|  |
| --- |
| Ecole Marocaines des Sciences de l’Ingénieur  Département Génie Informatique  Rapport de Projet de POO  Gestion d'une Cafétéria  Durant la période : du 01/03/2023 a 03/04/2024 |

Encadrant

Pr. AMEKSA Mohammed

Etudiants

Ouhmida Soulaimane

Ettakaddoumi Hamza

Targui Hajar

Kelladi Fatima ezzahra

# Dedicase

À mes collègues et partenaires de travail, pour leur collaboration et leur engagement tout au long de ce projet. Vos idées, votre soutien et votre camaraderie ont été essentiels pour surmonter les défis rencontrés en cours de route.

À ma famille, pour leur soutien inconditionnel et leur compréhension pendant les longues heures de travail et les moments de stress. Votre amour et votre soutien sont ma source d'inspiration quotidienne.

À toutes les personnes impliquées dans ce projet, qu'ils soient mentionnés ici ou non, je vous suis reconnaissant pour votre contribution et votre dévouement. Ce projet n'aurait pas été possible sans votre collaboration.

À ceux qui croient en l'importance de l'innovation et du progrès, cette dédicace est pour vous. Votre soutien à la recherche de nouvelles idées et de solutions créatives est ce qui alimente notre motivation à poursuivre.

# Remerciement

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué à la réalisation de ce projet. Tout d'abord, je souhaite remercier **Pr. AMEKSA Mohammed** pour sa direction et ses conseils précieux tout au long du processus. Sa vision et son expertise ont été des éléments essentiels pour atteindre nos objectifs.

Mes remerciements vont également à notre membre de l'équipe pour leur dévouement, leur travail acharné et leur collaboration exceptionnelle. Chacun de vous a apporté une contribution unique qui a enrichi notre projet et a permis d'atteindre les résultats que nous avons obtenus.

Ce projet a été une expérience enrichissante grâce à la contribution de chacun. Je suis reconnaissant d'avoir eu l'opportunité de travailler avec une équipe aussi talentueuse et dévouée. Merci à tous pour votre engagement et votre travail acharné.

# Abstract

Ce rapport présente la conception et l'implémentation d'une plateforme de commerce intégrant un chatbot intelligent. Dans l'introduction, les objectifs du projet sont exposés, mettant en lumière la nécessité d'améliorer l'expérience utilisateur et l'engagement client dans le commerce en ligne. La méthode décrit le processus de développement, incluant la sélection des technologies, la conception des interfaces utilisateur et la formation du chatbot. Les résultats montrent une amélioration significative de l'expérience utilisateur, avec une augmentation notable des taux de conversion et de rétention client. En conclusion, cette plateforme offre une solution innovante pour le commerce en ligne, ouvrant la voie à de nouvelles possibilités d'interaction et de fidélisation client grâce à l'intégration d'un chatbot intelligent.

# Résumé

Ce rapport de projet expose le développement d'une plateforme de commerce sophistiquée intégrant un chatbot intelligent. La plateforme vise à rationaliser l'expérience utilisateur et à améliorer l'engagement client en exploitant des technologies de pointe. Grâce à une conception et une mise en œuvre méticuleuses, l'équipe a créé une solution évolutive qui associe des interfaces utilisateur intuitives à des capacités conversationnelles avancées. Ce rapport met en évidence les principales fonctionnalités, les considérations de conception, les détails de mise en œuvre et les perspectives futures de la plateforme, soulignant son potentiel de révolutionner le commerce en ligne.

# Glossaire

|  |  |
| --- | --- |
| Mot | Signification |
| CRUD | CRUD est un acronyme issu du monde de la programmation informatique et fait référence aux quatre fonctions considérées comme nécessaires pour mettre en œuvre une application de stockage persistant : créer, lire, mettre à jour et supprimer. |
| EMSI | École Marocaine des Sciences de l'Ingénieur |
| TIC | Technologies de l'information et de la communication |
| REST | ne interface de programmation d'application (API ou API web) qui respecte les contraintes du style d'architecture REST et permet d'interagir avec les services web RESTful. |
| API | Application programming interface ou interface de programmation d'application |
| HTTP | L'HyperText Transfer Protocol |
| DOM | Document Object Model, C'est une représentation hiérarchique d'une page web qui permet aux programmes JavaScript et aux langages de script d'accéder et de manipuler le contenu, la structure et le style d'une page web. |

# Liste des figures

[Figure 1: le logo de l'EMSI 16](#_Toc165671944)

[Figure 2: Les acteurs et relation d’héritage. 31](#_Toc165671945)

[Figure 3: Diagramme général de cas d’utilisation 32](#_Toc165671946)

[Figure 4: Cas d'utilisation "Client" 33](#_Toc165671947)

[Figure 5: Cas d'utilisation "Vendeur" 34](#_Toc165671948)

[Figure 6: Cas d'utilisation "Administrateur" 35](#_Toc165671949)

[Figure 7: Diagramme de classe 37](#_Toc165671950)

[Figure 8: Sign In button 39](#_Toc165671951)

[Figure 9: Page de Connection et Registeer 40](#_Toc165671952)

[Figure 10: Logout Button 40](#_Toc165671953)

[Figure 11: Page de profile 41](#_Toc165671954)

[Figure 12: Zone de recherche 42](#_Toc165671955)

[Figure 13: Page des produits. 42](#_Toc165671956)

[Figure 14: Details du produit. 43](#_Toc165671957)

[Figure 15: Page panier. 43](#_Toc165671958)

[Figure 16: Page Stripe de payment. 44](#_Toc165671959)

[Figure 17: Page commandes. 44](#_Toc165671960)

[Figure 18: Page de wishlist. 45](#_Toc165671961)

[Figure 19: Button "Devenir Vendeur" 45](#_Toc165671962)

[Figure 20: Page des utilisateurs. 46](#_Toc165671963)

[Figure 21: Page modifier utilisateur. 46](#_Toc165671964)

[Figure 22: Page creer utilisateur. 47](#_Toc165671965)

[Figure 23: Page des produits. 47](#_Toc165671966)

[Figure 24: Page modifier produit. 48](#_Toc165671967)

[Figure 25: Page Creer produit. 48](#_Toc165671968)

[Figure 26: Page categories "admin". 49](#_Toc165671969)

[Figure 27: Page categories "home". 49](#_Toc165671970)

[Figure 28: Page modifier categorie. 50](#_Toc165671971)

[Figure 29: Page creer categorie. 50](#_Toc165671972)

[Figure 30: Page profile "vendeur". 51](#_Toc165671973)

[Figure 31: Page produits "vendeur". 51](#_Toc165671974)

[Figure 32: Button ajouter produit. 52](#_Toc165671975)

[Figure 33: Page ajouter produit "vendeur". 52](#_Toc165671976)

[Figure 34: Button supprimer produit "vendeur". 53](#_Toc165671977)

[Figure 35: Button modifier produit "vendeur". 53](#_Toc165671978)

[Figure 36: Page modifier produit "vendeur". 54](#_Toc165671979)

[Figure 37: Page d'accueil du postman. 56](#_Toc165671980)

[Figure 38: Comment fonctionne Postman? 56](#_Toc165671981)

[Figure 39: Requet http "GET" 57](#_Toc165671982)

[Figure 40: Requet http "POST" 58](#_Toc165671983)

[Figure 41: Requet http "PATCH" 58](#_Toc165671984)

[Figure 42: Requet http "DELETE" 59](#_Toc165671985)

[Figure 43: Comment fonctionne le commande ? 60](#_Toc165671986)

[Figure 44: Page panier. 60](#_Toc165671987)

[Figure 45: la page de paiement 61](#_Toc165671988)

[Figure 46: La page commande. 62](#_Toc165671989)

[Figure 47: La commande sur Stripe. 62](#_Toc165671990)

# Liste des tableaux

[Table 1: Description Textuelle « Commander les produits » 33](#_Toc165660896)

[Table 2: Description Textuelle « Consulter les recommendations » 34](#_Toc165660897)

[Table 3: Description Textuelle « Demander pour devenir vendeur » 34](#_Toc165660898)

[Table 4: Description Textuelle « Géstion des produits » 35](#_Toc165660899)

[Table 5: Description Textuelle « Gestion des commandes » 35](#_Toc165660900)

[Table 6: Description Textuelle « Gestion des utilisateurs » 36](#_Toc165660901)

[Table 7: Description Textuelle « Gestion des commandes » 36](#_Toc165660902)

Table of Contents

[Dedicase 2](#_Toc166277950)

[Remerciement 3](#_Toc166277951)

[Abstract 4](#_Toc166277952)

[Résumé 5](#_Toc166277953)

[Glossaire 6](#_Toc166277954)

[Liste des figures 7](#_Toc166277955)

[Liste des tableaux 8](#_Toc166277956)

[Introduction Générale 12](#_Toc166277957)

[Sujet Amené : Contexte et Cadre de l'Étude 12](#_Toc166277958)

[Sujet Posé : Problématique, Objectifs et Démarche 12](#_Toc166277959)

[Sujet Divisé : Plan de Rédaction 12](#_Toc166277960)

[Introduction 15](#_Toc166277961)

[Conclusion 16](#_Toc166277962)

[Introduction 18](#_Toc166277963)

[I. Commerce électronique 18](#_Toc166277964)

[a. Évolution du Commerce Électronique 18](#_Toc166277965)

[b. Modèles et Types de Commerce Électronique 18](#_Toc166277966)

[c. Avantages du Commerce Électronique 19](#_Toc166277967)

[d. Défis et Opportunités 19](#_Toc166277968)

[II. Intelligence Artificielle (IA) 19](#_Toc166277969)

[III. Chatbots 20](#_Toc166277970)

[a. Impact sur le Commerce Électronique 20](#_Toc166277971)

[Méthodologie 21](#_Toc166277972)

[I. Recherche Initiale 21](#_Toc166277973)

[II. Conception de l'Architecture 22](#_Toc166277974)

[III. Développement du Chatbot 22](#_Toc166277975)

[IV. Tests et Itérations 22](#_Toc166277976)

[React 23](#_Toc166277977)

[Django 23](#_Toc166277978)

[REST framework 24](#_Toc166277979)

[Axios 24](#_Toc166277980)

[Postman 24](#_Toc166277981)

[Redux 25](#_Toc166277982)

