RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université 20 Août 1955 - Skikda

FACULTÉ DES SCIENCES

DÉPARTEMENT D'INFORMATIQUE

MÉMOIRE DE FIN D'ÉTUDE

POUR L'OBTENTION DU DIPLÔME DE MASTER

OPTION : RÉSEAUX ET SYSTÈMES DISTRIBUÉS

Développement d'Une Plate-forme E-Commerce pour des Produits Personnalisés Comme Idée Innovant conformément à la décision ministérielle 1275.

Présenté Par :
Fenazi Idris
Bouznad Houssem Eddine

Supervisé Par :
Dr. BOULEHOUACHE Soufiane
Dr. ZEGHIDA Djamel

2022 - 2023

Remerciements

Nous tenons à remercier Allah qui nous a aidés et nous a accordé patience et courage tout au long des années d'études.

Nous tenons à exprimer nos profondes gratitude a nos parents, dont le soutien inébranlable et les sacrifices ont été les piliers de nos parcours académique. Nous tenons à vous exprimer notre reconnaissance éternelle. Que la vie vous récompense pour vos sacrifices et votre amour infini. Nous souhaitons ardemment que vous puissiez profiter d'une santé durable et d'une longue vie, entourés de bonheur.

Nous tenons également à remercier tous nos frères et soeurs et toute la famille Fenazi, Bouznad et Lameri, Puisse Allah vous donne santé, bonheur, courage et surtout réussite.

Nous tenons à remercier notre encadrants Dr.Boulehoueche Soufiane et Dr.Zeghida Djamel pour tout le temps qu'ils nous a consacrés, pour leurs conseils précieux, pour tous leurs aides et appui durant la réalisation de ce travail.

Nous tenons également à remercier tous nos enseignants de leur générosité et de la patience dont ils ont fait preuve malgré leurs responsabilités scolaires et professionnelles.

Enfin, nous adressons nos plus sincères remerciements à nos amies et toute personne qui a participé de près ou de loin, directement ou indirectement, à la réalisation de ce travail.

ملخص

على مدى السنوات القليلة الماضية، شهدت التجارة الإلكترونية نموًا ملحوظًا، مما أدى إلى تغيير جذري في طريقة تفاعل الشركات مع عملائها. وينبع هذا التطور بشكل أساسي من النمو السريع للإنترنت والانتشار الواسع للرقمنة داخل مجتمعنا المعاصر.

وفي هذا السياق، تركز هذه الدراسة على تطوير منصة للتجارة الإلكترونية موجهة نحو تخصيص المنتج. الهدف الرئيسي لهذه المنصة هو تبسيط عملية إنشاء وتخصيص وشراء المنتجات المطبوعة والشخصية، بما في ذلك الملابس والإكسسوارات.

توفر هذه المنصة فرصة استثنائية للمبدعين والفنانين لإضفاء الحيوية على أفكارهم من خلال تصميم تصميمات حصرية يمكنهم بعد ذلك تطبيقها على مجموعة متنوعة من المنتجات المتاحة على المنصة. يتمتع العملاء بالقدرة على شراء المنتجات التي تتميز بتصميمات من اختيارهم، مما يلغي الحاجة إلى إدارة المخزون الكبير والإنتاج والتسليم والتسويق وخدمة العملاء.

من خلال تسليط الضوء على التقدم التكنولوجي الذي يشكل عالمنا التجاري، تقدم هذه الدراسة منظوراً حول مستقبل التجارة الإلكترونية التي تركز على التخصيص.

كلمات مفتاحية :

تجارة الكترونية ، انترنت ، منصة الكترونية ، الويب ، عملية التخصيص ، الطباعة حسب الطلب ، التصميم.

Résumé

Au cours des dernières années, le commerce électronique a connu une croissance remarquable, transformant fondamentalement la manière dont les entreprises interagissent avec leur clientèle. Cette évolution découle principalement de l'essor rapide d'Internet et de la diffusion généralisée de la numérisation au sein de notre société contemporaine.

Dans ce contexte, cette étude se focalise sur le développement d'une plateforme e-commerce orientée vers la personnalisation de produits. L'objectif principal de cette plateforme est de simplifier le processus de création, de personnalisation et d'achat de produits imprimés et personnalisés, notamment des vêtements et des accessoires.

Cette plateforme offre une opportunité exceptionnelle aux créateurs et aux artistes pour concrétiser leurs idées en concevant des designs exclusifs qu'ils peuvent ensuite appliquer sur une variété de produits disponibles sur la plateforme. Les clients ont la possibilité d'acquérir des produits arborant les designs de leur choix, éliminant ainsi la nécessité de gérer d'importants stocks, la production, la livraison, le marketing et le service client.

En mettant en lumière les progrès technologiques qui façonnent notre monde commercial, cette étude offre une perspective sur l'avenir du commerce électronique centré sur la personnalisation.

Mots Clés :
E-commerce, Internet, Plate-forme, Web, Personnalisation, impression à la demande, Design.

Abstract

Over the past few years, e-commerce has seen remarkable growth, fundamentally transforming the way businesses interact with their customers. This development stems mainly from the rapid growth of the Internet and the widespread diffusion of digitalization within our contemporary society.

In this context, this study focuses on the development of an e-commerce platform oriented towards product customization. The main goal of this platform is to simplify the process of creating, customizing and purchasing printed and personalized products, including clothing and accessories.

This platform provides an exceptional opportunity for creators and artists to bring their ideas to life by designing exclusive designs which they can then apply to a variety of products available on the platform. Customers have the ability to purchase products featuring designs of their choice, eliminating the need to manage large inventory, production, delivery, marketing and customer service.

By highlighting the technological advancements shaping our commercial world, this study offers a perspective on the future of personalization-centric e-commerce.

Key Words :
E-commerce, Internet, Platforme, Web, Personnalisation, Print on Demand, Design.

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

Le commerce électronique, concrètement désigné sous le terme d'e-commerce, a connu une croissance exceptionnelle au cours des dernières années, apportant un changement significatif dans la manière dont les entreprises interagissent avec leur clientèle. Cette transformation trouve sa source principalement dans la montée rapide en puissance d'Internet et dans la généralisation de la numérisation au sein de notre société moderne. De nos jours, le commerce en ligne se positionne comme un pilier central de l'économie mondiale, offrant une large gamme de produits et de services accessible en quelques clics, au bénéfice de millions de consommateurs répartis à travers la planète.

Simultanément, le monde du web a ouvert la voie à de nouvelles opportunités en matière de communication, d'interaction, et de créativité. Il a engendré la création d'une multitude d'entreprises innovantes qui ont réussi à atteindre un public mondial grâce à l'utilisation de l'Internet. Dans ce contexte, ce mémoire se penche sur le développement d'une plateforme e-commerce axée sur la personnalisation des produits. L'objectif principal de cette plateforme est de simplifier le processus de création, de personnalisation et d'achat de produits imprimés et personnalisés, tels que les vêtements.

Cette plateforme offre une opportunité unique aux créateurs et aux artistes de donner vie à leurs idées en concevant des designs uniques, qu'ils peuvent ensuite appliquer sur une gamme de produits disponibles sur la plateforme. Les clients ont la possibilité d'acheter des produits avec les designs de leur choix, éliminant ainsi la nécessité de maintenir des stocks importants, de gérer la production, la livraison, le marketing, et le service client.

L'organisation de ce mémoire se décline en plusieurs chapitres, chacun explorant des aspects spécifiques liés à la conception et à la mise en œuvre de notre plateforme "Hi-Print" pour des produits personnalisés. Voici un aperçu de la structure de ce mémoire :

— le premier chapitre « E-Commerce et Print on Demand » fournit une introduction approfondie à l'ensemble du monde du commerce électronique et l'impression a la demande (POD) qui est notre domaine d'étude.
— Le deuxième chapitre « Le WEB 2.0 » : donne une perspective globale sur le World Wide Web et son développement au fil du temps, en plus de présenter les techniques et les outils clés employés dans le domaine du développement web.
— Le troisième Chapitre « La Conception » présente le processus de conception et mettre en évidence les objectifs, les acteurs et les besoins de notre plate-forme, qui sont essentiels pour passer à la phase de réalisation.

— Le quatrième Chapitre « L'implémentation » traite des divers outils, langues, cadres et techniques utilisés pour développer et réaliser notre plateforme. Par la suite, afin de mettre en évidence leur aspect pratique et facile à utiliser, qui nous a été l'un de nos objectifs principaux, nous présenterons quelques interfaces de notre plateforme.

L'ensemble de ces chapitres donne une vision exhaustive et éclairée de notre projet, fournissant une vue d'ensemble globale tout en posant les fondements essentiels à une compréhension contextuelle.

Ce mémoire offre une perspective sur l'avenir du commerce électronique axé sur la personnalisation tout en témoignant des avancées technologiques qui façonnent notre monde commercial. nous espérons que cette plateforme contribuera davantage à stimuler l'innovation et la créativité dans le monde du commerce en ligne.

Chapitre 1

E-Commerce et Print on Demand

1.1 Introduction

L'avènement de la technologie Web a révolutionné notre façon de vivre et nos activités commerciales. Dans ce contexte, trois concepts clés ont émergé : E-service, E-commerce et Print on Demand (POD). Ces concepts ont profondément transformé la commercialisation traditionnelle et ont ouvert de nouvelles opportunités pour les fournisseurs et les demandeurs des services.

Dans ce chapitre, nous aborderons ces trois concepts, et nous expliquerons chaque concept séparément et sa relation avec notre plateforme.

