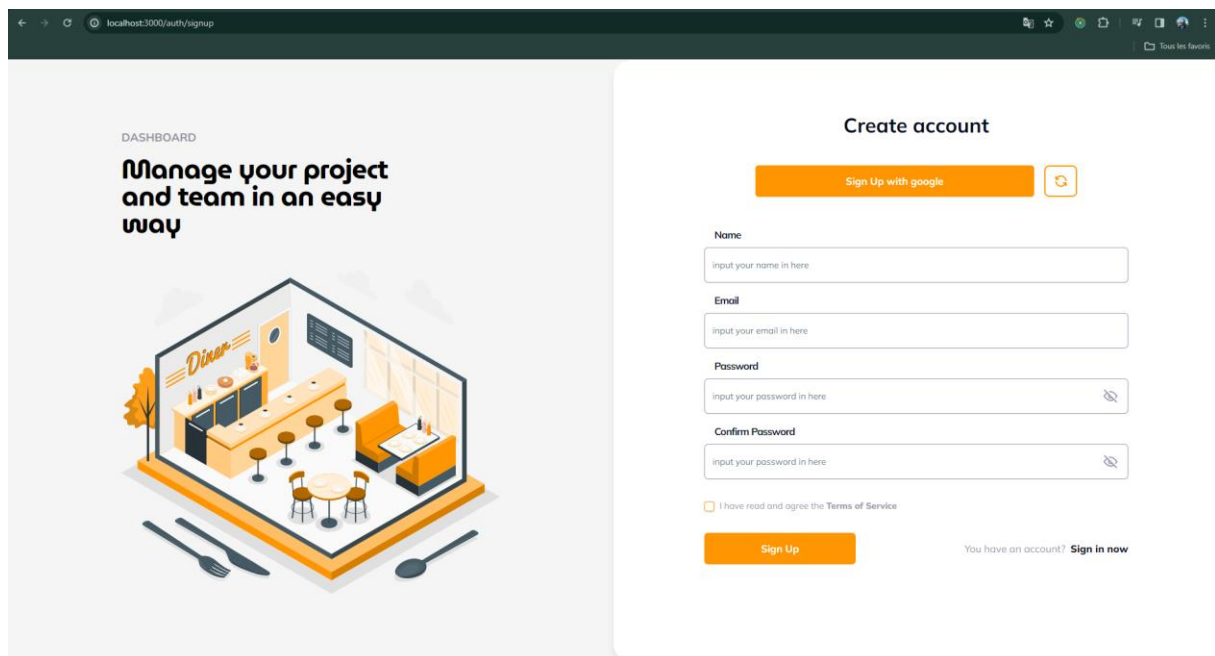


Figure 36: Page de Création de Compte.

DASHBOARD

## Manage your project and team in an easy way

Welcome!

**Tell us about your company**

Thousands company like yours are manage their project and team in easy way

Your Company Name

Company name

Your Company Description

Tell me about what your Company look like

How Many Employees?

1-10 20-30 50+

Next

Figure 37: Page d'enregistrement du Restaurant - Étape 1.

DASHBOARD

## Manage your project and team in an easy way

**Complete your company profile**

Thousands company like yours are manage their project and team in easy way

Select your company profile picture

Browse

Your Company Location

Where is your company located

Next

Figure 38: Page d'enregistrement du Restaurant - Étape 2.