[React Router DOM 25](#_Toc166277983)

[SQLite 25](#_Toc166277984)

[GitHub 26](#_Toc166277985)

[Visual Studio Code 26](#_Toc166277986)

[Google Generative AI 26](#_Toc166277987)

[Stripe 27](#_Toc166277988)

[Conclusion 27](#_Toc166277989)

[I. Introduction 29](#_Toc166277990)

[II. Diagramme de cas d’utilisation 29](#_Toc166277991)

[1. Identification des acteurs du système 29](#_Toc166277992)

[2. Diagramme général de cas d’utilisation 31](#_Toc166277993)

[3. Description textuelle des cas généraux d'utilisation 32](#_Toc166277994)

[a. Module Client 32](#_Toc166277995)

[b. Module Vendeur 33](#_Toc166277996)

[c. Module Administrateur 34](#_Toc166277997)

[III. Diagramme de classes 35](#_Toc166277998)

[IV. Conclusion 36](#_Toc166277999)

[Introduction 38](#_Toc166278000)

[I. Implémentation de la base de données 38](#_Toc166278001)

[C’est quoi base de données? 38](#_Toc166278002)

[Choix du système de gestion de la base de données 38](#_Toc166278003)

[II. Server (Back End) 39](#_Toc166278004)

[a. Presentation du framework express js 39](#_Toc166278005)

[Pourqoui Express js? 39](#_Toc166278006)

[Rôle de Express 39](#_Toc166278007)

[L’application Express.js 40](#_Toc166278008)

[a. Mise en place du backend de conception-MVC (voir annexe) 42](#_Toc166278009)

[b. Rôle de Express 43](#_Toc166278010)

[c. Fonctionnement du serveur 43](#_Toc166278011)

[III. Client (Front End) 44](#_Toc166278012)

[Conclusion 45](#_Toc166278013)

[Conclusion et perspective 46](#_Toc166278014)

[Annexe 47](#_Toc166278015)

[I. L’architecture MVC et l’utilisation des Frameworks 47](#_Toc166278016)

[a. Les Frameworks 47](#_Toc166278017)

[b. L’architecture MVC 47](#_Toc166278018)

[II. Gestion des produits avec Postman 49](#_Toc166278019)

[1. La page principale de Postman 49](#_Toc166278020)

[2. Comment fonctionne Postman? 49](#_Toc166278021)

[HTTP Requests 50](#_Toc166278022)

[Responses 50](#_Toc166278023)

[III. Comment la Commande travailler sous couverture 53](#_Toc166278024)

[a. Choisissez les produits 53](#_Toc166278025)

[b. Valider le payement 54](#_Toc166278026)

[c. Consulter la commande 54](#_Toc166278027)

[Bibliographie 56](#_Toc166278028)

# Introduction Générale

## Sujet Amené : Contexte et Cadre de l'Étude

La transformation numérique a révolutionné le paysage du commerce en ligne, offrant aux consommateurs des possibilités d'achat et d'interaction plus vastes que jamais. Dans ce contexte en constante évolution, les entreprises cherchent continuellement à améliorer l'expérience utilisateur et à optimiser leurs processus pour rester compétitives. C'est dans ce cadre que s'inscrit notre projet de conception d'une plateforme de commerce en ligne intégrant un chatbot intelligent.

## Sujet Posé : Problématique, Objectifs et Démarche

Face à la complexité croissante des attentes des consommateurs et à la nécessité de fournir un service client efficace, le besoin de solutions innovantes se fait ressentir. Ainsi, notre projet vise à répondre à cette problématique en développant une plateforme de commerce en ligne dotée d'un chatbot intelligent. Les objectifs principaux de ce rapport sont de détailler les étapes de conception, de développement et d'intégration de cette plateforme, ainsi que de présenter les résultats obtenus.

Les objectifs spécifiques incluent la création d'une interface utilisateur conviviale, l'intégration de fonctionnalités de chatbot pour l'assistance client et la personnalisation de l'expérience utilisateur, et enfin, l'évaluation de l'efficacité de la plateforme dans l'amélioration de l'expérience utilisateur et des performances commerciales.

Il convient de noter que ce rapport ne vise pas à présenter les résultats du projet, mais plutôt à décrire le processus de développement et les choix méthodologiques effectués.

## Sujet Divisé : Plan de Rédaction

Ce rapport sera divisé en plusieurs sections principales, chacune abordant une phase spécifique du projet :

Analyse des Besoins Utilisateurs : Cette section explorera les attentes et les besoins des utilisateurs en matière de commerce en ligne et d'assistance client.

Conception de l'Architecture : Nous détaillerons ici l'architecture logicielle conçue pour la plateforme de commerce et le chatbot.

Développement et Intégration : Cette section traitera du processus de développement des fonctionnalités de la plateforme et de l'intégration du chatbot.

Tests et Évaluation : Nous présenterons les tests réalisés pour évaluer la convivialité de la plateforme et les performances du chatbot.

Conclusion et Perspectives : Enfin, nous conclurons en résumant les résultats obtenus et en discutant des perspectives d'amélioration futures.

Chapitre1: L’organisme d’accueil

## Introduction



Figure 1: le logo de l'EMSI

L'École Marocaine des Sciences de l'Ingénieur (EMSI) est une institution d'enseignement supérieur située au Maroc, reconnue pour son excellence académique et son engagement envers la formation d'ingénieurs compétents et qualifiés. Au sein de l'EMSI, le Département de Génie Informatique se distingue par son programme académique complet et sa formation axée sur les technologies de l'information et de la communication (TIC).

Le Département de Génie Informatique de l'EMSI offre un cursus d'études approfondi couvrant les principaux domaines de l'informatique, tels que la programmation, les systèmes d'information, les réseaux informatiques, la sécurité informatique, l'intelligence artificielle, le génie logiciel, et bien plus encore. Les étudiants sont exposés à des cours théoriques de pointe ainsi qu'à des travaux pratiques concrets, leur permettant d'acquérir des compétences techniques solides et une compréhension approfondie des principes fondamentaux de l'informatique.

En plus de sa rigueur académique, le Département de Génie Informatique de l'EMSI se distingue par son engagement envers l'innovation et la recherche. Les étudiants sont encouragés à participer à des projets de recherche, des concours technologiques et des événements industriels pour développer leurs compétences et explorer de nouvelles tendances dans le domaine de l'informatique.

Les diplômés du Département de Génie Informatique de l'EMSI sont hautement prisés sur le marché du travail, tant au niveau national qu'international. Leur formation polyvalente et leur expérience pratique leur permettent de s'adapter rapidement aux exigences de l'industrie informatique et de contribuer de manière significative à des projets variés dans divers secteurs, tels que les technologies de l'information, les télécommunications, la finance, la santé, et bien d'autres encore.

En résumé, le Département de Génie Informatique de l'EMSI offre aux étudiants une formation de qualité, alignée sur les besoins du marché et axée sur l'excellence académique, les préparant ainsi à une carrière réussie dans le domaine passionnant et dynamique de l'informatique.

## Conclusion

En conclusion, le Département de Génie Informatique de l'EMSI se positionne comme un pilier de l'excellence académique et de l'innovation dans le domaine de l'informatique au Maroc. Son programme complet et équilibré, alliant théorie et pratique, prépare les étudiants à relever avec succès les défis de l'industrie informatique moderne. Avec un engagement ferme envers la recherche et l'innovation, ainsi qu'une forte réputation d'employabilité, l'EMSI offre une voie solide pour ceux qui aspirent à une carrière réussie et enrichissante dans le domaine dynamique des technologies de l'information et de la communication.Haut du formulaire

Chapitre 2: Analyse et Conception du projet

## Introduction

Ce chapitre donner une vision de conception à travers des diagrammes de cas d’utilisation, des diagrammes de séquences et des diagrammes de classes.

## Diagramme de cas d’utilisation

### Identification des acteurs du système

Les types d’acteurs qui participent au système sont les suivants :

* **Client** : un utilisateur final de l'application. Il peut consulter les plats, les ajouter au favoris et au panier et les commander. Il possède un login et un mot de passe et accède à un nombre d'écrans et vues de l'application qui sont relatifs à sa responsabilité.
* **Administrateur** : Un utilisateur administrateur avec un niveau d'habilitation qui lui permet:
  + Gestion des utilisateurs.
* Gestion des catégories.
* Gestion des plats.
* Gestion des commandes.

La figure ci-dessous représente les relations d'héritage entre ces acteurs :

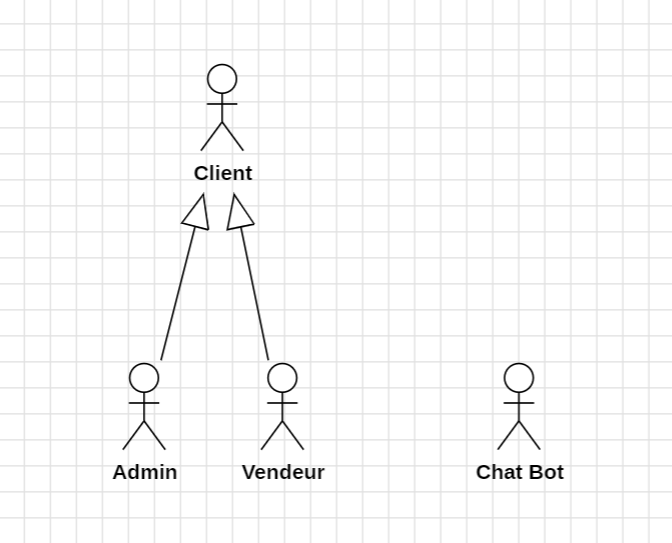


Figure 2: Les acteurs et relation d’héritage.