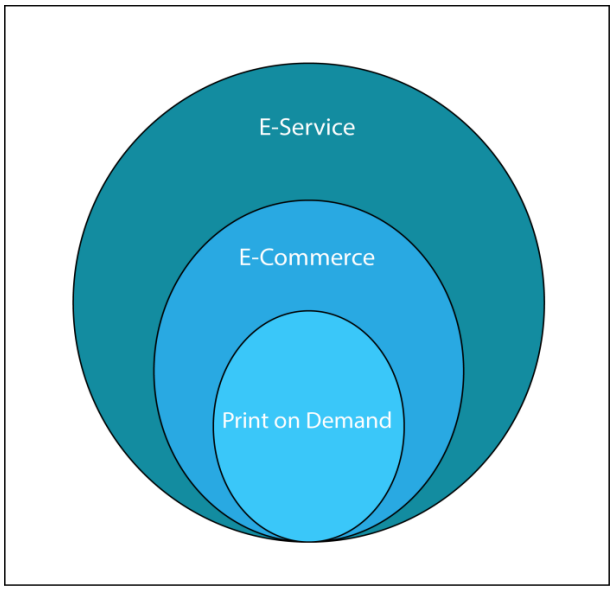

FIGURE 1.1 – relation entre l'e-service, e-commerce et Print on Demand (POD)

1.2 E-Service

1.2.1 Définition

Le terme **E-service** fait référence à un service fourni a distance par le biais d'Internet ou d'autres technologies numériques. Il s'agit d'un service qui est accessible, utilisé ou livré en ligne. Les e-services peuvent être fournis par des entreprises, des organismes gouvernementaux, des établissements d'enseignement ou d'autres entités pour faciliter diverses activités et interactions en ligne (4).

L'avantage principal des e-services est la possibilité d'accéder aux services à tout moment et de n'importe où, en utilisant des dispositifs connectés à Internet, tels que des ordinateurs, des smartphones ou des tablettes. Les e-services offrent souvent une plus grande commodité, une rapidité de traitement et une accessibilité accrue pour les utilisateurs.

1.2.2 Types de E-Service

Il existe de nombreux types de services en ligne qui sont disponibles aujourd'hui. Voici quelques-uns des principaux types de services électroniques :

E-Commerce

Ces services englobent l'achat et la vente de produits en ligne. Les places de marché en ligne, les boutiques en ligne et les systèmes de paiement en ligne sont des exemples de services de e-commerce.

Services de video a la demande

Ces services permettent aux utilisateurs de diffuser et de consommer des contenus multimédias en ligne, tels que des films, des séries télévisées, de la musique, des podcasts, etc. Des exemples courants incluent Netflix, Spotify, YouTube, Twitch, etc.

Services bancaires en ligne

Ces services permettent aux utilisateurs de gérer leurs opérations bancaires à distance. Ils peuvent consulter leur solde, effectuer des virements, payer des factures, etc., en utilisant des services bancaires en ligne ou des applications mobiles.

Services de stockage en ligne

Ces services permettent aux utilisateurs de stocker et de sauvegarder leurs fichiers et leurs données en ligne. Les exemples populaires incluent Dropbox, Google Drive, etc.

Services de réservation en ligne

Ces services permettent aux utilisateurs de réserver des voyages, des hébergements, des restaurants, des billets de spectacle, etc., en ligne. Des plateformes telles que Booking.com, Airbnb, Expedia, OpenTable, etc., offrent ce type de services.

E-apprentissage (ou E-learning)

Ces services permettent aux utilisateurs de réserver des voyages, des hébergements, des restaurants, des billets de spectacle, etc., en ligne. Des plateformes telles que Booking.com, Airbnb, Expedia, OpenTable, etc., offrent ce type de services.

E-santé (ou E-healthcare)

Ces services offrent des consultations médicales en ligne, la prise de rendez-vous, la prescription de médicaments, etc. Ils permettent aux utilisateurs de recevoir des soins de santé à distance. Des exemples incluent Teladoc, Doctor on Demand, Amwell, etc.

E-gouvernement

Ces services impliquent des interactions en ligne entre les citoyens et les agences gouvernementales. Des exemples incluent la déclaration d'impôts en ligne, la demande de permis ou licences, et l'accès à des informations gouvernementales.

Services de recherche et d'information en ligne

Les bases de données en ligne, les bibliothèques numériques, les dépôts de recherche et les portails d'information offrent un accès à une large gamme d'informations et de ressources.

Ce ne sont là que quelques exemples parmi de nombreux types de services électroniques disponibles aujourd'hui. La technologie numérique a ouvert de nombreuses possibilités pour les utilisateurs afin d'accéder à une variété de services en ligne, simplifiant ainsi de nombreux aspects de notre vie quotidienne.

1.3 E-Commerce

L'e-commerce, ou commerce électronique, fait référence à l'achat et à la vente de produits et de services en ligne. Au cours des dernières décennies, l'e-commerce a connu une croissance exponentielle grâce aux avancées technologiques et à l'adoption généralisée d'Internet, dans les prochains titres on en saura plus sur le commerce électronique.

FIGURE 1.2 – Croissance des ventes du e-commerce de détail dans le monde 2021-2026

1.3.1 Définitions

L'e-commerce est le commerce de biens et de services sur Internet (5) :

— **Biens** : Sont des produits ou marchandises, ce sont des objets matériels qui peuvent être échangés, vendus ou stockés. Ils sont utilisés comme intermédiaires pour combler un besoin.

— **Services** : Les services sont immatériels et reposent sur la mise à disposition d'un savoir-faire technique ou intellectuel.

Le commerce électronique peut être défini comme une méthodologie commerciale moderne qui répond aux besoins des organisations, des commerçants et des consommateurs pour réduire les coûts tout en améliorant la qualité des biens et services et en augmentant la vitesse de prestation des services, en utilisant Internet (6).

Le commerce électronique est l'endroit où les transactions commerciales ont lieu via les réseaux de télécommunications, en particulier Internet(7).

1.3.2 Outils E-commerce

Site E-Commerce

Un site e-commerce peut être défini comme une plate-forme en ligne dédiée aux transactions commerciales (généralement des achats et des ventes) il facilite la transaction entre un acheteur et vendeur.

Entreprise E-Commerce

Une entreprise de commerce électronique est une entreprise qui génère des revenus en vendant des produits ou des services en ligne.Par exemple, une entreprise de commerce électronique peut vendre des logiciels, des vêtements, des articles ménagers ou des services de conception Web (5).

1.3.3 Types de Commerce Électronique

C2B (Consumer to Business)

Implique un commerce entre les consommateurs et les entreprises dans lequel les consommateurs décident ce qu'ils veulent payer et les vendeurs décident d'accepter ou non. Le business model C2B repose sur 3 acteurs : un consommateur agissant en tant que vendeur, une entreprise agissant en tant qu'acheteur et un intermédiaire s'occupant de la mise en relation entre vendeurs et acheteurs (8).

B2B (Business to Business)

Il s'agit de commerce électronique entre entreprises. Cette forme de commerce est plus ancienne que la précédente. Historiquement, elle s'appuie sur des solutions d'interconnexion de réseaux utilisant l'EDI (9).

Electronic data interchange (EDI) est le concept d'entreprises communiquant électroniquement des informations qui étaient traditionnellement communiquées sur papier, telles que des bons de commande, des préavis d'expédition et des factures.

C2C (Consumer to Consumer)

Ce type de commerce existait avant Internet (petites annonces entre particuliers). Internet lui donne une nouvelle dimension puisqu'il démultiplie les possibilités d'échanges et facilite la recherche d'un bien (9).

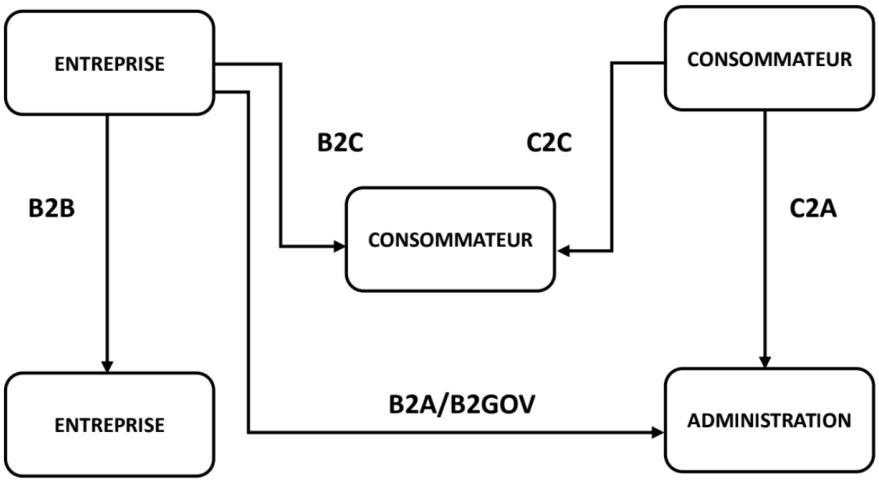

FIGURE 1.3 – Les relations commerciales

1.3.4 Avantages et défis du E-commerce

Le commerce électronique offre plusieurs avantages, mais il est important de considérer également les inconvénients et les défis associés à cette méthode d'achat en ligne.

Les avantages

- **Accessibilité** : le commerce électronique permet aux consommateurs d'acheter des produits ou des services à tout moment et de n'importe où, à condition qu'ils aient une connexion Internet.
- **Large choix de produits** : les consommateurs ont accès à une large gamme de produits et de services provenant de différentes régions du monde.
- **Prix compétitifs** : les prix des produits en ligne sont souvent moins chers que ceux proposés dans les magasins physiques.
- **Commodité** : e-commerce électronique permet aux consommateurs d'acheter des produits sans se déplacer ou faire la queue, ce qui est particulièrement pratique pour les personnes occupées.
- **Personnalisation** : le commerce électronique permet aux consommateurs de personnaliser leurs achats en fonction de leurs préférences.
- **Analyse de données** : Le commerce électronique offre un accès à des quantités massives de données client. En analysant ces données, les entreprises peuvent obtenir des informations sur les préférences, le comportement et les habitudes d'achat des clients. Cela permet de mettre en place des campagnes marketing personna-

lisées, des recommandations de produits ciblées et une expérience d'achat plus adaptée.
— **Portée mondiale et expansion du marché** : Le commerce électronique permet aux entreprises d'atteindre une audience mondiale et d'élargir leur base de clients au-delà des frontières géographiques. Il élimine les limitations des magasins physiques, permettant aux entreprises de fonctionner 24h/24 et d'atteindre des clients du monde entier.