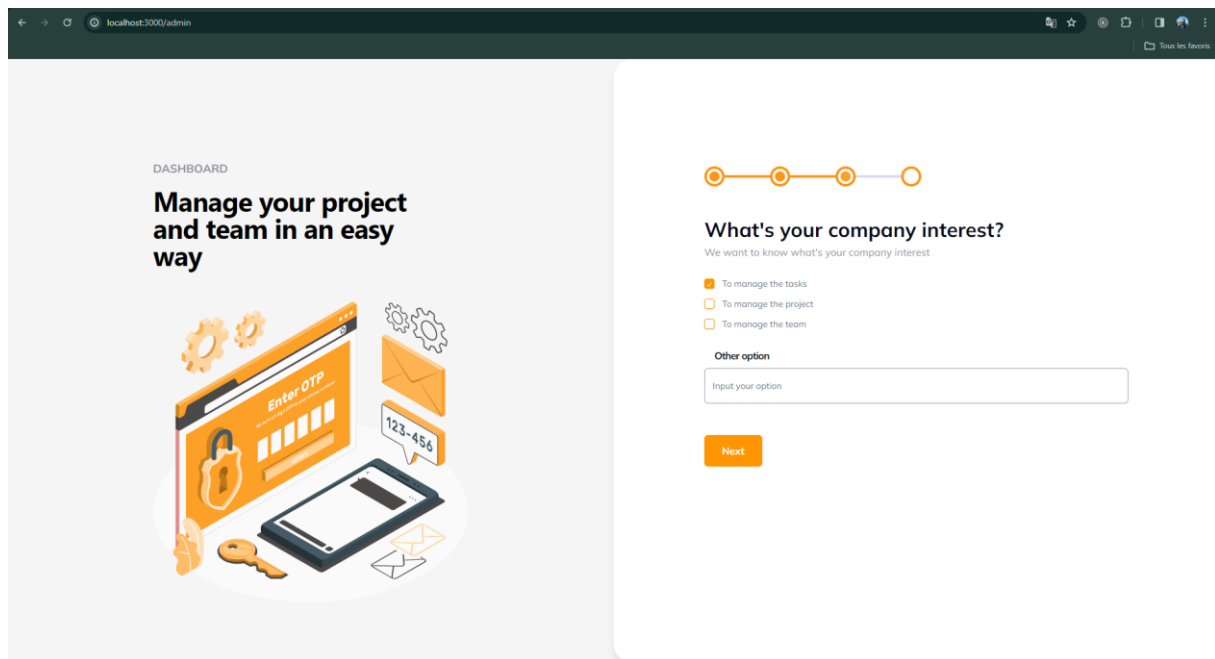


Figure 39: Page d'enregistrement du Restaurant - Étape 3.

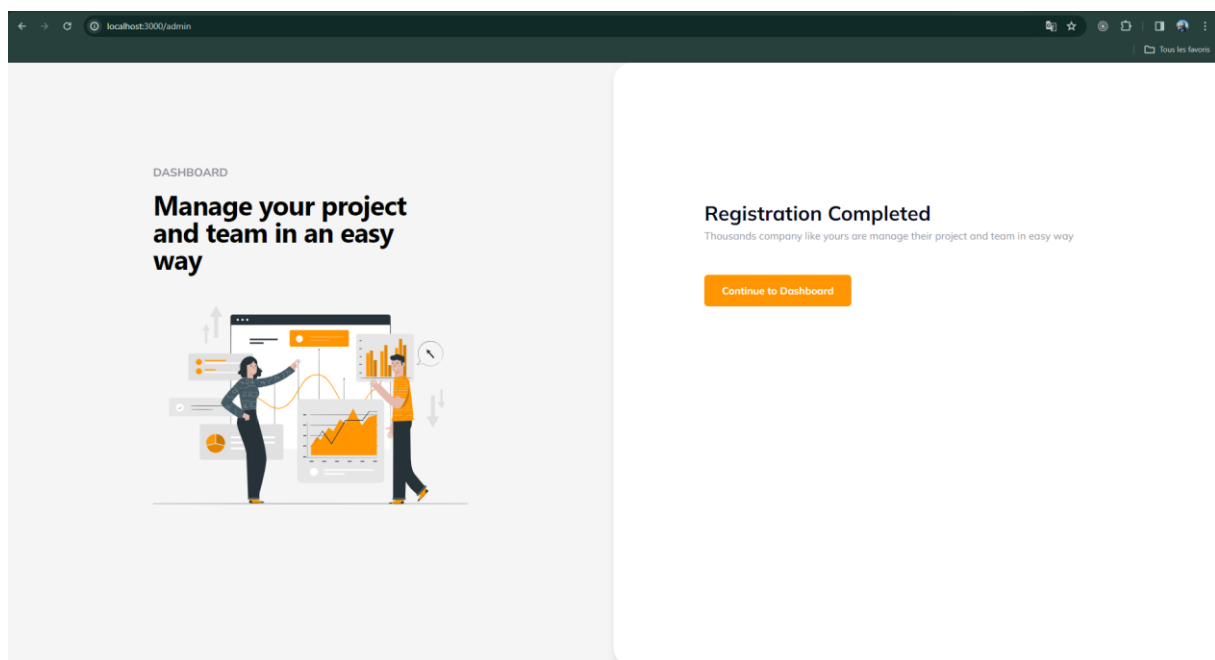


Figure 40: Page d'enregistrement du Restaurant - Étape 4.