### Diagramme général de cas d’utilisation

Le diagramme de cas d’utilisation ci-dessous représente l’ensemble des cas d’utilisations relative à l’application. Le but de ce diagramme est d’avoir une vision globale sur CLOE :

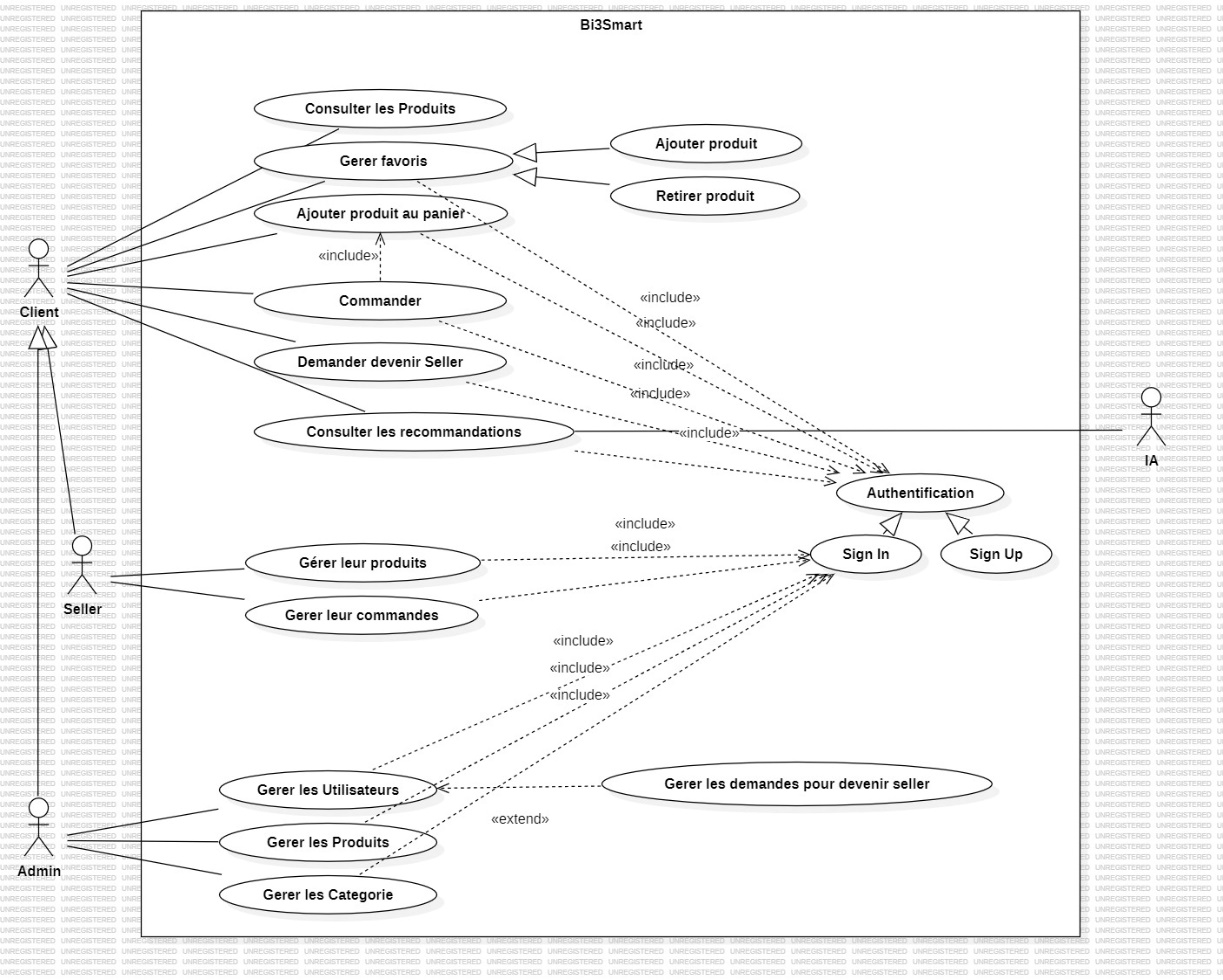


Figure 3: Diagramme général de cas d’utilisation

### Description textuelle des cas généraux d'utilisation

#### Module Client

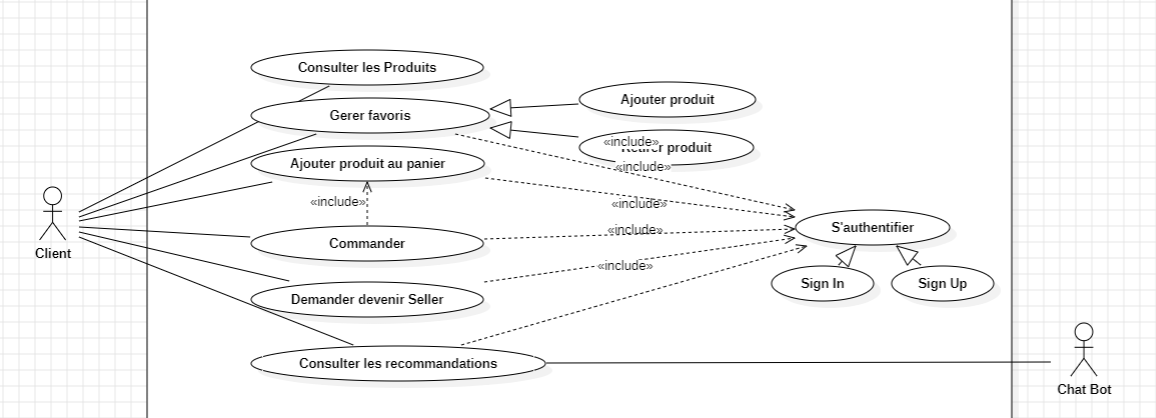


Figure 4: Cas d'utilisation "Client"

##### Description Textuelle:

On prend quelques exemples des cas d’utilisation.

|  |  |
| --- | --- |
| Titre de scénario | Commander les plats |
| Acteur | Client |
| Description |  |
| Précondition | Authentification |

Table 1: Description Textuelle « Commander les produits »

|  |  |
| --- | --- |
| Titre de scénario | Ajouter les plats aux favoris |
| Acteur | Client |
| Description |  |
| Précondition | Authentification |

Table 2: Description Textuelle « Consulter les recommendations »

#### Module Administrateur

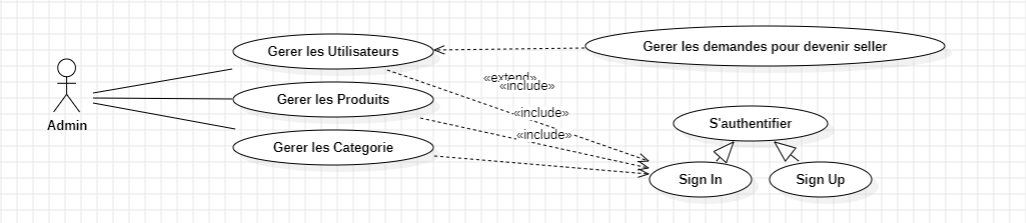


Figure 6: Cas d'utilisation "Administrateur"

##### Description Textuelle:

On prend quelques exemples des cas d’utilisation.

|  |  |
| --- | --- |
| Titre de scénario | Géstion des utilisateurs |
| Acteur | Administrateur |
| Description | Ce cas d'utilisation permet à l'administrateur de gérer les utilisateurs (clients, admins), et de gérer les demandes de devenir vendeur des clients. |
| Précondition | Authentification |

Table 6: Description Textuelle « Gestion des utilisateurs »

|  |  |
| --- | --- |
| Titre de scénario | Géstion des plats |
| Acteur | Administrateur |
| Description | . |
| Précondition | Authentification |

Table 7: Description Textuelle « Gestion des commandes »

## Diagramme de classes

Avant de commencer à concevoir le système d’information, et donc la base de données, il faut tout d’abord passer par la phase de la modélisation du schéma relationnel qui va implémenter la base de données.

Le schéma relationnel est réalisé après l’étude du système cultural, en respectant la méthode Merise ; qui est la référence pour la conception des bases de données.

Elle permet d'organiser les données du système d’information en suivant des règles et des normes. Cela permet de comprendre rapidement comment est structurée la base de données. Cette méthode a aidé à mettre en évidence les différentes entités et relations caractérisant le système cultural.