Les défis

— **Absence de contact physique** : contrairement aux magasins physiques, les consommateurs ne peuvent pas toucher et tester les produits avant de les acheter
— **Frais de livraison** : Les frais de livraison peuvent être un inconvénient pour les consommateurs qui cherchent à économiser de l'argent
— **Risque de fraude** : il y a toujours un risque de fraude en ligne, en particulier lorsque les consommateurs achètent à partir de sites peu connus.
— **Besoin d'une connexion Internet** : le commerce électronique nécessite une connexion Internet, ce qui peut être un inconvénient pour les consommateurs qui n'ont pas accès à Internet.
— **Problèmes de sécurité** : les transactions en ligne peuvent être vulnérables aux pirates informatiques, aux attaques de phishing et aux virus.

1.4 Service On Demand

Le service à la demande est un concept innovant qui a révolutionné la façon dont nous accédons à divers produits et services, nous l'expliquerons en détail dans les prochains titres.

1.4.1 Définitions

Définition I

Dans le domaine de l'informatique, le service à la demande désigne un modèle de prestation de services informatiques où le client peut demander et recevoir des services spécifiques au besoin, sans avoir à investir dans et à maintenir leur propre infrastructure informatique. Les services sont généralement fournis sur Internet ou un réseau, et peuvent inclure des choses comme le cloud computing, storage, la sécurité réseau et les software applications (10).

Dans l'ensemble, le service à la demande est un moyen flexible et efficace pour les organisations d'accéder aux services informatiques dont elles ont besoin, sans les tracas et les frais de gestion de leur propre infrastructure informatique.

Définition II

Le service à la demande est un modèle commercial qui est de plus en plus populaire dans l'industrie informatique, en particulier avec l'avènement de l'informatique en nuage et d'autres services basés sur Internet. Le concept de service à la demande repose sur l'idée que les organisations peuvent bénéficier de l'externalisation de leurs besoins en matière d'informatique à des fournisseurs tiers spécialisés dans la prestation de services informatiques spécifiques.

Le service à la demande est généralement fourni sous forme de modèle par abonnement, où les clients paient des frais récurrents pour accéder aux services dont ils ont besoin. Les services peuvent être accessibles via un portail Web ou des interfaces de programmation d'applications (API), ce qui permet aux clients d'intégrer facilement les services dans leurs systèmes informatiques existants (11).

1.4.2 Types de services à la demande

Dans le domaine des technologies de l'information (IT), les types de services à la demande peuvent inclure :

Cloud computing

Les entreprises peuvent fournir des services de cloud computing à la demande pour permettre aux clients de stocker et d'accéder à leurs données et applications sur des serveurs distants (12).

Infrastructure-as-a-Service (IaaS)

Les entreprises peuvent fournir des services d'infrastructure à la demande pour permettre aux clients de louer des ressources informatiques, telles que des serveurs, des réseaux et des espaces de stockage, sur demande.

Platform-as-a-Service (PaaS)

Les entreprises peuvent fournir des services de plateforme à la demande pour permettre aux clients de développer, de tester et de déployer des applications sur une plateforme cloud sans avoir à gérer l'infrastructure sous-jacente.

Software-as-a-Service (SaaS)

Les entreprises peuvent fournir des services de logiciel à la demande pour permettre aux clients d'utiliser des applications sur le cloud via une interface web ou une API.

Security-as-a-Service (SECaaS)

Les entreprises peuvent fournir des services de sécurité à la demande pour permettre aux clients de protéger leurs systèmes et leurs données contre les menaces en ligne, telles que les attaques de logiciels malveillants et les violations de données.

Data-as-a-Service (DaaS)

Les entreprises peuvent fournir des services de données à la demande pour permettre aux clients d'accéder à des données en temps réel à partir de différentes sources.

IT Support-as-a-Service

Les entreprises peuvent fournir des services de support informatique à la demande pour aider les clients à résoudre des problèmes liés à leur infrastructure ou à leurs applications.

Production-as-a-Service

Offre une solution flexible et rentable pour la production de produits personnalisés ou spécifiques, tout en réduisant les coûts de stockage et de gestion de l'inventaire.

Il existe bien sûr de nombreux autres types de services à la demande dans le domaine de lT, en fonction des besoins spécifiques des clients et des entreprises.

1.5 Print on demand (POD)

C'est une service qui fusionnée deux types de service a la demande (SaaS et la Production-as-Service).

1.5.1 Définitions

Définition I

Le Print-on-Demand (POD) ou impression à la demande en français, est un modèle de production et de vente de produits, où les articles ne sont imprimés qu'une fois qu'une commande est passée. Cela signifie qu'il n'y a pas de stockage ou de production en masse de produits, et que chaque article est imprimé et fabriqué individuellement en fonction de la demande (13).

Définition II

le Print-on-Demand est un modèle de production et de vente de produits qui permet de produire des articles individuels en fonction de la demande, offrant ainsi plus de flexibilité et de réduction des coûts tout en réduisant les déchets et les coûts environnementaux.

Définition III

Le Print on Demand, aussi appelé POD ou impression à la demande, est un modèle commercial qui vous permet de vendre vos propres designs personnalisés sur différents types de produits (T-shirts, mugs, tote bags, casquettes…) en marque blanche sous votre propre marque. Votre fournisseur de print on demand procède à l'impression de votre design sur un produit uniquement après réception de la commande. Cela signifie que vous ne payez pas le produit avant de l'avoir vendu, ce qui vous évite d'avoir du stock et de réaliser un investissement initial (14).

Définition IV

Le Print On Demand, souvent abrégé POD, offre la possibilité de produire des articles personnalisés comportant des motifs, des textes et des designs spécifiques. Il s'agit d'un processus de personnalisation (15).

Si on décompose le mot Print On Demand :
— **Print** : Impression.
— **On Demand** : Sur demande.

Le Print on Demand (POD) est une méthode d'impression qui permet de produire des articles sur demande, sans nécessiter de gestion de stock. Il existe une grande diversité de produits pouvant être imprimés :*t-shirt, chemise, sweat, casquette, veste, legging, mug, coque de téléphone, affiche, tableau,...*

1.5.2 Cible du Print On Demand

Le print on demand (POD) est une méthode d'impression à la demande qui permet aux entreprises et aux particuliers de produire des produits personnalisés tels que des t-shirts, des sacs, des tasses, des casquettes, des posters, etc., en quantité limitée ou unitaire.

Le POD s'adresse principalement aux personnes et entreprises qui souhaitent vendre des produits personnalisés sans avoir à investir dans des stocks importants ou à supporter les coûts initiaux de production.

Les utilisateurs typiques du POD peuvent être des artistes, des créateurs de mode, des blogueurs, des influenceurs, des entrepreneurs, des petites entreprises ou des associations qui souhaitent offrir des produits personnalisés à leurs clients ou à leur communauté.

Le POD est également populaire auprès des start-ups, car il permet de tester rapidement la demande pour un produit sans investir de grandes sommes d'argent dans la production ou les stocks.

En résumé, le POD s'adresse à toute personne ou entreprise qui souhaite créer des produits personnalisés sans avoir à investir dans des stocks importants ou à supporter les

coûts initiaux de production.

1.5.3 Fonctionnement du Print On Demand

Le fonctionnement de l'impression est assez similaire à ce que vous avez l'habitude de voir avec les imprimantes. Ce sont simplement les machines utilisées qui sont différentes. Il existe plusieurs types d'impression :

Impression numérique

Cette technique permet de personnaliser des produits avec des visuels complexes, sans limite de couleurs sur du textile clair et foncé. La customisation tient très bien au lavage.

FIGURE 1.4 – Operation d'une impression numérique

Sérigraphie

Très populaire pour faire de l'impression sur le textile, la sérigraphie utilise un système de dépôt d'encre. En revanche, elle ne donne pas la possibilité d'imprimer des visuels complexes avec beaucoup de couleurs.

FIGURE 1.5 – Operation de Sérigraphie

Flocage / Flex

Cette technique d'impression se limite à une à deux couleurs. Le flocage est très populaire dans le domaine sportif. Les noms et les numéros des maillots de joueurs sont imprimés avec la technique du flocage.

FIGURE 1.6 – Operation de Flocage/Flex

Broderie

C'est de l'impression haut de gamme. La qualité est souvent supérieure avec cette technique. La broderie permet d'apporter un côté artisanal au produit avec du relief sur le textile.

Dès lors que vous envoyez une commande, l'imprimante va scanner votre design et l'imprimer sur votre produit.

FIGURE 1.7 – Operation de Broderie.

Avec le POD, vous êtes certains de vendre vos propres produits et votre propre marque ! Le design et les visuels que vous allez faire imprimer seront uniques et originaux ; de quoi vous démarquer facilement de la concurrence pour réussir dans la vente en ligne. Surtout, vos concurrents ne peuvent pas vendre les produits que vous créez !

Quand vous allez imprimer des articles, que ce soit des t-shirts personnalisés avec un dessin, des mugs avec un logo, et autres, votre design sera protégé par les droits d'auteurs.