La capture d'écran ci-dessous illustre l'interface de la console.cloud.google. Elle démontre la création d'un projet et la mise en place de l'authentification Google. Cette configuration permet à nos utilisateurs de se connecter en utilisant leurs comptes Google.

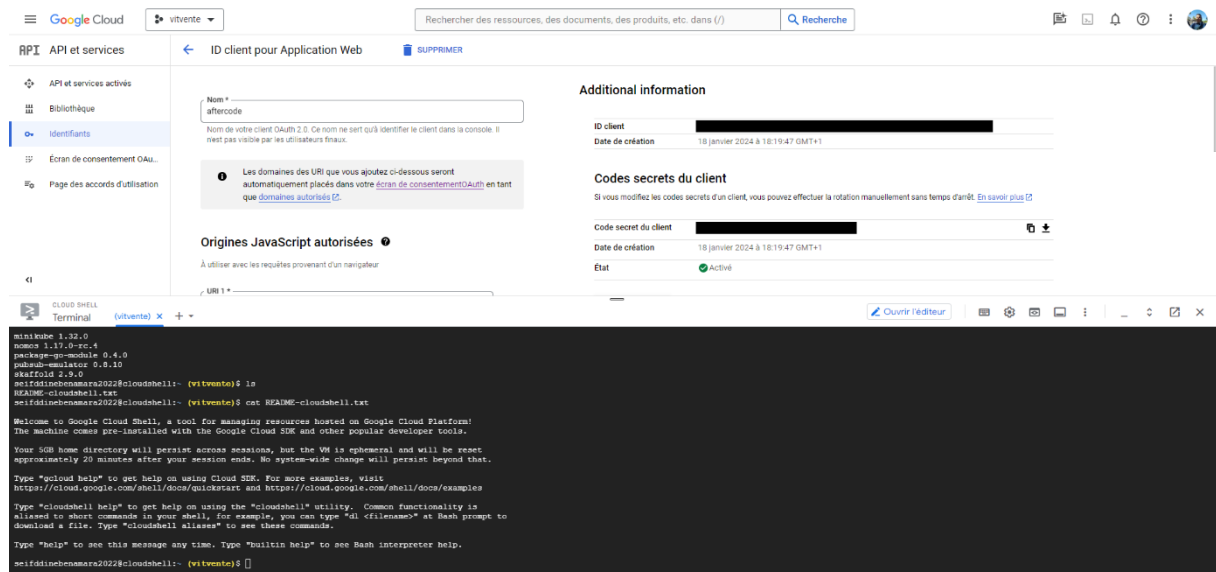


Figure 41: Interface de la console Google Cloud.

Après la mise en place du système d'authentification, j'ai développé le panneau du caissier. Voici des captures d'écran présentant le panneau caissier.