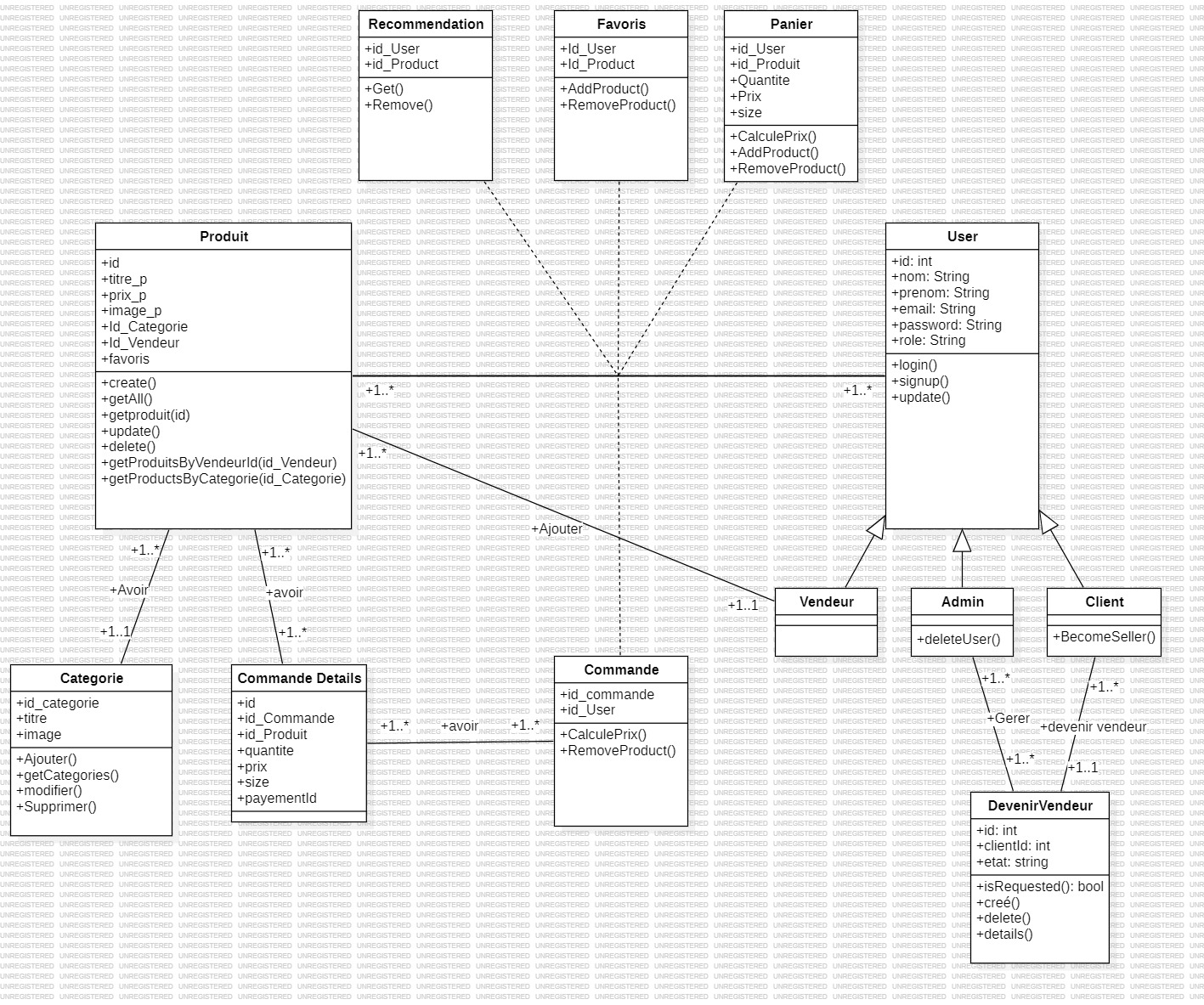


Figure 7: Diagramme de classe

## Diagramme de sequences

## Conclusion

Dans ce chapitre, nous avons décrit la conception globale du système, à travers les différents diagrammes de UML ainsi qu’une description fonctionnelle du projet. Dans le chapitre suivant, nous abordons la phase de l’étude technique du projet.

Chapitre 3: Etat d'art

## Introduction

L'état de l'art, souvent appelé également "état de la technique" ou "état de la pratique", représente le panorama actuel des connaissances, des technologies, des méthodes ou des pratiques dans un domaine spécifique à un moment donné. Il s'agit d'une exploration approfondie et systématique des avancées les plus récentes, des tendances émergentes et des meilleures pratiques qui définissent le paysage d'une discipline donnée.

### React

Nous avons opté pour React pour plusieurs raisons. Tout d'abord, React est largement reconnu comme l'un des frameworks front-end les plus populaires et les plus performants dans l'industrie du développement web. Sa popularité est due en partie à sa flexibilité, sa facilité d'utilisation et sa grande communauté de développeurs. De plus, React adopte une approche basée sur les composants, ce qui permet de créer des interfaces utilisateur modulaires et réutilisables, facilitant ainsi la maintenance et l'évolutivité des applications. En outre, la capacité de React à gérer efficacement les mises à jour de l'interface utilisateur grâce à sa virtual DOM offre des performances optimales, ce qui est essentiel pour offrir une expérience utilisateur fluide et réactive. (egacy.reactjs.org, 2020)

### C:\Users\Soulaimane\AppData\Local\Microsoft\Windows\INetCache\Content.Word\60d5df5a051d8765f7027878_expressjs.pngExpress

### Axios

Une image contenant Police, Graphique, capture d’écran, noir

Description générée automatiquementNous avons choisi Axios comme client HTTP pour interagir avec notre API RESTful depuis le côté client. Axios est une bibliothèque JavaScript légère et simple à utiliser qui offre une syntaxe conviviale pour effectuer des requêtes HTTP asynchrones. Son support intégré pour les promesses permet une gestion efficace des appels réseau et des réponses asynchrones, ce qui est essentiel pour garantir la réactivité et la fluidité de l'interface utilisateur. De plus, Axios offre un large éventail de fonctionnalités avancées telles que la gestion des intercepteurs, la gestion des erreurs, et la configuration personnalisée des requêtes, ce qui en fait un choix idéal pour les applications web modernes. (axios, 2017)

### Postman

Une image contenant cercle, Graphique

Description générée automatiquementPostman est un outil essentiel pour les développeurs d'API, offrant des fonctionnalités de test, de débogage et de documentation. Il permet de tester facilement les API, d'automatiser les tests, de collaborer avec les membres de l'équipe et de garantir la sécurité des API. Dans le cadre de votre projet de plateforme de commerce avec chatbot intelligent, l'utilisation de Postman facilitera le développement, en assurant la qualité et la fiabilité des API sous-jacentes. (learning.postman.com, 2020)

### Une image contenant Graphique, symbole, Police, logo Description générée automatiquementRedux

Redux est une bibliothèque JavaScript utilisée avec React pour gérer l'état global de l'application de manière prévisible. Avec Redux, vous définissez des actions pour décrire les changements d'état, des reducers pour spécifier comment ces changements sont appliqués, et un store centralisé pour stocker l'état global. Dans votre projet de plateforme de commerce avec chatbot intelligent, Redux vous aidera à maintenir un état cohérent et à gérer les interactions utilisateur de manière efficace. (redux, 2015)

### Une image contenant Graphique, Police, graphisme, conception Description générée automatiquementReact Router DOM

React Router DOM est une bibliothèque essentielle pour la navigation dans les applications web à page unique avec React. En définissant des routes correspondant à des URL spécifiques, elle permet de charger dynamiquement les composants associés. Vous pouvez également gérer la navigation programmée et les paramètres d'URL, offrant ainsi une expérience utilisateur fluide et dynamique dans votre plateforme de commerce avec chatbot intelligent. (reactrouter, 2020)

### GitHub

Une image contenant noir, cercle, noir et blanc, obscurité

Description générée automatiquementGitHub est une plateforme de développement collaboratif basée sur Git, permettant le stockage centralisé du code source, la collaboration entre les membres de l'équipe via les pull requests et les issues, le suivi des modifications, l'automatisation du déploiement avec GitHub Actions, et la gestion de projet avec des fonctionnalités telles que les tableaux Kanban. Son utilisation dans votre projet de plateforme de commerce avec chatbot intelligent facilite la gestion du code, la collaboration et le déploiement continu de votre application. (docs.github, 2016)

### Visual Studio Codevscode-logo

Visual Studio Code (VS Code) est un éditeur de code gratuit et open-source développé par Microsoft. Il offre une expérience de codage fluide avec une coloration syntaxique, une saisie semi-automatique et une intégration Git native. VS Code est hautement extensible grâce à un large éventail d'extensions et offre des outils de débogage et un terminal intégré. Son utilisation dans votre projet de plateforme de commerce avec chatbot intelligent peut améliorer la productivité et simplifier le développement. (code.visualstudio, 2018)

### Une image contenant Police, Graphique, logo, graphisme Description générée automatiquementStripe

Django Stripe est une intégration qui permet d'ajouter les fonctionnalités de paiement de Stripe à vos applications Django. Cette combinaison offre une gestion facile des paiements en ligne, y compris les paiements uniques, les abonnements et les remboursements. Avec Django Stripe, vous pouvez sécuriser et gérer efficacement les transactions financières dans vos applications Django. (docs.stripe, 2016)

## Conclusion

En conclusion, l'état de l'art dans le développement d'une plateforme de commerce avec un chatbot intelligent implique l'utilisation d'une variété de technologies et de méthodologies de pointe. Cela inclut l'utilisation de techniques avancées de traitement du langage naturel (NLP) pour l'interaction avec le chatbot, l'intégration avec des plateformes telles que Stripe pour un traitement des paiements fluide, et l'adoption de frameworks comme React et Django pour la construction d'interfaces web réactives et de services backend.

De plus, l'incorporation d'outils comme Postman pour les tests d'API, Visual Studio Code pour l'édition de code, et Git pour le contrôle de version améliore le processus de développement, assurant l'efficacité, la collaboration et la qualité du code. En outre, l'exploration de technologies émergentes telles que les modèles d'IA générative de Google peut potentiellement enrichir les capacités de la plateforme, offrant des solutions innovantes pour des expériences utilisateur personnalisées.

Dans l'ensemble, rester à jour sur les dernières avancées technologiques, tout en sélectionnant et intégrant soigneusement les outils et méthodologies appropriés, est essentiel pour créer une plateforme de commerce avec un chatbot intelligent de pointe qui répond aux besoins et aux attentes évolutifs des utilisateurs.

Chapitre4 : Mise en oeuvre et realisation

## Introduction

Cette partie présente les étapes de développement de ce projet. Le modèle relationnel implémenté du côté de la base de données, et la présentation du framework **React js** et **Express.js** côté développement web. Nous citerons également quelques exemples d’interfaces web de l’application.

## Implémentation de la base de données

Après avoir conçu le système d'information (diagramme de classes) et donc la base de données, il va maintenant falloir construire la base de données basée sur le système.

### C’est quoi base de données?



Une base de données est une collection organisée d’informations ou de données structurées, généralement stockées électroniquement dans un système informatique. Une base de données est généralement contrôlée par un système de gestion de base de données (SGBD).

### Choix du système de gestion de la base de données

|  |  |
| --- | --- |
| Relationnelles | Non relationnelles |
| Les bases de données relationnelles sont devenues dominantes dans les années 1980. Les éléments d'une base de données relationnelle sont organisés sous forme d'un ensemble de tables avec des colonnes et des lignes. La technologie des bases de données relationnelles constitue le moyen le plus efficace et le plus flexible d’accéder à des informations structurées. | Une base de données NoSQL, ou base de données non relationnelle, permet de stocker et de manipuler des données non structurées et semi-structurées (contrairement à une base de données relationnelle). Les bases de données NoSQL sont devenues populaires à mesure que les applications Web devenaient plus courantes et plus complexes. |
| * MySQL * Microsoft SQL Server | * Mongo DB * Cassandra |

Le système de gestion de bases de données (SGBD) a été axé sur SQL Server car il constitue une des conditions de travail sur le projet proposé par les professeurs, qui est le système de gestion de bases de données relationnelles développé par Microsoft.

## Server (Back End)

### Presentation du framework express js

Aujourd’hui, l’utilisation d’un Framework est devenue obligatoire dans le monde professionnel du développement web. Les Framework offrent un mode de travail structuré permettant de maintenir facilement un projet.