1.5.4 Produits pour faire du Print On Demand

En POD, vous avez de nombreuses possibilités.Bien sûr, cela va aussi dépendre de votre choix de prestataire. En règle générale, vous pouvez faire de l'impression à la demande sur une large gamme de textile (t-shirt, sweat, casquette, etc.) et accessoires (mug, tableau, poster, sac, coque, bijou, coussin, stylo, etc.). Le choix est assez vaste !

«Les produits sur lesquels vous imprimez sont déjà créés pour la plupart. Vous ne pouvez pas les modifier»

Pour un tee-shirt, par exemple, vous n'avez pas la possibilité de rajouter une poche au niveau de la poitrine. Idem pour un mug, la forme ne peut être modifiable.

1.5.5 Avantages du POD

Aucun stock et aucune logistique

A l'inverse du e-commerce traditionnel ou l'e-commerçant dispose de son propre stock et de ses produits,le Print On Demand a le même fonctionnement que le dropshipping. Vous créez le produit dès que vous le vendez.C'est à dire que l'impression sur textile et l'achat chez le fournisseur vont se faire uniquement lorsque vous aurez une commande sur votre boutique en ligne.

Faible investissement

Du moment où vous ne gérez aucun produit, que vous n'avez pas d'entreposage, aucune logistique, etc. les coûts baissent très rapidement.

Ce modèle économique du "Juste à temps", vous permet de vous lancer dans le POD sans un investissement important. Les seuls frais que vous allez avoir seront ceux pour la boutique et les leviers web marketing. Bien entendu, il y aura aussi l'achat des produits lorsque vous les aurez vendus, mais vous aurez déjà touché l'argent de votre client.

Secteur niché

L'e-commerce est concurrencé mais une boutique de Print On Demand vous permet de vous démarquer de la concurrence et de vendre vos propres produits, Des produits uniques disponibles sur aucune autre boutique.

Avec le POD, vous proposez à vos clients des produits originaux qui sortent de l'ordinaire. Et plus c'est ciblé, plus vous avez de chances de réussir !

Produit personnalisé sur-mesure

Avec le POD, vous pouvez facilement imaginer de personnaliser vos produits selon le choix de vos clients.Dans vos designs, vous pouvez, par exemple, changer des mots en ajoutant un prénom, une date, etc. Libre cours à votre imagination !

Rapidité d'exécution

L'avantage, avec le POD, c'est la rapidité d'exécution. En effet, dès que vous avez une idée et un design, vous pouvez mettre en avant vos produits sur votre boutique.

Vous voulez rebondir sur une tendance ou un événement particulier ? Aucun problème , Créez le design, importez-le sur votre boutique et vendez-le. C'est aussi rapide que ça.

Livraison rapide

La plupart des fournisseurs et prestataires en Print On Demand que vous allez trouver sur le marché proposent une livraison très rapide en quelques jours.

Oubliez les délais de livraison de 3 semaines.Mettez en avant sur votre shop que vous livrez en 2-3 jours et soyez en fier !

1.5.6 Défis du POD

Bien que le print on demand (POD) présente de nombreux avantages, il existe également des défis à relever pour les entreprises qui l'utilisent. Voici quelques-uns des défis courants du POD :

Concurrence accrue

Le POD est de plus en plus populaire auprès des entrepreneurs et des petites entreprises, ce qui signifie que la concurrence est de plus en plus forte. Pour réussir dans ce marché, les entreprises doivent se démarquer en proposant des designs uniques, de haute qualité et en offrant un excellent service client.

Coûts élevés

Les coûts de production pour les produits POD peuvent être plus élevés que pour les produits en gros. Bien que le POD permette de produire des quantités plus petites de produits, les coûts unitaires peuvent être plus élevés. Les entreprises doivent donc fixer des prix de vente appropriés pour leurs produits pour couvrir leurs coûts tout en restant compétitives.

Contrôle qualité

Comme les produits POD sont produits à la demande, il est important que les entreprises maintiennent un contrôle qualité rigoureux pour éviter les erreurs d'impression ou les produits défectueux. Les entreprises doivent travailler en étroite collaboration avec leurs fournisseurs pour garantir que les produits répondent aux normes de qualité.

Gestion des stocks

Le POD élimine le besoin de stocker de grandes quantités de produits, mais les entreprises doivent être en mesure de gérer leur inventaire et de maintenir des niveaux de stock appropriés pour répondre à la demande. Les entreprises doivent également surveiller les délais de production et de livraison pour s'assurer que les produits sont disponibles lorsque les clients en ont besoin.

En résumé, le POD offre des avantages pour les entreprises qui cherchent à créer des produits personnalisés sans investir dans des stocks importants, mais il y a également des défis à relever, tels que la concurrence accrue, les coûts élevés, le contrôle qualité et la gestion des stocks. Les entreprises doivent être prêtes à relever ces défis pour réussir sur le marché du POD.

1.6 Conclusion

Dans ce chapitre, nous avons présenté une vue d'ensemble d'e-commerce, qui est une partie intégrante des e-services. Nous avons également mis en lumière et mené une étude approfondie sur l'impression à la demande (POD), qui est le noyau de notre étude et la base de notre projet.

Dans le prochain chapitre, nous allons fournir un aperçu complet et approfondi du Web 2.0.

Chapitre 2

Le WEB 2.0

2.1 Introduction

Ce chapitre offre une vue d'ensemble du World Wide Web et de son évolution, ainsi que des principales techniques et outils utilisés dans le développement web. Nous examinerons les fondements du web, depuis sa genèse historique jusqu'à son architecture actuelle.

L'importance du World Wide Web dans notre société moderne est indéniable. Il a transformé la façon dont nous accédons et partageons l'information, passant d'un système utilisé initialement par les scientifiques pour échanger des articles de recherche à une plateforme mondiale accessible à tous. Nous étudierons les éléments clés de son architecture, tels que les navigateurs web, les serveurs, les protocoles et les normes qui permettent l'échange d'informations sur Internet. De plus, nous aborderons les différents aspects de la communication entre les clients et les serveurs, ainsi que les technologies web, les modes de communication synchrones et asynchrones, et les tendances futures.

Ce chapitre jettera les bases nécessaires à une compréhension approfondie du fonctionnement du web, de son architecture et des technologies clés qui l'animent. Nous explorerons également les implications actuelles et les évolutions à venir dans le domaine de la communication web.

2.2 Le Web et de son importance

2.2.1 Contexte historique et évolution

Le World Wide Web (WWW) est un système d'information mondial qui relie des milliards d'appareils à travers le monde. Il s'agit d'un système de documents hypertexte interconnectés accessibles via Internet. Le WWW a été inventé par *Tim Berners-Lee* en 1989 alors qu'il travaillait au CERN, une organisation européenne de recherche.

Le WWW repose sur le concept d'hypertexte, qui permet aux utilisateurs de naviguer à travers des documents en cliquant sur des liens. Ces liens peuvent être du texte, des images ou d'autres types de médias. Le WWW est constitué de sites web, qui sont des collections de documents hypertexte liés.

L'histoire du WWW peut être divisée en trois phases principales (16) :

— **Les premières années (1989-1993)** : Pendant cette période, le WWW a été développé par Tim Berners-Lee et un petit groupe de collaborateurs au CERN. Le premier site web a été créé en 1991, et le WWW a été ouvert au public en 1993.

— **Les années de croissance (1994-2000)** : Pendant cette période, le WWW a connu une croissance rapide. Le nombre de sites web a augmenté de manière exponentielle, et le WWW est devenu un outil populaire pour la communication, l'éducation et les divertissements.

— **L'ère moderne (2001-présent)** : Le WWW a continué à se développer et à évoluer à l'ère moderne. De nouvelles technologies, telles que les médias sociaux et les appareils mobiles, ont rendu le WWW encore plus accessible et convivial.

Le WWW a eu un impact profond sur notre manière de vivre et de travailler. Il a rendu possible l'accès à l'information de n'importe où dans le monde et a changé notre façon de communiquer les uns avec les autres. Le WWW continue d'évoluer et il est susceptible d'avoir un impact encore plus important sur nos vies à l'avenir.

Voici quelques-unes des caractéristiques clés du WWW :

— **Hypertexte** : Le WWW est basé sur le concept d'hypertexte, qui permet aux utilisateurs de naviguer à travers des documents en cliquant sur des liens. Ces liens peuvent être du texte, des images ou d'autres types de médias.

  Il existe différents types d'hypertextes comme par exemple :

  — **Hypertexte linéaire** : Il s'agit du type le plus simple d'hypertexte, et il est constitué d'une séquence unique de pages liées les unes aux autres de manière linéaire.
  — **Hypertexte axial** : Ce type d'hypertexte a un noyau central de pages, et d'autres pages sont liées à ce noyau de manière radiale.
  — **Hypertexte en réseau** : Ce type d'hypertexte présente un réseau complexe de pages liées entre elles de manière non linéaire.
  — **Hypertexte en couches** : Ce type d'hypertexte comporte plusieurs couches de pages, et chaque couche est liée aux autres couches de manière non linéaire.

— **Navigateurs web** : Les navigateurs web sont des applications logicielles qui permettent aux utilisateurs d'accéder et de visualiser des sites web. Les navigateurs web les plus populaires incluent Chrome, Firefox et Edge.
— **Serveurs web** : Les serveurs web sont des ordinateurs qui stockent et fournissent des sites web. Lorsqu'un utilisateur demande un site web, le serveur web envoie les fichiers du site à l'ordinateur de l'utilisateur.
— **Protocoles** : Les protocoles sont des règles qui régissent la communication entre les ordinateurs sur le WWW. Le protocole le plus important pour le WWW est le protocole de transfert hypertexte **HTTP**, (17) (18).

Le WWW est un outil puissant qui a changé notre façon de vivre et de travailler. Il est susceptible d'avoir un impact encore plus important sur nos vies à l'avenir.