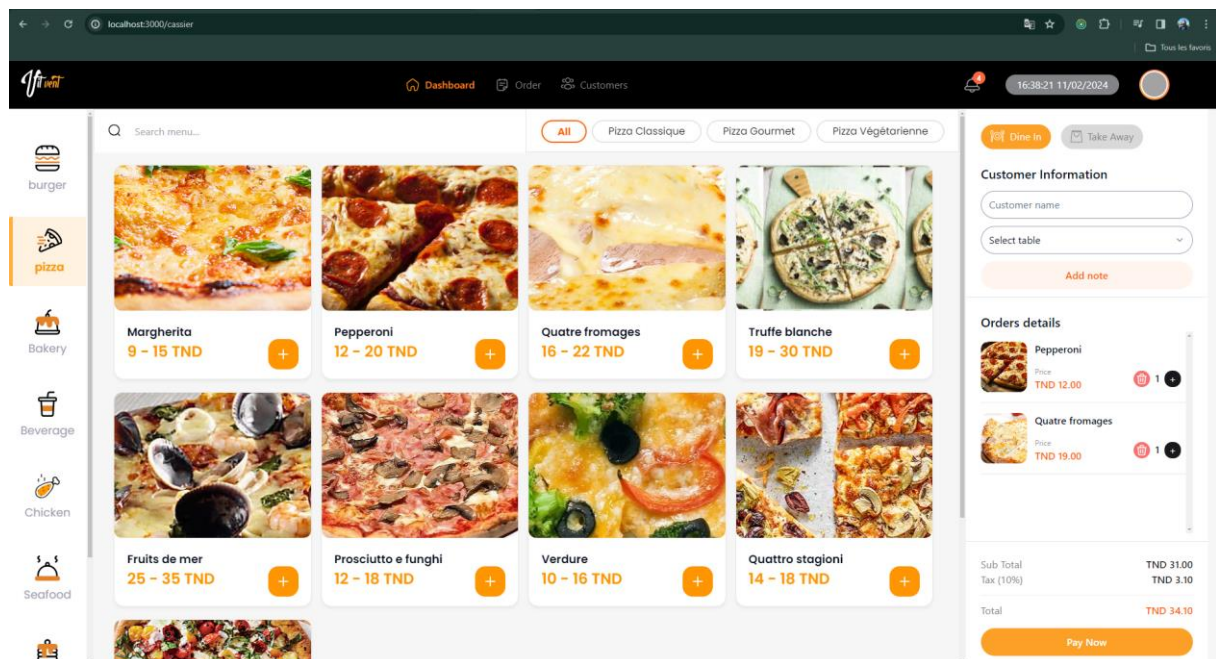


Figure 42: Page de caissier.



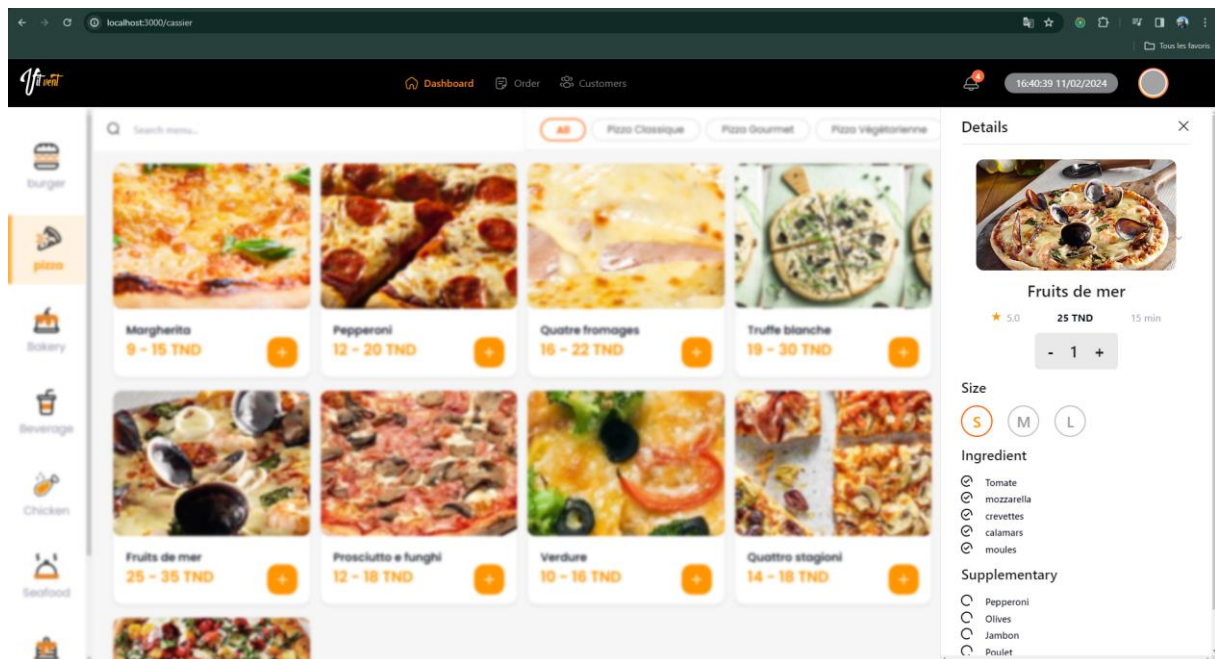


Figure 43: Onglet 'Détails du produit'.

L'utilisation de Redux est essentielle pour gérer l'état entre les composants. Voici une capture d'écran illustrant l'architecture des états dans Redux DevTools.