#### Pourqoui Express js?

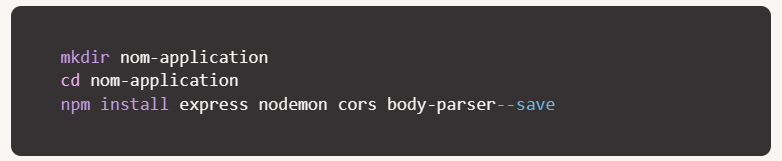
Ce framework fournit un ensemble d’outils pour les applications web, [les requêtes et les réponses HTTP](https://kinsta.com/fr/blog/reduire-nombre-requetes-http/), le routage et les intergiciels permettant de créer et de déployer des applications à grande échelle, prêtes pour l’entreprise. (kinsta, 2022)

#### Rôle de Express

Dans ce projet, nous ne voulons rien afficher en vue express, toutes les données seront envoyées par api, pour une utilisation dans une autre application Web développée par react js, elle gérera la partie conception.

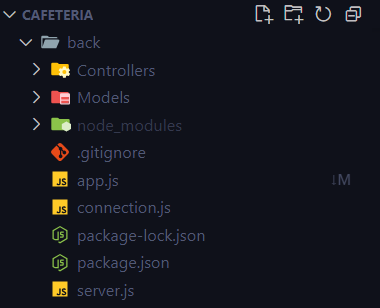
#### L’application Express.js

Maintenant, créons une application de démonstration du monde réel en utilisant le nouveau Express.js 4.19. Pour commencer, créez un répertoire pour l'application et installez les paquets suivants :



##### Structure de l’application

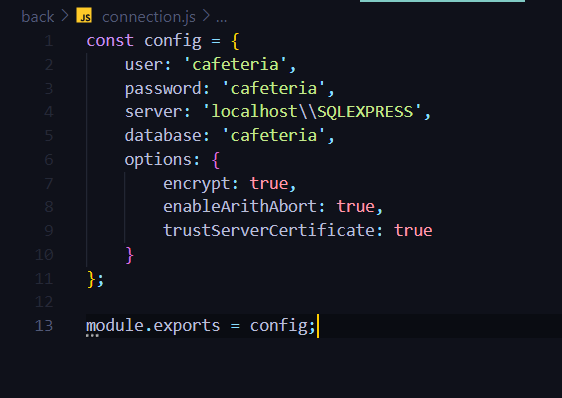
Un projet Express est constitué de plusieurs dossiers et fichiers comme dans la figure:



|  |  |
| --- | --- |
| Ficher / Dossier |  |
| Server.js |  |
| App.js |  |
| Connection.js |  |
| Models |  |
| Controllers |  |
| Node\_modules |  |

##### Configuration

Le fichier connection.js contient toute la configuration du connection avec la base de donnes SQL Server. Il définit les paramètres de connexion à la base de données.



### Mise en place du backend de conception-MVC (voir annexe)

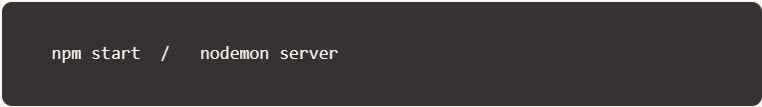
|  |  |
| --- | --- |
| Element |  |
| Le contrôleur | Un controller est une classe qui va contenir différentes méthodes. Chaque méthode correspondant généralement à une opération (URL) de l’application.  Dans le modèle MVC (modèle-vue-contrôleur), le contrôleur contient la logique concernant les actions effectuées par l’utilisateur. En pratique, dans une application express, l’utilisation de contrôleurs permet de libérer les routes du code qu’elles contiennent dans leurs fonctions de rappel. |
| Le modèle | Un modèle est une classe métier représentant une partie des données d'une application.  Dans la plupart des cas, un modèle est associé à une table de la base de données. Model est la classe de base des modèles d'une application. Cette classe met à disposition des fonctionnalités CRUD (pour Create, Read, Update, Delete), offre des possibilités de recherche avancées et permet de gérer les relations entre modèles, le tout sans avoir besoin d'utiliser SQL. |
| La vue | Les vues dans express contiennent une logique qui détermine comment présenter le résultat final de l’application Web aux utilisateurs. La sortie est souvent la sortie HTML qui, après chaque réponse, revient aux utilisateurs. |
| L’api | Pour les applications web, une API (Application Programming Interface) ou « Interface de Programmation d'Applications » en français, est un ensemble de définitions et de protocoles qui permettent à des applications de communiquer et de s'échanger mutuellement des données ou des services. |

### Rôle de Express

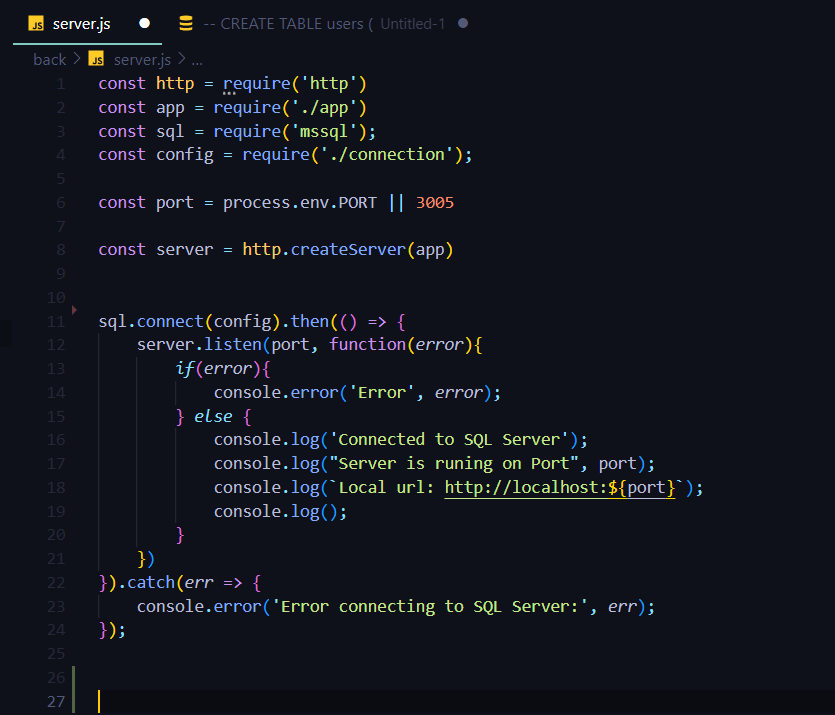
Dans ce projet, nous ne voulons rien afficher en vue express, toutes les données seront envoyées par api, pour une utilisation dans une autre application Web développée par react js, elle gérera la partie conception.

### Fonctionnement du serveur

On démarre le serveur en exécutant la commande npm start, qui exécute la commande nodemon server comme raccourci.



Ensuite nous pouvons utiliser les requêtes HTTP du serveur (CRUD).



## Client (Front End)

## Conclusion

Cette partie a été consacrée à la présentation de la méthode de développement qui a été menée dans ce projet, et les outils qui nous ont servi d’appui pour trouver des solutions à la problématique qui a été posée au début du projet, afin de satisfaire les besoins des utilisateurs.

# Conclusion et perspective

# Annexe

## L’architecture MVC et l’utilisation des Frameworks

### Les Frameworks

Un framework (ou infrastructure logicielle en français) désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application, encore appelé structure logicielle, canevas ou socle d'applications en français, qui établit les fondations d'un logiciel ou son squelette applicatif. Tous les développeurs qui l'utilisent peuvent l'enrichir pour en améliorer l'utilisation.

Avantages de l’utilisation des framework

* + - L'objectif premier d'un framework est d'améliorer la productivité des développeurs qui l'utilisent en offrant des briques prêtes à être utilisées, autrement dit, le framework s'occupe de la forme et permet au développeur de se concentrer sur le fond.
    - Un framework améliore la façon de travailler, en utilisant l’architecture MVC, qui permet d’organiser le code des développeurs.
    - Une communauté active, une documentation de qualité et régulièrement mise à jour.

Inconvénients

* + - Une courbe d'apprentissage élevée. En effet, pour maîtriser un framework, il faut un temps d'apprentissage non négligeable. Chaque brique qui compose un framework a sa complexité et d’autre part des connaissances préalables du Design pattern.

### L’architecture MVC

L'architecture Modèle/Vue/Contrôleur (MVC) est une façon d'organiser une interface graphique d'un programme. Elle consiste à distinguer trois entités distinctes qui sont, le modèle, la vue et le contrôleur ayant chacun un rôle précis dans l'interface.

L'organisation globale d'une interface graphique est souvent délicate. Bien que la façon MVC d'organiser une interface ne soit pas la solution miracle, elle fournit souvent une première approche qui peut ensuite être adaptée. Elle offre aussi un cadre pour structurer une application.

Dans l'architecture MVC, les rôles des trois entités sont les suivants :

* + - Modèle : données (accès et mise à jour).
    - Vue : interface utilisateur (entrées et sorties).
    - Contrôleur : gestion des événements et synchronisation.
    - **Modèle** : Son rôle est d'aller récupérer les informations brutes dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur.
    - **Vue** : S’occupe des interactions avec l’utilisateur, la présentation, la saisie et la validation des données. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions.
    - **Contrôleur** : cette partie gère la dynamique de l’application. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue.