2.2.2 Architecture Web

Cette partie explore l'architecture du web, la base de l'échange d'informations sur Internet. Nous examinons les composants, les couches et l'évolution historique de cette architecture. Cela permet de comprendre les fondements et les concepts clés nécessaires pour créer et interagir avec les sites web modernes.

2.3 Architecture Client-Serveur

L'architecture client-serveur est un type d'architecture d'application distribuée qui repose sur deux composants principaux distincts appelés serveurs et clients. Les serveurs fournissent les ressources, tandis que les clients envoient des requêtes. Les deux composants communiquent via un réseau informatique en utilisant des protocoles et des modes de communication convenus.

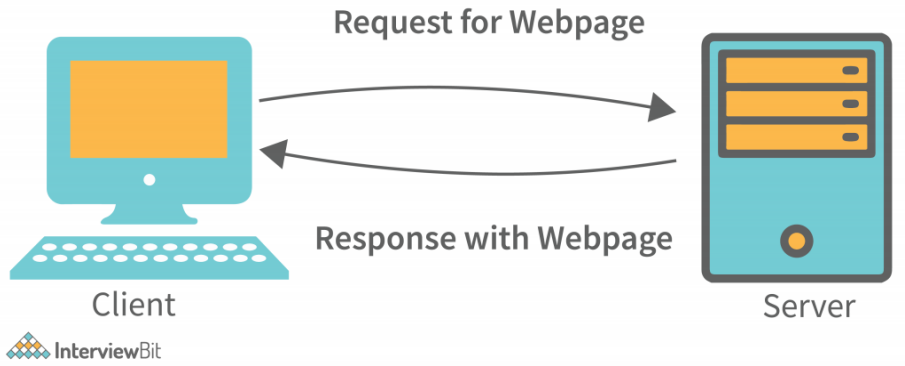

FIGURE 2.1 – Architecture client-serveur (1).

2.3.1 Modèle Requête-Réponse

Le modèle de *requête-réponse* est un schéma d'échange de messages dans lequel **un client** envoie une requête à **un serveur**, puis le serveur reçoit cette requête, la traite et renvoie une réponse au client. Dans ce schéma, le client est l'initiateur de la communication et généralement, l'échange se fait de manière *synchrone*, bien que des communications *asynchrones* puissent également être utilisées. La réponse du serveur peut varier, allant d'une simple confirmation à un objet de données complexe.

2.3.2 Statelessness et gestion des sessions

La **statelessness** (absence d'état) est une caractéristique des systèmes dans lesquels aucune information ou état des interactions passées avec les clients n'est conservé. Dans le domaine du développement web, la statelessness se réfère au traitement de chaque demande d'un client vers un serveur comme une transaction indépendante et isolée, sans rétention d'informations sur les requêtes précédentes.

Dans une application web sans état, le serveur ne stocke aucune donnée ou information de session relative au client entre les requêtes. Cette approche diffère des applications avec état, où le serveur conserve des données de session ou des informations spécifiques à l'utilisateur.

Pour maintenir l'état de l'utilisateur et permettre des fonctionnalités telles que des expériences personnalisées ou l'authentification, des techniques de gestion de session sont

mises en œuvre. **La gestion de session** consiste à créer et gérer une session pour chaque utilisateur, souvent à l'aide *d'identifiants de session* ou de *cookies*. Le serveur génère un identifiant de session unique pour chaque client et l'associe à ses données de session. Cet identifiant de session est renvoyé au client, généralement sous la forme d'un cookie, et le client le transmet lors des requêtes ultérieures. Le serveur peut ainsi utiliser cet identifiant de session pour récupérer les données associées et offrir une expérience cohérente à travers les requêtes.

2.3.3 Protocoles Web

Les protocoles Web sont un ensemble de règles qui régissent la manière dont les ordinateurs communiquent entre eux sur internet. Ils définissent comment les données sont formatées, transmises et reçues. Sans les protocoles Web, l'internet ne pourrait pas fonctionner.

Certains des protocoles web les plus courants comprennent :

— **HTTP (Hypertext Transfer Protocol)** : C'est le protocole utilisé pour transférer les pages web et autres documents hypertextes sur internet.
— **HTTPS (Hypertext Transfer Protocol Secure)** : Il s'agit d'une version sécurisée de HTTP qui utilise le chiffrement pour protéger les données contre l'interception.
— **TCP (Transmission Control Protocol)** : C'est un protocole qui garantit la livraison des données dans le bon ordre.
— **UDP (User Datagram Protocol)** : C'est un protocole plus rapide que TCP, mais il ne garantit pas la livraison des données.
— **FTP (File Transfer Protocol)** : C'est un protocole utilisé pour transférer des fichiers entre des ordinateurs sur internet.
— **UDP (User Datagram Protocol)** : C'est un protocole plus rapide que TCP, mais il ne garantit pas la livraison des données.

Ces protocoles jouent un rôle essentiel dans le bon fonctionnement d'Internet en facilitant l'échange d'informations entre les appareils connectés au réseau. Ils sont responsables de la transmission sécurisée des données, du transfert de fichiers et de l'envoi de courriers électroniques, contribuant ainsi à la communication et à l'accessibilité à l'information à l'échelle mondiale.

2.4 Protocole de transfert hypertexte (HTTP)

Le protocole HTTP (Hypertext Transfer Protocol) est un protocole de communication utilisé pour transférer des données sur le Web. Il joue un rôle fondamental dans l'échange d'informations entre les clients (navigateurs web) et les serveurs web, permettant ainsi de visualiser des pages web, d'envoyer des formulaires, de télécharger des fichiers, et bien

plus encore.

HTTP est construit autour d'une philosophie simple mais puissante, connue sous le nom de principe **REST** (**Re**presentational **S**tate **T**ransfer). Ce principe stipule que chaque ressource (par exemple, une page web ou un fichier) doit avoir une URL unique, et que les actions sur ces ressources doivent être définies de manière uniforme, utilisant les méthodes HTTP standard telles que *GET, POST, PUT et DELETE*.

Au fil du temps, HTTP a connu plusieurs évolutions pour améliorer ses performances et répondre aux besoins croissants du Web. Les versions les plus couramment utilisées sont *HTTP/1.1* et *HTTP/2*. *HTTP/1.1* a introduit la possibilité de maintenir des connexions persistantes et de gérer les en-têtes pour améliorer l'efficacité des transferts de données. Quant à HTTP/2, il a apporté des améliorations significatives en matière de multiplexage des flux, de compression des en-têtes et d'autres optimisations pour accélérer le chargement des pages.

Aujourd'hui, le protocole HTTP continue d'être le fondement de la communication sur le Web et reste un élément clé pour l'accès et l'échange d'informations en ligne. Avec l'évolution constante du Web, de nouvelles versions et améliorations de HTTP continuent d'être développées pour répondre aux besoins émergents du monde numérique (19) (20) (21).

2.4.1 Méthodes de requête HTTP (GET, POST, etc.)

Les *verbes HTTP*, également appelés *méthodes HTTP*, sont utilisés dans le protocole HTTP pour interagir avec les ressources du Web. Ils permettent aux clients (comme les navigateurs web) de communiquer avec les serveurs pour récupérer, envoyer, mettre à jour ou supprimer des ressources(22).

Verbe HTTP	Utilisation
GET	Récupérer une ressource
POST	Créer une nouvelle ressource
PUT	Mettre à jour une ressource
DELETE	Supprimer une ressource
HEAD	Récupérer les en-têtes d'une ressource
OPTIONS	Indiquer les méthodes HTTP supportées pour une ressource
TRACE	Tracer la chaîne de requête-réponse

TABLE 2.1 – Les Verbes HTTP

2.4.2 En-têtes HTTP et codes d'état

Les en-têtes HTTP et les codes d'état sont des éléments cruciaux dans le fonctionnement du protocole HTTP (Hypertext Transfer Protocol). Voici une explication de chacun :

En-têtes HTTP :

Les en-têtes HTTP sont des informations supplémentaires envoyées avec une requête HTTP ou une réponse HTTP. Ils fournissent des détails sur la requête ou la réponse, ce qui permet aux serveurs et aux navigateurs de comprendre comment traiter les données qui sont échangées.

Voici quelques exemples d'en-têtes HTTP couramment utilisés :

— **User-Agent** : Identifie le logiciel client (comme un navigateur web) qui envoie la requête.
— **Content-Type** : Indique le type de contenu (comme du texte, des images, etc.) dans la requête ou la réponse.
— **Content-Length** : Spécifie la taille du contenu dans la requête ou la réponse en octets.
— **Authorization** : Utilisé pour inclure des informations d'identification (comme un nom d'utilisateur et un mot de passe) dans la requête.
— **Cookie** : Contient des informations sur les sessions utilisateur et les préférences stockées sur le navigateur.
— **Cache-Control** : Contrôle la mise en cache des ressources, spécifiant comment elles doivent être stockées et pour combien de temps.
— **Location** : Utilisé dans les réponses pour rediriger le client vers une autre URL.
— **Accept** : Indique les types de contenu que le client est capable de recevoir.

Codes d'état HTTP :

Les codes d'état HTTP sont des nombres à trois chiffres inclus dans les réponses HTTP. Ils indiquent l'état de la requête et du serveur au client.

Voici quelques-uns des codes d'état HTTP les plus courants :
— **200 OK** : La requête a été traitée avec succès.
— **201 Created** : Une nouvelle ressource a été créée suite à la requête.
— **204 No Content** : La requête a été traitée avec succès, mais il n'y a pas de contenu à renvoyer.
— **301 Moved Permanently** : La ressource demandée a été déplacée de manière permanente vers une nouvelle URL.
— **400 Bad Request** : La requête du client est mal formée ou ne peut pas être traitée par le serveur.
— **404 Not Found** : La ressource demandée n'a pas été trouvée sur le serveur.
— **500 Internal Server Error** : Indique une erreur interne du serveur.
— **503 Service Unavailable** : Le serveur n'est pas prêt à traiter la requête. Il peut être temporairement surchargé ou en maintenance.