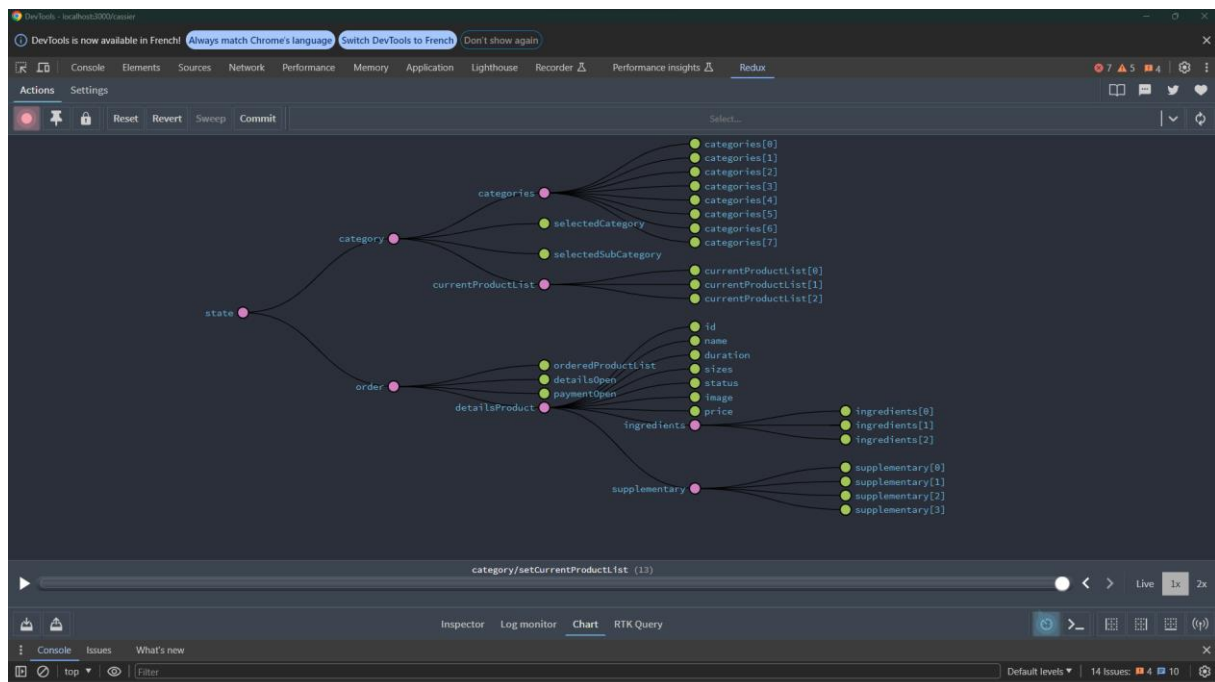


Figure 44: Utilisation de Redux DevTools lors du développement.