Figure 37: : Interactions entre le modèle, la vue et le contrôleur

## Gestion des produits avec Postman

### La page principale de Postman

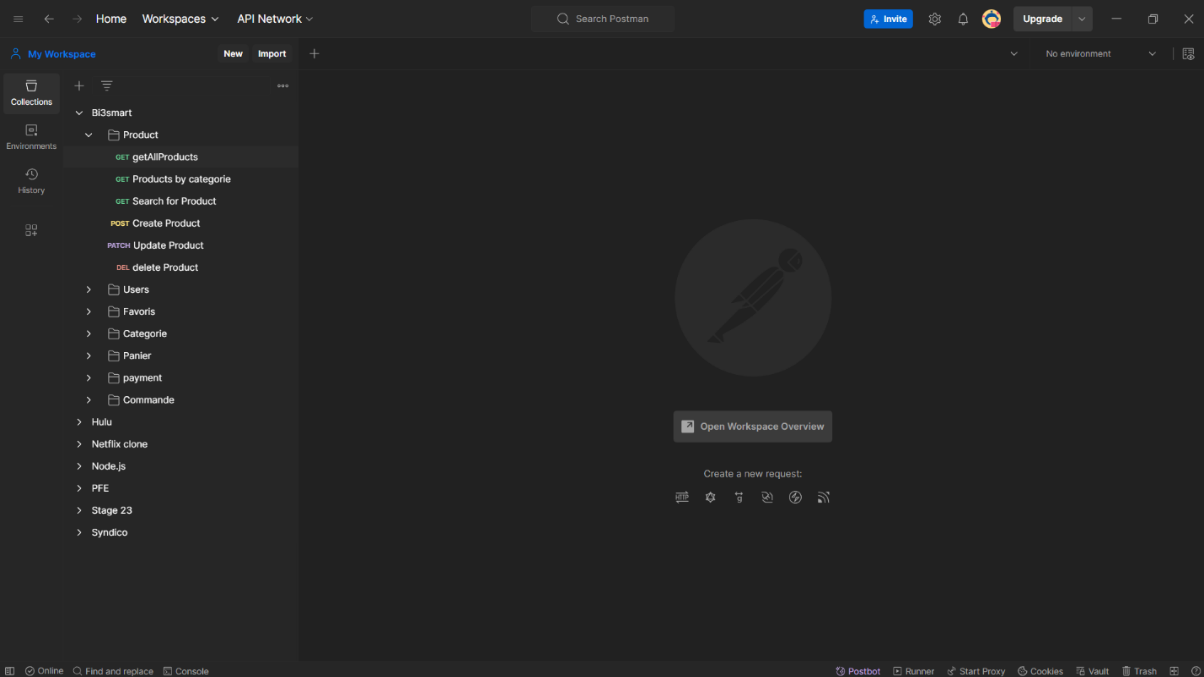


Figure 38: Page d'accueil du postman.

### Comment fonctionne Postman?

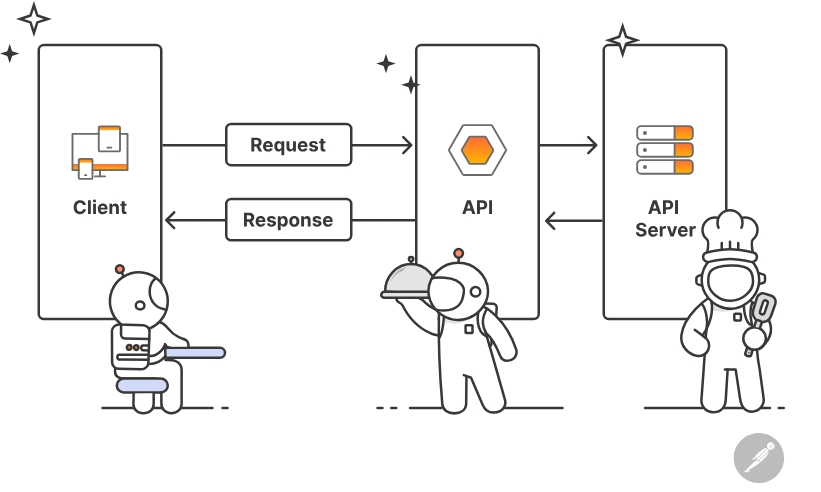


Figure 39: Comment fonctionne Postman?

#### HTTP Requests

|  |  |  |
| --- | --- | --- |
| Titre | Request | Link |
| All plats | Get | http://127.0.0.1:8000/api/plats/ |
| Obtenir les plats par catégorie | Get | http://localhost:3005/plats/categorie/1 |
| Create plat | Post | http://127.0.0.1:8000/api/plats/ |
| Update plat | Patch | <http://127.0.0.1:8000/api/plats/7> |
| Delete plat | Delete | <http://127.0.0.1:8000/api/plats/7> |

Tableau 1: Requets Produits

#### Responses

##### All plats

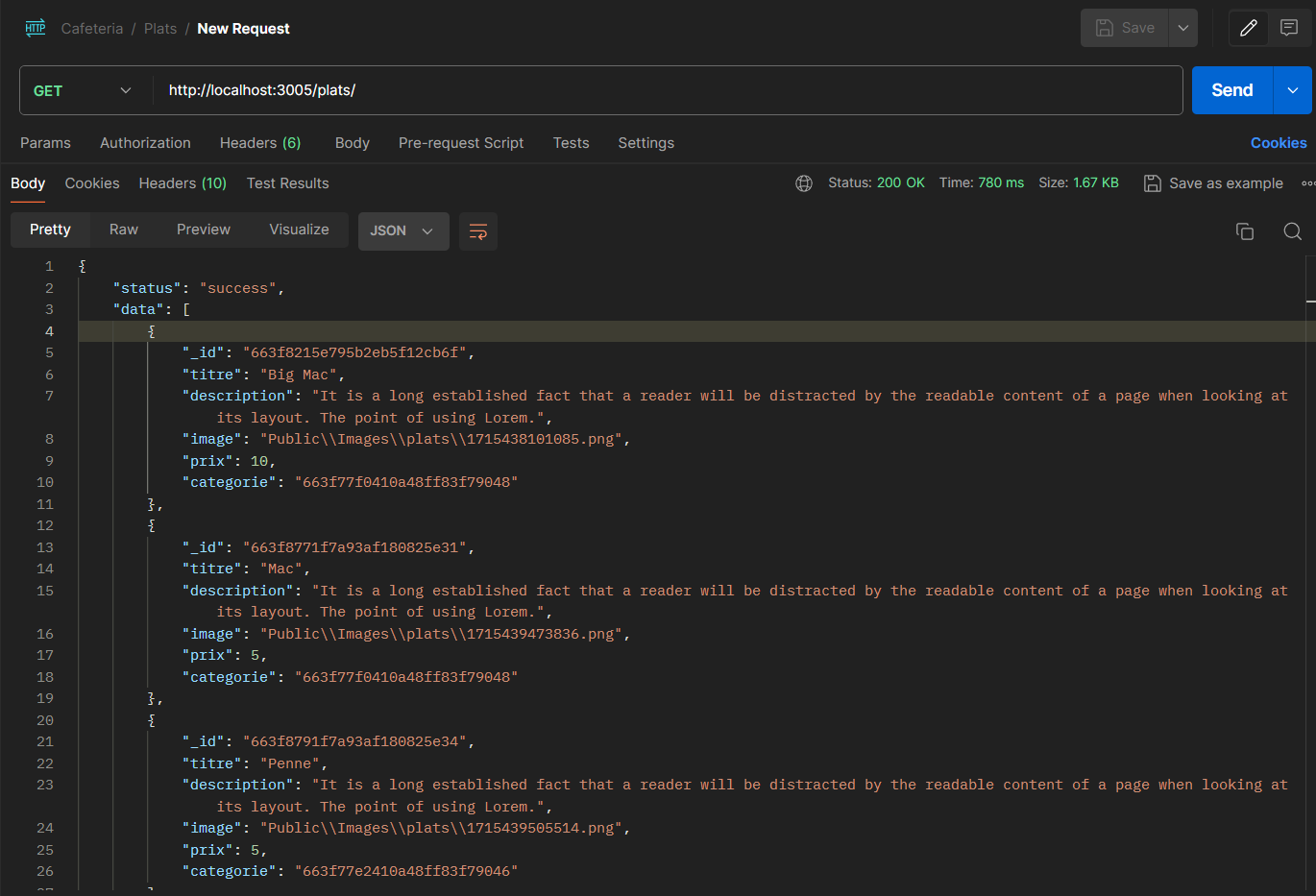


Figure 40: Requet http "GET"

##### Create Product

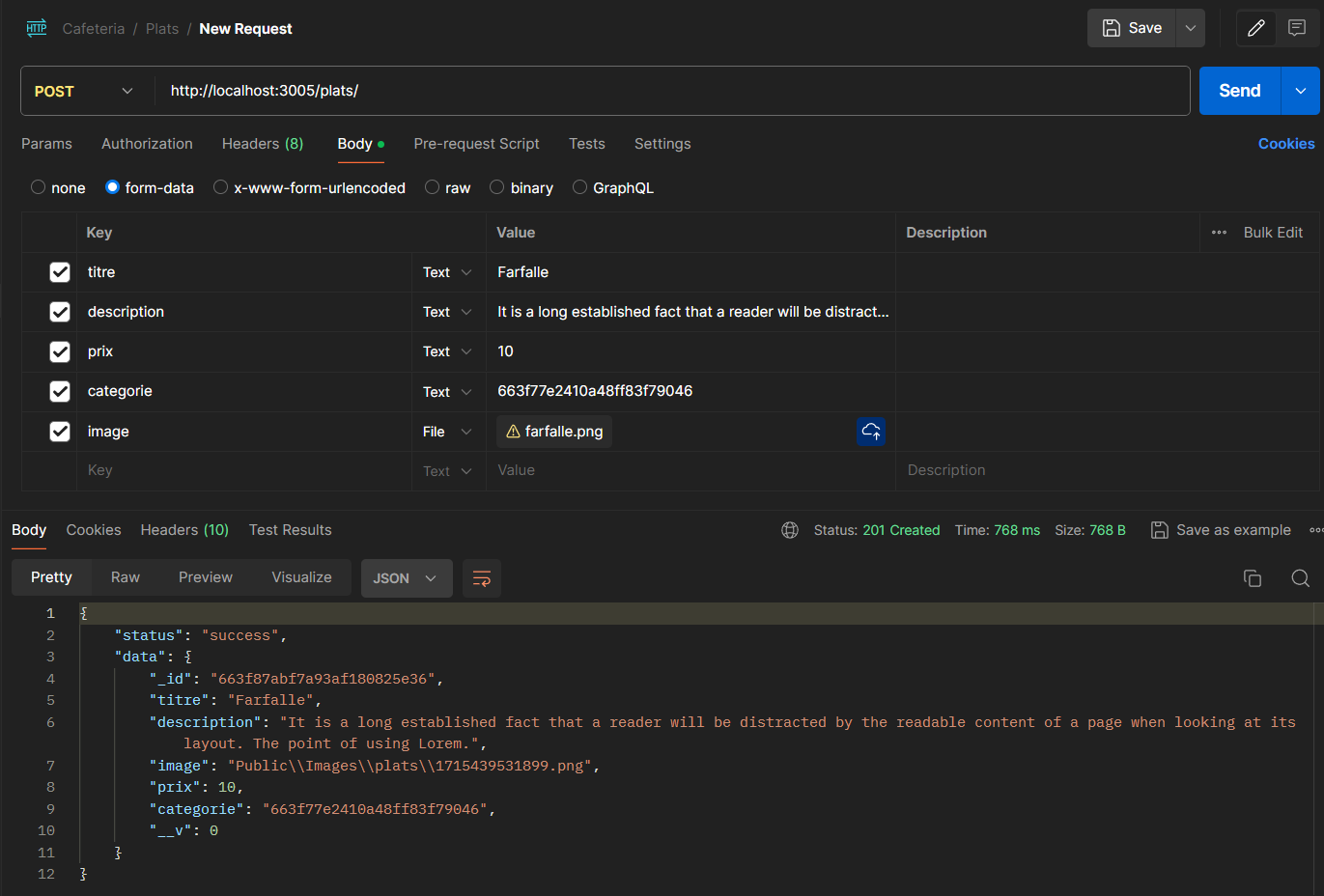


Figure 41: Requet http "POST"