Ces codes d'état sont un moyen pour le serveur de communiquer avec le client sur le succès ou l'échec d'une requête, et ils sont essentiels pour le bon fonctionnement des interactions web.

2.4.3 HTTP sécurisé (HTTPS)

Les certificats TLS (Transport Layer Security) sont utilisés pour activer HTTPS (Hypertext Transfer Protocol Secure) sur les sites web, ce qui représente la version sécurisée du protocole HTTP. Lorsqu'un site web est accédé en utilisant HTTPS, les données échangées entre le client et le serveur sont chiffrées, ce qui les protège contre l'écoute indésirable et l'accès non autorisé

La principale différence entre HTTP et HTTPS réside dans l'utilisation du chiffrement pour protéger les données transférées entre le serveur web et le navigateur web. Cela rend HTTPS beaucoup plus sécurisé que HTTP. Lorsqu'un site web utilise HTTPS, les données sont chiffrées, ce qui signifie qu'elles sont codées et ne peuvent être lues que par le serveur web et le navigateur web légitimes. Cela garantit que les informations sensibles, telles que les mots de passe, les informations personnelles et les données de paiement, restent confidentielles et ne peuvent pas être compromises lors de leur transmission sur Internet.

Par exemple, lorsque vous effectuez un achat en ligne et saisissez vos informations de carte de crédit, HTTPS chiffre ces données avant de les envoyer au serveur web, assurant ainsi la confidentialité de vos informations financières. Les sites de médias sociaux, les banques en ligne, les sites de commerce électronique et d'autres sites web nécessitant la confidentialité des données utilisent HTTPS pour sécuriser les connexions.

Feature	**HTTP**	**HTTPS**
Protocol	Hypertext Transfer Protocol	Hypertext Transfer Protocol Secure
Encryption	No	Yes
Security	Less secure	More secure
Port	80	443

TABLE 2.2 – HTTP VS HTTPS

2.4.4 Chiffrement SSL/TLS

SSL et TLS sont des protocoles de communication qui chiffrent les données entre les serveurs, les applications, les utilisateurs et les systèmes. Ils authentifient deux parties connectées sur un réseau afin qu'elles puissent échanger des données en toute sécurité(23).

	SSL	**TLS**
Correspond à	SSL signifie **S**ecure **S**ockets **L**ayer	TLS signifie **T**ransport **L**ayer **S**ecurity
Historique des versions	SSL est maintenant remplacé par TLS. SSL est passé aux versions 1.0, 2.0 et 3.0	TLS est la version améliorée de SSL. TLS est passé aux versions 1.0, 1.1, 1.2 et 1.3
Utilisation	Chaque version de SSL est désormais obsolète.	Les versions 1.2 et 1.3 de TLS sont activement utilisées
Messages d'alertes	SSL n'a que deux types de messages d'alerte. Les messages d'alerte ne sont pas chiffrés	Les messages d'alerte TLS sont cryptés et plus diversifiés
Authentification des messages	SSL utilise des MAC.	TLS utilise des HMAC
Techniques de chiffrement	SSL prend en charge les algorithmes plus anciens avec des vulnérabilités de sécurité connues	TLS utilise des algorithmes de chiffrement avancés.

TABLE 2.3 – TLS vs SSL

2.4.5 Certification TLS

Les certificats TLS (Transport Layer Security) sont utilisés pour activer HTTPS (Hypertext Transfer Protocol Secure) sur les sites web, ce qui représente la version sécurisée du protocole HTTP. Lorsqu'un site web est accédé en utilisant HTTPS, les données échangées entre le client et le serveur sont chiffrées, ce qui les protège contre l'écoute indésirable et l'accès non autorisé (24) (25).

2.5 Communication Client(s)/Serveur(s)

2.5.1 URL (Uniform Resource Locator)

Une adresse URL (Uniform Resource Locator) est une référence vers une ressource sur Internet.

Une URL se compose de trois composants principaux :
Identifiant de protocole : Ceci spécifie le type de ressource. Les protocoles les plus courants sont HTTP et HTTPS. Nom d'hôte : Il s'agit du nom du serveur où se trouve la ressource. Chemin de la ressource : Ceci spécifie l'emplacement de la ressource sur le serveur.

Par exemple, l'URL **https ://www.google.com/** se compose des éléments suivants :

— **Identifiant de protocole** : https
— **Nom d'hôte** : www.google.com
— **Chemin de la ressource** : **/**

L'identifiant de protocole **https** indique que la ressource est sécurisée et que les données seront chiffrées pendant la transmission. Le nom d'hôte **www.google.com** spécifie le serveur où se trouve la ressource. Le chemin de la ressource **/** spécifie l'emplacement de la ressource sur le serveur (26).

Les URL sont utilisées par les navigateurs web pour accéder aux ressources sur Internet. Lorsque vous saisissez une URL dans un navigateur web, le navigateur envoie une requête au serveur et le serveur renvoie la ressource demandée (27).

Voici quelques avantages de l'utilisation des URL :

— **Facilité de mémorisation** : Les URL sont généralement courtes et faciles à mémoriser, ce qui facilite la recherche et l'accès aux ressources sur Internet.
— **Standardisation** : Les URL sont normalisées, ce qui signifie qu'elles sont les mêmes pour tout le monde. Cela facilite le partage des URL avec d'autres personnes.
— **Polyvalence** : Les URL peuvent être utilisées pour accéder à diverses ressources, telles que des pages web, des images, des vidéos et des fichiers.

Les adresses URL jouent un rôle essentiel dans le fonctionnement d'Internet en facilitant l'accès aux ressources en ligne. Elles permettent aux utilisateurs d'accéder rapidement et facilement à des sites web, des images, des vidéos et bien d'autres types de contenus sur le réseau mondial.

Grâce à la standardisation des URL, les navigateurs web peuvent interpréter les adresses et les acheminer vers les serveurs appropriés pour récupérer les données demandées. Cette infrastructure robuste contribue à l'interopérabilité et à la connectivité d'Internet, permettant aux utilisateurs du monde entier de partager des informations et d'accéder à des ressources diverses sans entraves.

Les URL sont également utilisées dans le référencement et l'indexation des sites web par les moteurs de recherche. Les moteurs de recherche parcourent le Web en suivant les liens des pages pour indexer leur contenu, permettant ainsi aux utilisateurs de rechercher des informations spécifiques et d'accéder aux pages pertinentes (28).

2.5.2 DNS (Domain Name System)

Le **Système de Noms de Domaine (DNS)** est un système de nommage hiérarchique distribué pour les ordinateurs, les services ou toute ressource connectée à Internet ou à un réseau privé. Il associe les noms de domaine à leurs adresses *IP* respectives.

Lorsque vous saisissez un nom de domaine dans votre navigateur Web, tel que **www.[site].com**, votre ordinateur envoie une requête DNS à un serveur DNS. Le serveur DNS recherche ensuite le nom de domaine dans sa base de données et renvoie l'adresse IP correspondante. Votre ordinateur utilise ensuite l'adresse IP pour se connecter au site Web.

Le DNS est une partie essentielle d'Internet. Il nous permet d'utiliser des noms de domaine au lieu d'adresses IP, qui sont beaucoup plus faciles à retenir. Le DNS contribue également à rendre Internet plus fiable et efficace.

Voici certains avantages du DNS :
— Il facilite la mémorisation des noms de domaine par les humains par rapport aux adresses IP.
— Il rend Internet plus fiable en fournissant un moyen de trouver la bonne adresse IP pour un nom de domaine, même si l'adresse IP change.
— Il rend Internet plus efficace en mettant en cache les requêtes DNS, de sorte qu'elles n'ont pas besoin d'être répétées à chaque visite d'un site Web.

Cependant, le DNS présente également certains inconvénients :
— Il peut être vulnérable aux attaques, telles que le détournement DNS et l'empoisonnement du cache DNS.
— Il peut être lent, en particulier si le serveur DNS est éloigné de l'ordinateur de l'utilisateur

En conclusion, le DNS est une partie cruciale d'Internet. Il facilite notre utilisation d'Internet et contribue à le rendre plus fiable et efficace(29)(2).

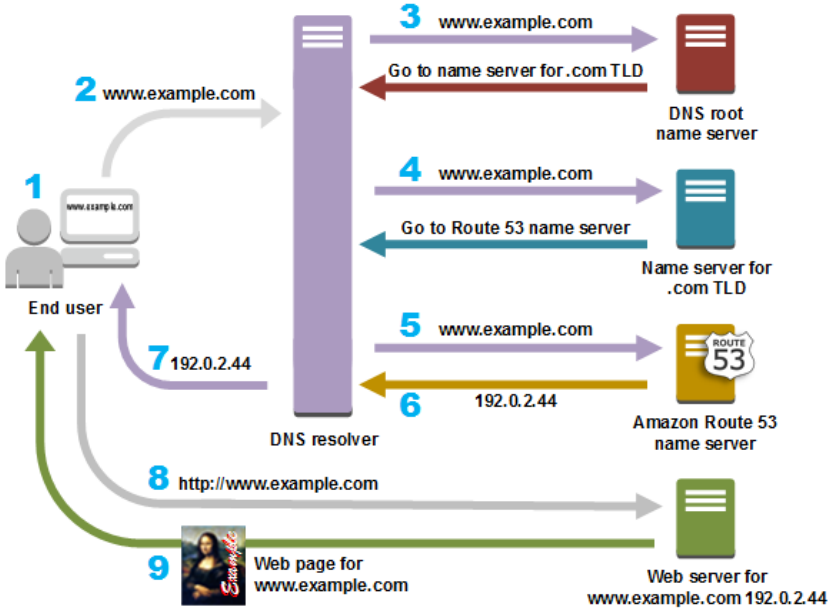

FIGURE 2.2 – Routage DNS (2).