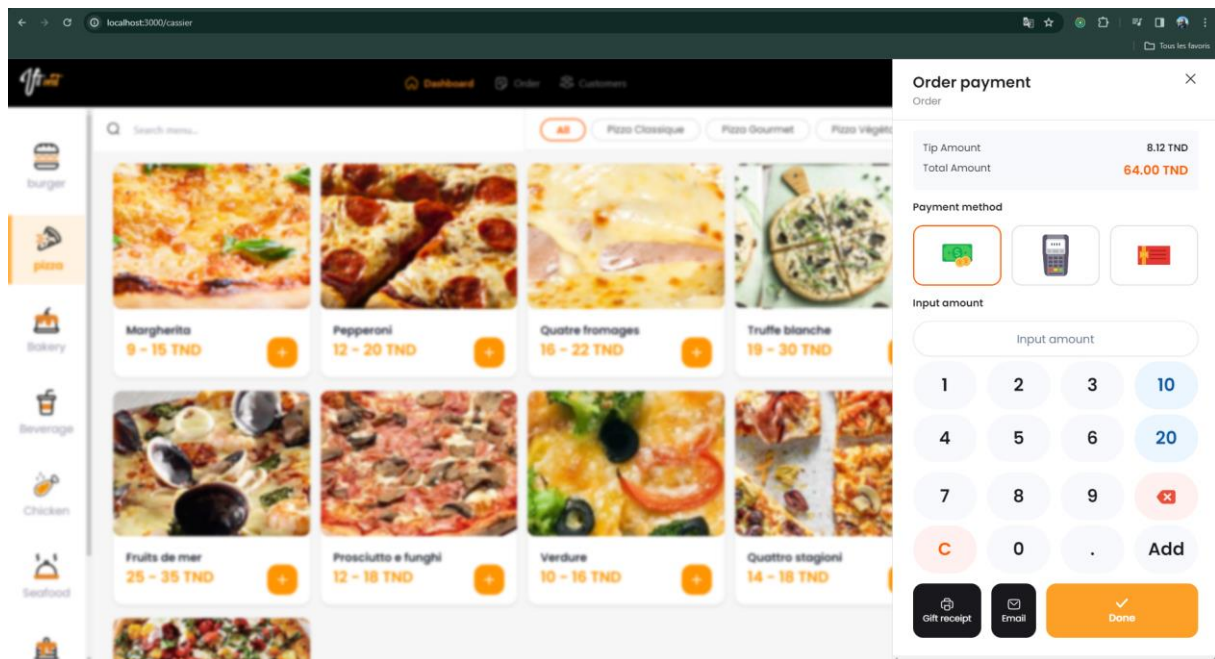


Figure 45: Onglet 'Paiement de l'ordre'.

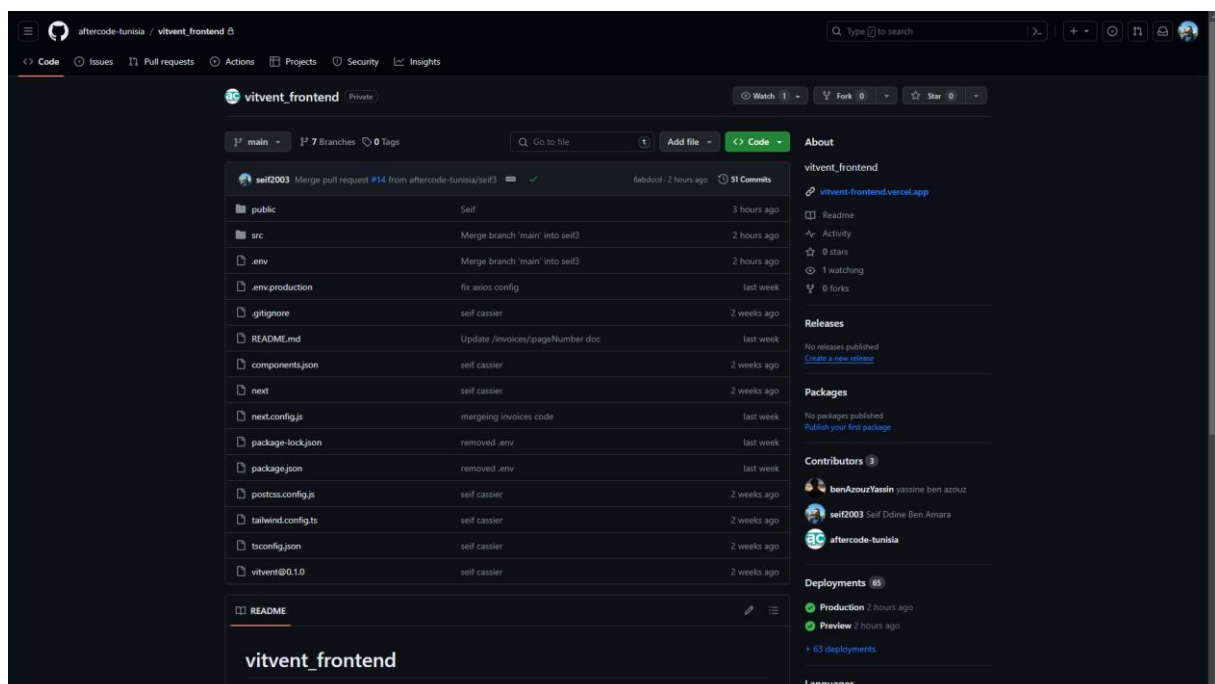


Figure 46: Vue du dépôt Front-End sur GitHub.

Cette image montre l'interface Neon, où la table 'app\_product' est affichée. Cette table contient des détails sur différents produits.