##### Update Product

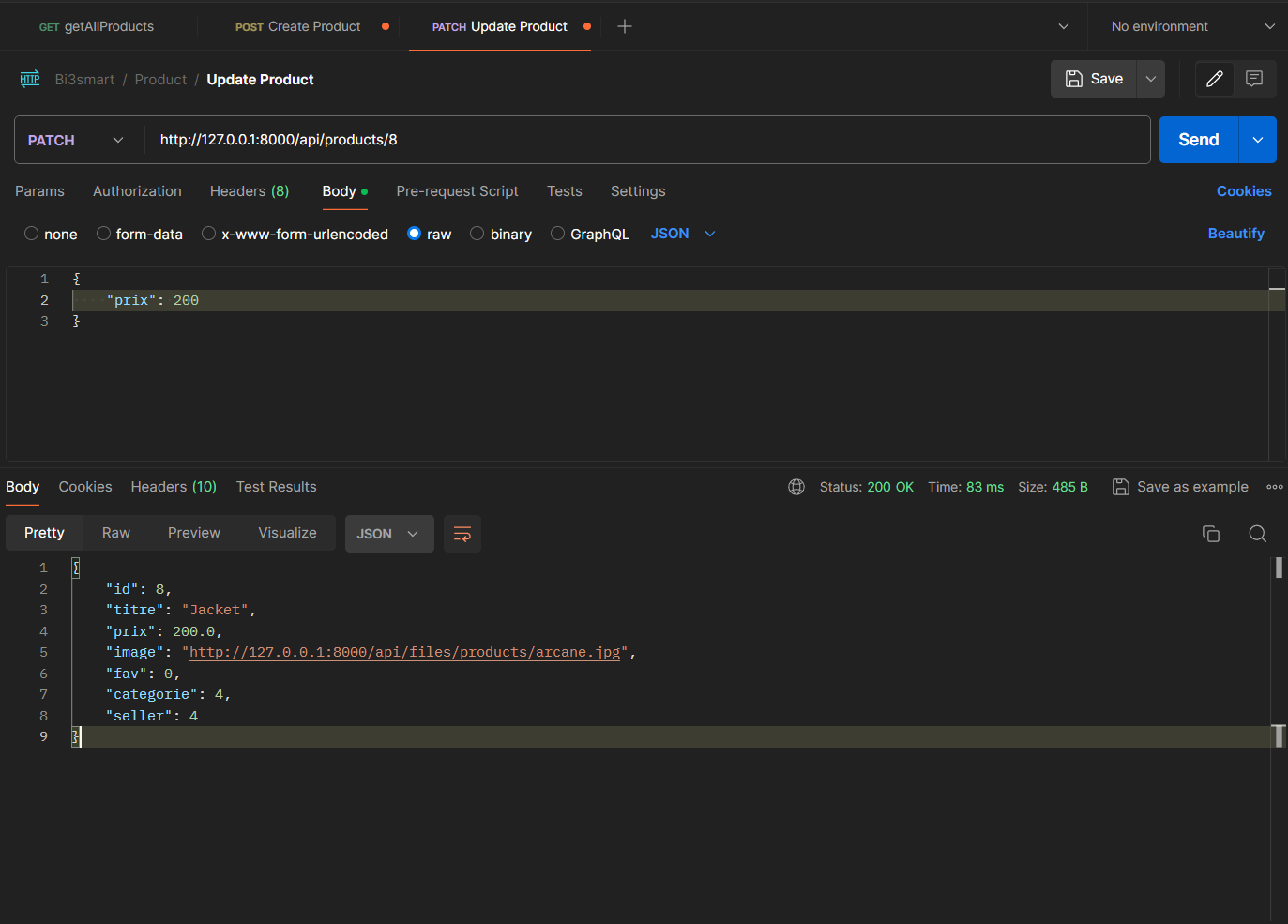


Figure 42: Requet http "PATCH"

##### Delete Product

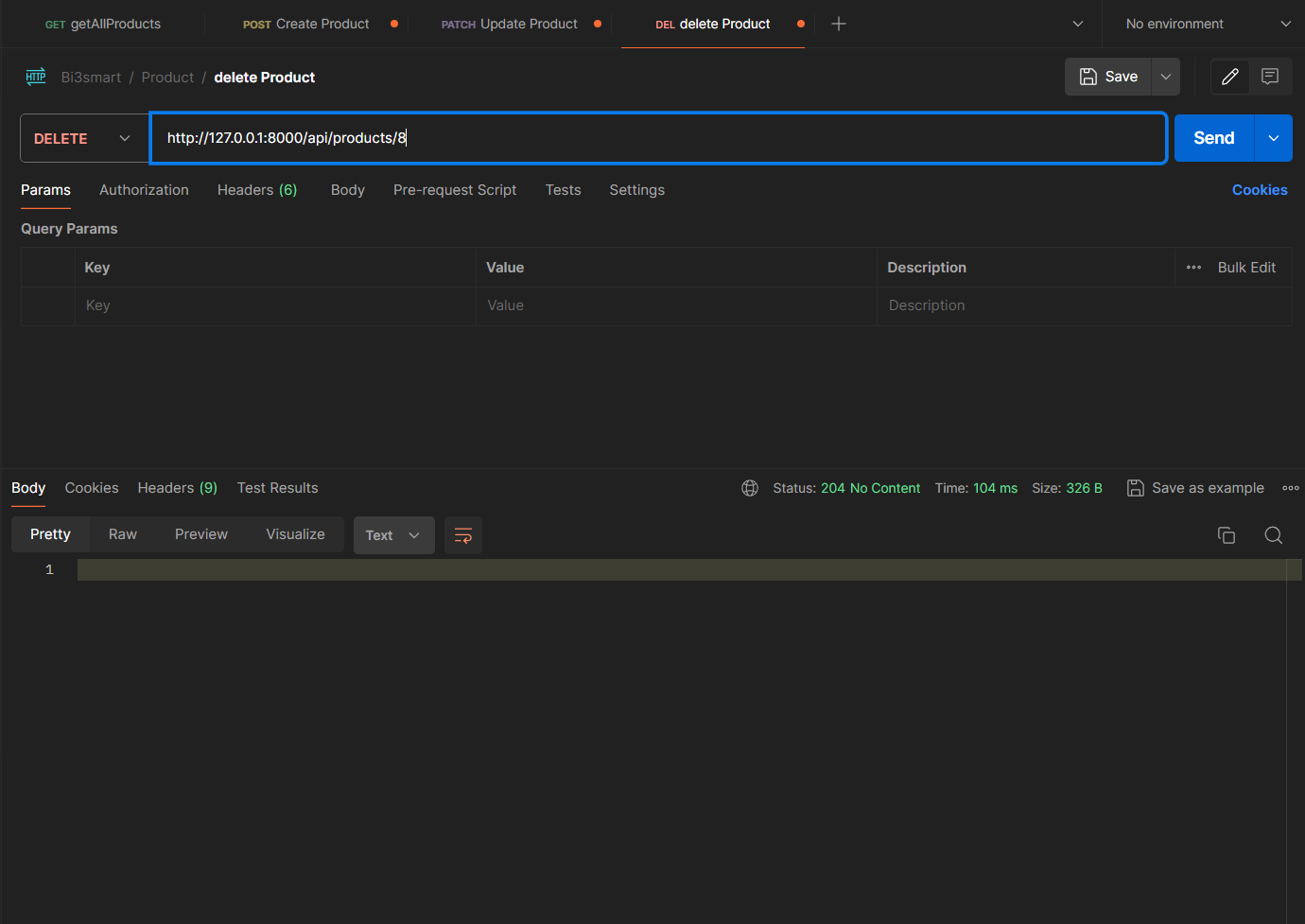


Figure 43: Requet http "DELETE"