2.5.3 Adressage IP

L'adressage IP est un système qui attribue des adresses uniques aux appareils d'un réseau. Ces adresses permettent aux appareils de communiquer entre eux et d'être identifiés par les autres appareils du réseau.

Les adresses IP sont des nombres de 32 bits qui sont généralement écrits en format décimal, avec quatre nombres séparés par des points. Par exemple, l'adresse IP 192.168.1.1 est un nombre de 32 bits qui peut être écrit comme **11000000 10101000 00000001 00000001** en format binaire.

Les adresses IP sont divisées en deux parties :
— l'identifiant du réseau (network ID) et l'identifiant de l'hôte (host ID)
— L'identifiant du réseau identifie le réseau sur lequel l'appareil se trouve, et l'identifiant de l'hôte identifie l'appareil sur le réseau.

Par exemple, l'adresse IP **192.168.1.1** a un identifiant du réseau de **192.168.1** et un identifiant de l'hôte de 1.

L'Autorité des Numéros Assignés sur l'Internet (IANA) est responsable de l'allocation des adresses IP aux organisations. Les organisations attribuent ensuite des adresses IP aux appareils de leurs réseaux.

Les adresses IP sont utilisées par les *routeurs* pour acheminer le trafic entre les réseaux. Lorsqu'un appareil envoie un paquet de données, le routeur examine l'adresse IP de destination dans le paquet et détermine à quel réseau le paquet doit être envoyé. Le routeur transfère ensuite le paquet au routeur suivant sur le chemin vers le réseau de destination.

Il est important de noter qu'il existe deux versions d'adresses IP :
— **IPv4** (Internet Protocol version 4) qui utilise des adresses de 32 bits
— **IPv6** (Internet Protocol version 6) qui utilise des adresses de 128 bits

IPv6 a été développé pour répondre à l'épuisement des adresses **IPv4** et offrir un espace d'adressage plus vaste pour soutenir la croissance d'Internet (30).

Composant	Description
ID réseau	Le réseau sur lequel se trouve l'appareil
ID d'hôte	L'appareil sur le réseau
Nombre total de bits	32 bits
Format décimal	Quatre nombres séparés par des points
Format binaire	Un nombre 32 bits

TABLE 2.4 – Les différents composants d'une adresse IP

2.5.4 TCP/IP (Transmission Control Protocol/Internet Protocol)

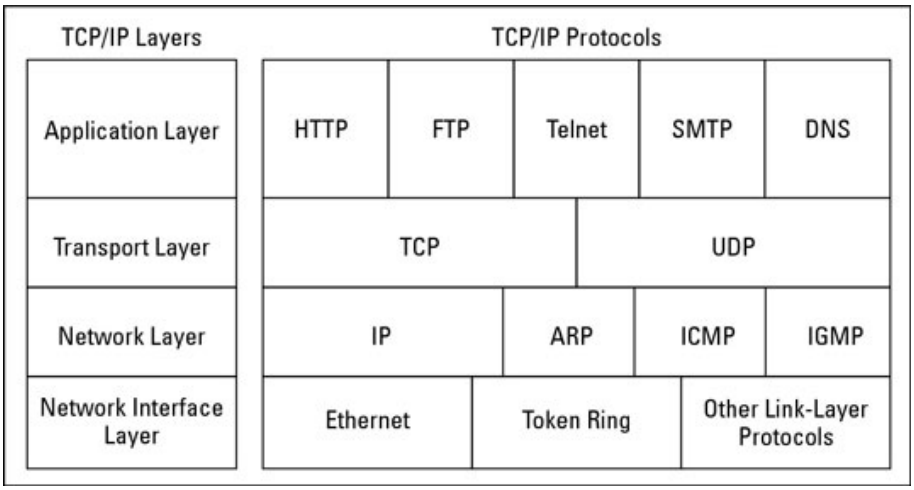

FIGURE 2.3 – Couches et protocoles TCP/IP

Le protocole TCP/IP (Transmission Control Protocol/Internet Protocol) est un ensemble de protocoles de réseau qui constituent la base de l'internet et de la plupart des réseaux informatiques modernes. Il assure une communication fiable entre les appareils connectés au réseau et permet la transmission des données sur l'internet.

TCP/IP est une suite de protocoles organisés en quatre couches : la couche Application, la couche Transport, la couche Internet et la couche d'Accès au réseau. Chaque couche possède son propre ensemble de protocoles qui travaillent ensemble pour permettre la communication entre les appareils.

Voici un bref aperçu de *la pile* de protocoles TCP/IP :

— **Couche Application** : C'est la couche supérieure de la pile TCP/IP et elle est responsable de la communication entre les applications fonctionnant sur différents appareils. Elle inclut des protocoles tels que HTTP (Hypertext Transfer Protocol) pour la navigation web, SMTP (Simple Mail Transfer Protocol) pour les e-mails et FTP (File Transfer Protocol) pour les transferts de fichiers.

— **Couche Transport** : La couche Transport est responsable de la communication de bout en bout et assure la livraison fiable des données. Elle comprend deux protocoles principaux : TCP (Transmission Control Protocol) et UDP (User Datagram Protocol). TCP assure une communication fiable et orientée connexion, tandis qu'UDP assure une communication non fiable et sans connexion.
— **Couche Internet** : La couche Internet est responsable du routage des paquets de données entre différents réseaux. Elle utilise le protocole IP (Internet Protocol) pour attribuer des adresses uniques aux appareils du réseau et déterminer le chemin le plus efficace pour la transmission des données.
— **Couche d'Accès au réseau** : La couche d'Accès au réseau est la couche la plus basse de la pile TCP/IP et elle est responsable de la transmission des données sur le réseau physique. Elle comprend des protocoles qui régissent la transmission des données sur Ethernet, Wi-Fi ou autres connexions physiques (31).

2.5.5 Protocole WebSocket

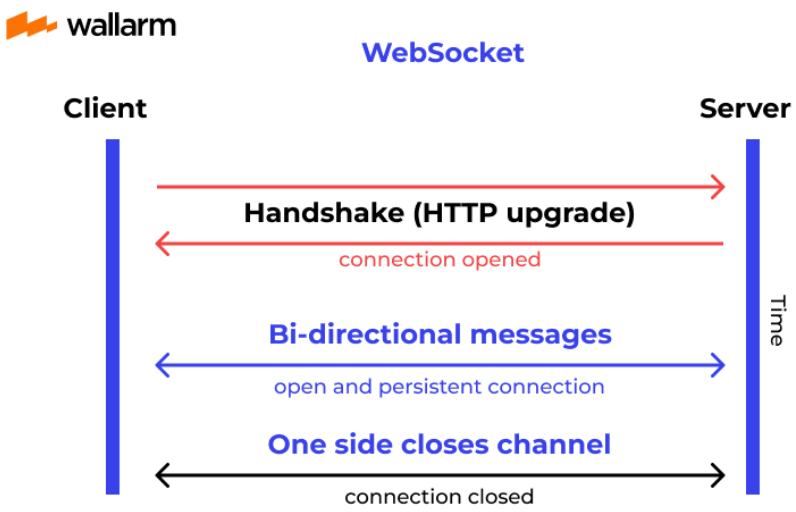

FIGURE 2.4 – Protocole WebSocket (3)

Le protocole **WebSocket** est un protocole de communication informatique qui offre des canaux de communication full-duplex sur une seule connexion TCP.

WebSocket est un protocole de communication en **temps réel**, **bidirectionnel** et **événementiel** qui permet des connexions persistantes entre les clients et les serveurs. Cela signifie que la connexion entre le client et le serveur reste ouverte même après le traitement de la requête initiale, et des messages peuvent être envoyés dans les deux sens à tout moment.

WebSocket est une méthode plus efficace pour communiquer avec un serveur que les méthodes HTTP traditionnelles, telles que le **polling** ou le **long polling**. Cela est dû au fait

que les connexions WebSocket sont persistantes, de sorte que le client et le serveur n'ont pas besoin d'ouvrir et de fermer constamment de nouvelles connexions.

WebSocket est utilisé dans diverses applications, notamment :
— Applications de chat en temps réel
— Applications de streaming média
— Applications de jeux en ligne
— Applications de trading financier

Voici quelques avantages de l'utilisation de WebSocket :
— **Efficacité accrue** : Les connexions WebSocket sont plus efficaces que les méthodes HTTP traditionnelles, car elles n'exigent pas que le client et le serveur ouvrent et ferment constamment de nouvelles connexions.
— **Performances améliorées** : Les connexions WebSocket peuvent offrir de meilleures performances que les méthodes HTTP traditionnelles, car elles permettent un transfert de données plus efficace.
— **Communication en temps réel** : Les connexions WebSocket peuvent être utilisées pour créer des applications de communication en temps réel, telles que des applications de chat et de streaming média.

Cependant, il y a aussi quelques inconvénients liés à l'utilisation de WebSocket :
— **Sécurité** : Les connexions WebSocket ne sont pas aussi sécurisées que les connexions HTTP traditionnelles, car elles peuvent être utilisées pour envoyer des données arbitraires entre le client et le serveur.
— **Complexité** : Les connexions WebSocket peuvent être plus complexes à mettre en œuvre que les connexions HTTP traditionnelles.

En résumé, WebSocket est un protocole puissant qui peut être utilisé pour créer une variété d'applications en temps réel. Cependant, il est important de prendre en compte les problèmes de sécurité et de complexité avant d'utiliser WebSocket.