#	id	name	sizes	discount	keywords	duration	nutrition_info	image	created_at
1	2	Margherita	["size": "S", "price": "9"], ["size": "M", "price": "11"], ["size": "L", "price": "15"]	0.00		0		/assets/menuPage/pizza-margherita.jpg	2024-02-11 12:28:40.180958+00
2	3	Pepperoni	["size": "S", "price": "12"], ["size": "M", "price": "15"], ["size": "L", "price": "20"]	0.00		0		/assets/menuPage/pizza-pepperoni.jpg	2024-02-11 12:31:05.689583+00
3	4	Quatre fromages	["size": "S", "price": "16"], ["size": "M", "price": "18"], ["size": "L", "price": "22"]	0.00		0		/assets/menuPage/pizza-4-fromages.jpg	2024-02-11 12:32:49.046711+00
4	5	Truffe blanche	["size": "S", "price": "18"], ["size": "M", "price": "23"], ["size": "L", "price": "30"]	0.00	Truffe blanche	0		/assets/menuPage/truffeblanche.jpg	2024-02-11 12:35:03.135551+00
5	7	Prosciutto e funghi	["size": "S", "price": "12"], ["size": "M", "price": "15"], ["size": "L", "price": "18"]	0.00		0		/assets/menuPage/Prosciutto.jpg	2024-02-11 12:40:19.324024+00
6	8	Verdure	["size": "S", "price": "10"], ["size": "M", "price": "12"], ["size": "L", "price": "16"]	0.00		0		/assets/menuPage/Verdure.jpg	2024-02-11 12:44:12.904744+00
7	9	Quattro stagioni	["size": "S", "price": "14"], ["size": "M", "price": "18"], ["size": "L", "price": "18"]	0.00		0		/assets/menuPage/Quattro_stagioni.jpg	2024-02-11 12:45:56.195882+00
8	10	Caprese	["size": "S", "price": "18"], ["size": "M", "price": "22"], ["size": "L", "price": "25"]	0.00		0		/assets/menuPage/caprese-pizza.jpg	2024-02-11 12:48:18.807824+00
9	6	Fruits de mer	["size": "S", "price": "25"], ["size": "M", "price": "30"], ["size": "L", "price": "35"]	0.00		15		/assets/menuPage/pizza-fruits-de-mer.webp	2024-02-11 12:37:05.399814+00

Figure 47: Table de produit dans la base de données.

## Conclusion

Ce chapitre a mis en lumière la réalisation des interfaces pour le projet VitVent, ainsi que l'application efficace des technologies pertinentes. En tant que contributeur à ce projet, je suis fier de l'impact significatif que j'ai pu apporter à sa réalisation.

## Conclusion générale

En conclusion, ce stage chez Aftercode a été une expérience enrichissante et formatrice. Il m'a permis de mettre en pratique mes compétences acquises à l'Institut Supérieur d'Etudes Technologiques de Nabeul, tout en me donnant l'opportunité d'apprendre de nouvelles techniques et méthodologies.

J'ai également eu l'occasion d'apprécier l'importance du travail en équipe et de la communication efficace dans la réalisation d'un projet technologique. Chaque membre de l'équipe avait un rôle crucial à jouer et notre capacité à travailler ensemble a été un facteur clé de notre succès.

Enfin, ce stage a renforcé ma passion pour le développement de logiciels et m'a confirmé que je souhaite poursuivre ma carrière dans ce domaine. Je suis impatient de relever de nouveaux défis et de continuer à apprendre et à grandir en tant que développeur.

Je tiens à remercier toute l'équipe d'Aftercode pour leur soutien et leurs conseils tout au long de ce stage. Leur expertise et leur dévouement ont été une source d'inspiration constante. Je suis fier d'avoir contribué à la réalisation de la plateforme Vitvent et j'ai hâte de voir comment elle aidera les restaurateurs à optimiser leur service et à augmenter leur chiffre d'affaires.

## Webographie

- [Web 1]: <https://code.visualstudio.com/docs>
- [Web 2]: <https://git-scm.com/video/what-is-git>
- [Web 3]: <https://docs.github.com/fr>
- [Web 4]: <https://docs.djangoproject.com/en/5.0/>
- [Web 5]: <https://nextjs.org/docs>
- [Web 6]: <https://ui.shadcn.com/docs>
- [Web 7]: <https://www.framer.com/motion/>
- [Web 8]: <https://tailwindcss.com/docs>
- [Web 9]: <https://graphql.org/learn/>
- [Web 10]: <https://vercel.com/docs>
- [Web 11]: <https://axios-http.com/docs/intro>
- [Web 12]: <https://redux.js.org/introduction/why-rtk-is-redux-today>
- [Web 13]: <https://slack.com/intl/fr-fr/solutions/information-technology>
- [Web 14]: <https://www.postgresql.org/docs/>
- [Web 15]: <https://www.blogdumoderateur.com/tools/diagrams-net-draw-io/>
- [Web 16]: <https://help.figma.com/hc/en-us>
- [Web 17]: <https://github.com/reduxjs/redux-devtools>
- [Web 18]: <https://neon.tech/docs/introduction>