## Comment la Commande travailler sous couverture

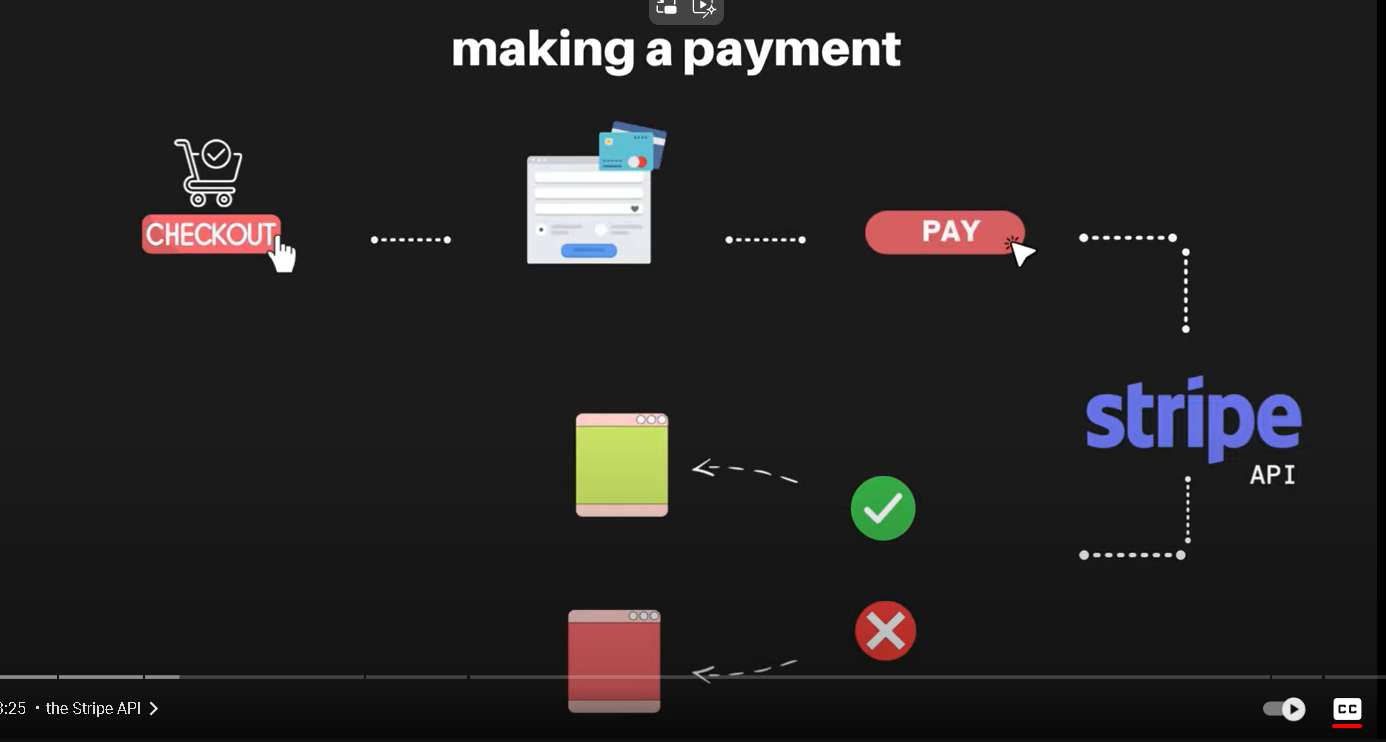


Figure 44: Comment fonctionne le commande ?

* 1. Valider le payement

Après commander le client sera redirigé vers la page de paiement propulsée par Stripe, le client choisira donc de payer par carte ou paypal, puis il renseignera ses informations.

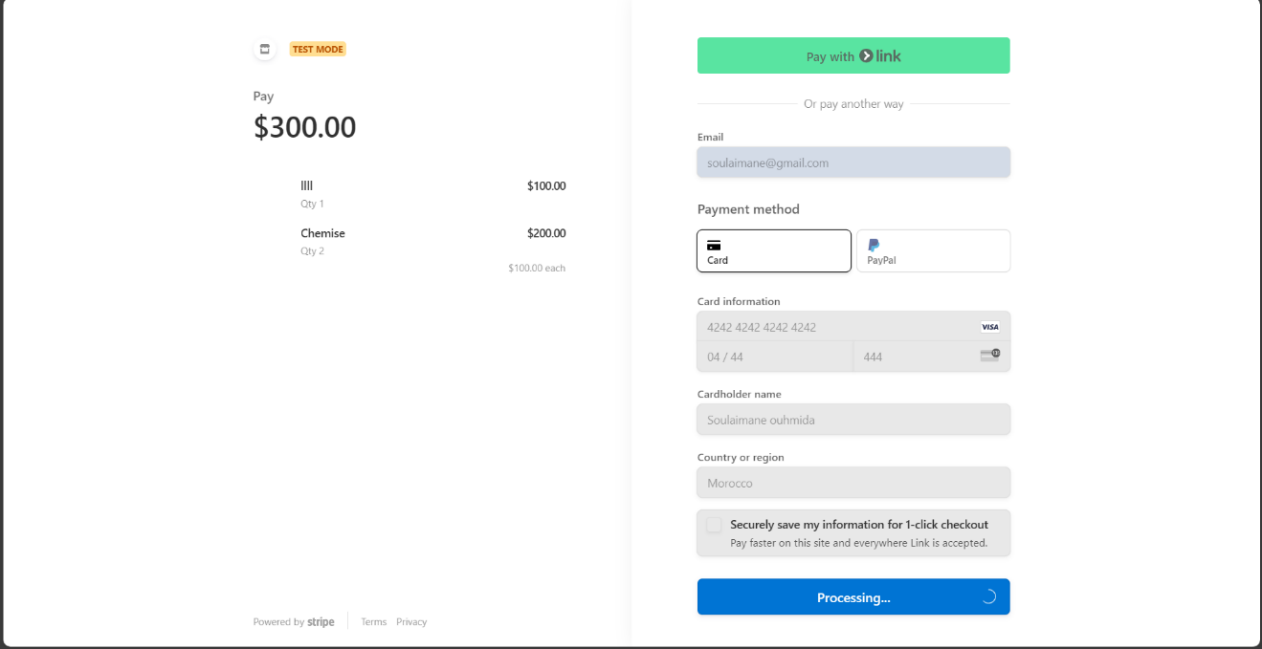


Figure 46: la page de paiement

* 1. Consulter la commande

C:\Users\Soulaimane\AppData\Local\Microsoft\Windows\INetCache\Content.Word\vvv.pngAprès cela, le moteur Stripe traitera le paiement.

S’il n'est pas correct, le client sera redirigé vers la page de la commande.



Si tout se passe bien, la commande sera créée dans la base de données et le client sera redirigé vers la page des commandes, afin qu'il puisse voir sa commande.

Voici les détails de la commande sur le site Stripe :

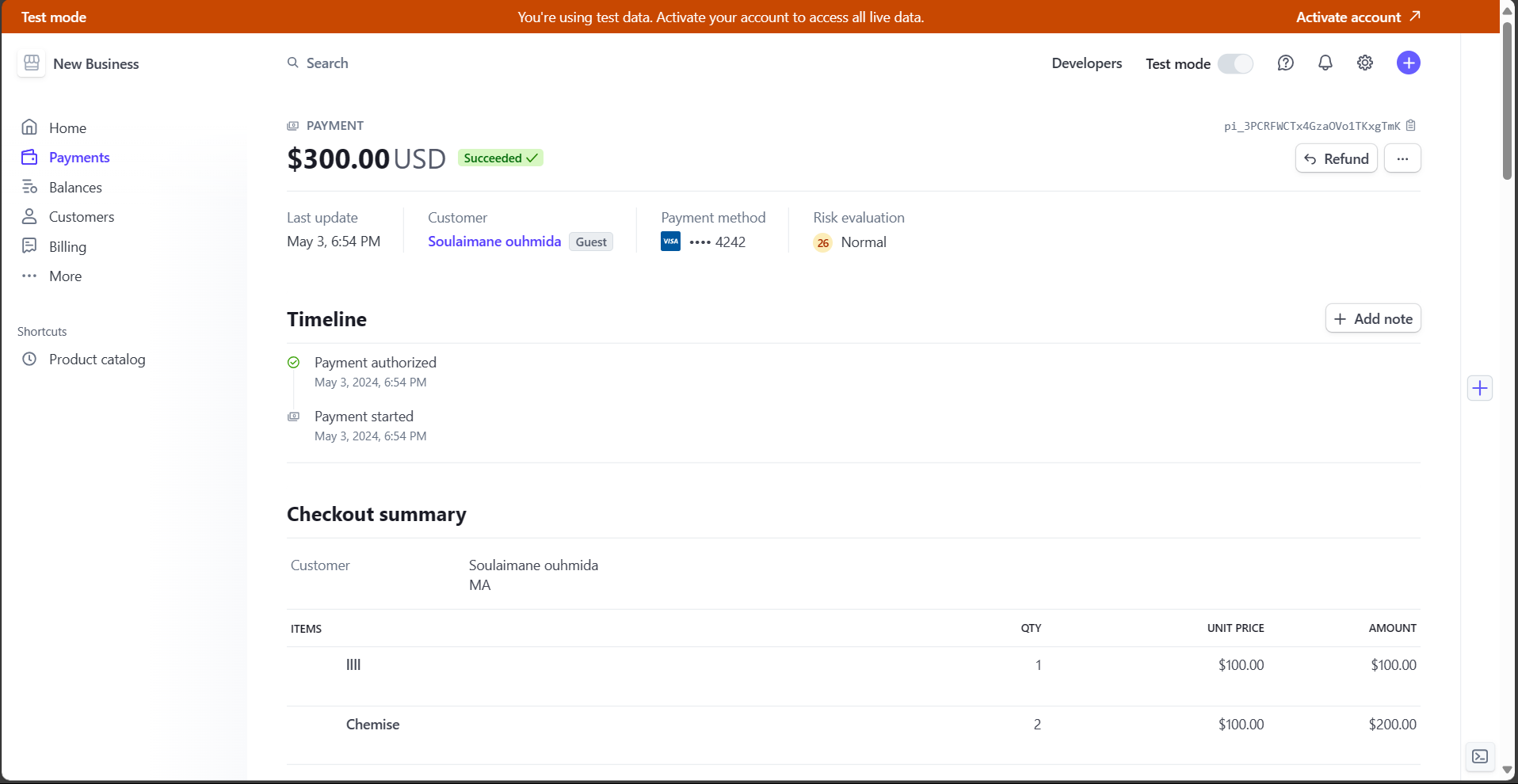


Figure 48: La commande sur Stripe.

# Bibliographie

*axios*. (2017, 10 6). Retrieved from axios: https://axios-http.com/docs/intro

*cloud.google*. (2012, 10 3). Retrieved from cloud.google: https://cloud.google.com/docs

*code.visualstudio*. (2018, 1 20). Retrieved from code.visualstudio: https://code.visualstudio.com/docs

*docs.djangoproject*. (2020, 06 22). Retrieved from djangoproject: https://docs.djangoproject.com/en/5.0/

*docs.github*. (2016, 10 6). Retrieved from docs.github: https://docs.github.com/en

*docs.stripe*. (2016, 10 02). Retrieved from docs.stripe: https://docs.stripe.com/api

*egacy.reactjs.org*. (2020, 02 02). Retrieved from legacy: https://legacy.reactjs.org/docs/getting-started.html

*https://restfulapi.net*. (2012, 02 11). Retrieved from https://restfulapi.net: https://restfulapi.net

*learning.postman.com*. (2020, 8 10). Retrieved from learning.postman.com: https://learning.postman.com/docs/introduction/overview/

*reactrouter*. (2020, 10 16). Retrieved from reactrouter: https://reactrouter.com/en/main

*redux*. (2015, 10 6). Retrieved from redux: https://redux.js.org

*sqlite*. (2006, 8 6). Retrieved from sqlite: https://www.sqlite.org/docs.html