Fonctionnalité	Protocole WebSocket	Protocole HTTP
Objectif	Fournit des canaux de communication full-duplex sur une seule connexion TCP.	Utilisé pour la communication client-serveur en mode requête-réponse.
Transfert de données	Les messages peuvent être envoyés dans les deux sens à tout moment.	Les messages sont envoyés dans un seul sens, du client au serveur ou du serveur au client.
État	Les connexions WebSocket sont stateful, ce qui signifie que la connexion reste ouverte même après que la demande initiale a été traitée.	Le réseau sur lequel se trouve l'appareil
Sécurité	Les connexions WebSocket peuvent être sécurisées à l'aide d'un cryptage.	Les connexions HTTP peuvent être sécurisées à l'aide du cryptage SSL/TLS.
Complexité	Les connexions WebSocket peuvent être plus complexes à implémenter que les connexions HTTP.	Les connexions HTTP sont relativement simples à implémenter.
Cas d'utilisation	Applications de chat en temps réel, applications de streaming media, applications de jeux en ligne, applications de trading financier.	Livraison de contenu statique, services Web, téléchargements de fichiers.

TABLE 2.5 – Les différences entre le protocole WebSocket et le protocole HTTP

2.5.6 API RESTful (Representational State Transfer)

REST signifie **Re**presentational **S**tate **T**ransfer, et c'est un style architectural pour la conception d'API web. Les API REST sont conçues pour être *sans-état*, *cacheables* et *auto-descriptives* :

- **Sans état** : Les API REST ne conservent aucun état entre les requêtes. Cela signifie que chaque requête doit contenir toutes les informations nécessaires pour la traiter. Cela rend les API REST évolutives et efficaces, car le serveur n'a pas besoin de suivre l'état de chaque client.
- **Cacheable** : Les API REST peuvent être mises en cache par les clients et les intermédiaires. Cela peut améliorer les performances, car les clients n'ont pas besoin de faire une nouvelle requête pour une ressource qui n'a pas changé depuis la dernière requête.
- **Auto-descriptives** : Les API REST utilisent des URI pour identifier les ressources. Ces URI doivent être significatifs et descriptifs, de sorte que les clients puissent comprendre les ressources auxquelles ils accèdent.

Les API REST sont un choix populaire pour les API web, car elles sont faciles à com-

prendre et à implémenter. Elles sont également évolutives et efficaces, ce qui les rend bien adaptées aux applications à grande échelle.

Voici quelques autres principes des **API RESTful** :

— **Interface uniforme** : Toutes les API RESTful doivent utiliser la même interface uniforme, quelle que soit l'implémentation sous-jacente. Cela facilite l'utilisation de différentes API RESTful par les clients.

— **Système en couches** : Les API RESTful peuvent être en couches, de sorte que différentes couches peuvent être implémentées indépendamment. Cela facilite la mise à l'échelle et la maintenance des API RESTful.

— **Code à la demande** : Les API RESTful peuvent être utilisées pour fournir du code aux clients. Cela peut être utilisé pour implémenter des fonctionnalités telles que le scripting côté client (32) (33).

2.5.7 GraphQL

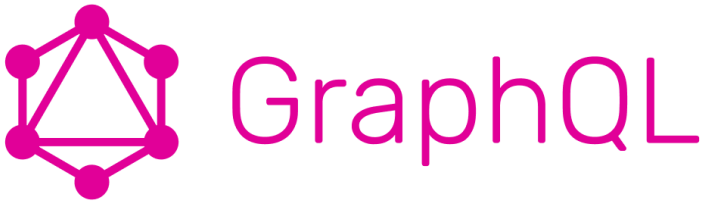

FIGURE 2.5 – GraphQL logo

Définition

GraphQL est un langage de requête et une spécification pour les API qui permet aux clients de demander exactement les données dont ils ont besoin, et rien de plus. Il a été créé par Facebook en 2012 pour résoudre les limitations des API REST traditionnelles en matière de surcharge de requêtes et de sur-fetching de données (34).

Principales caractéristiques de GraphQL

— **Récupération précise des données** : Les clients peuvent spécifier les champs exacts dont ils ont besoin, ce qui permet d'éviter le sur-fetching et de réduire la taille des réponses.

— **Résolution de données côté serveur** : Contrairement aux API REST qui déterminent la forme de la réponse côté serveur, GraphQL permet aux clients de spécifier la structure de la réponse, offrant ainsi une flexibilité accrue.

— **Typage fort** : GraphQL utilise un schéma pour définir les types de données disponibles et les relations entre eux, ce qui garantit une documentation complète et des validations de requêtes précises.

Cas d'utilisation

— **Applications mobiles** : Les applications mobiles ont souvent besoin de récupérer des données spécifiques pour une expérience utilisateur fluide, ce qui fait de GraphQL un choix idéal pour optimiser les requêtes.
— **Applications web** : GraphQL facilite la récupération de données spécifiques pour les interfaces utilisateur complexes et réactives.
— **Microservices** : Dans un environnement de microservices, GraphQL permet d'agréger des données de plusieurs services en une seule requête.

Avantages de GraphQL

— **Économie de bande passante** : Étant donné que les clients spécifient les données nécessaires, les réponses sont plus petites et plus efficaces.
— **Flexibilité des requêtes** : Les clients peuvent récupérer les données dont ils ont besoin en une seule requête, évitant ainsi les problèmes de surcharge.
— **Meilleure documentation** : Le schéma GraphQL fournit une documentation précise des types de données disponibles et de leurs relations.

Inconvénients de GraphQL

— **Courbe d'apprentissage** : Comprendre les concepts et le fonctionnement de GraphQL peut être complexe pour certains développeurs.
— **Complexité côté serveur** : L'implémentation de GraphQL peut être plus complexe que les API REST traditionnelles.

Exemple pratique

Supposons que nous avons un schéma GraphQL avec deux types : **Post** et **Author**. Les Posts ont des champs tels que **title**, **content**, et **author**, qui est une référence à l'Author du post. Les Authors ont des champs tels que **name**, **email**, etc.

Avec une requête GraphQL, nous pouvons récupérer tous les titres des posts et les noms des auteurs associés en une seule requête :

```
query {
  posts {
    title
    author {
```

```
            name
        }
    }
}
```

Cette requête retournera un résultat contenant les titres des posts et les noms des auteurs respectifs.

2.6 Technologies Web

2.6.1 HTML (Hypertext Markup Language)

HTML (**H**yper**T**ext **M**arkup **L**anguage) est un langage de balisage utilisé pour structurer le contenu d'une page web. Il a été inventé par *Tim Berners-Lee* au *CERN* (Organisation Européenne pour la Recherche Nucléaire) en 1990. L'objectif initial était de créer un moyen de partager des informations scientifiques entre les chercheurs.

HTML permet aux développeurs web de décrire la structure logique d'une page en utilisant des balises qui définissent différents éléments tels que les titres, les paragraphes, les images, les liens, les formulaires, etc. Ces balises permettent également de créer des liens hypertextes, ce qui facilite la navigation entre les pages web.

HTML est essentiel pour la création de pages web car il définit la structure et le contenu de la page, qui est ensuite présenté et stylisé à l'aide de **CSS** (**C**ascading **S**tyle **S**heets). Il joue également un rôle important dans l'accessibilité des pages web, en permettant aux développeurs de fournir des alternatives textuelles pour les images et de créer des formulaires accessibles aux personnes handicapées.

Caractéristiques importantes d'HTML

- **Syntaxe basée sur les balises** : HTML utilise des balises pour marquer le début et la fin des éléments sur une page web. Par exemple, <*p*> est utilisé pour définir un paragraphe, <*h1*> pour un titre de niveau 1, etc.
- **Structure hiérarchique** : Les éléments HTML sont organisés de manière hiérarchique, ce qui signifie que certains éléments peuvent être contenus à l'intérieur d'autres éléments. Par exemple, un paragraphe peut contenir du texte et des images.
- **Compatibilité avec les navigateurs** : HTML est interprété par les navigateurs web, ce qui permet aux utilisateurs de visualiser les pages web sur différents appareils et navigateurs.
- **Évolution continue** : HTML a subi plusieurs versions au fil des ans, avec HTML5 étant la version la plus récente. HTML5 introduit de nouvelles fonctionnalités et améliorations pour une meilleure expérience utilisateur et une meilleure prise en charge des médias.

HTML est utilisé dans une variété de cas, notamment pour la création de sites web, de pages de destination, de blogs, d'applications web, de newsletters électroniques et bien plus encore (35).

2.6.2 CSS (Cascading Style Sheets)

CSS (**C**ascading **S**tyle **S**heets) est un langage de style utilisé pour contrôler la présentation et la mise en forme du contenu d'une page web écrite en HTML. Il a été développé par Håkon Wium Lie et Bert Bos au début des années 1990. L'objectif de CSS était de séparer le contenu de la présentation, ce qui permet aux développeurs web de créer des styles cohérents pour leurs pages et de les appliquer à plusieurs pages en même temps.

CSS fonctionne en appliquant des règles de style aux éléments HTML en utilisant des sélecteurs. Les règles de style définissent des propriétés telles que la couleur, la taille de police, la mise en page, etc. Chaque règle est constituée d'un sélecteur et d'un bloc de déclarations de style.

Voici quelques caractéristiques importantes de CSS :

— **Séparation du contenu et de la présentation** : CSS permet aux développeurs web de séparer la structure du contenu (HTML) de sa mise en forme (CSS). Cela facilite la maintenance du code et permet de créer des designs cohérents pour l'ensemble du site.
— **Cascade** : Le terme "cascading" dans CSS signifie que plusieurs règles de style peuvent être appliquées au même élément HTML. Les règles de style avec un sélecteur plus spécifique auront une priorité plus élevée que celles avec un sélecteur moins spécifique.
— **Héritage** : Les propriétés de style définies pour un élément parent peuvent être héritées par ses éléments enfants, ce qui permet d'appliquer des styles globaux à l'ensemble du site.
— **Flexibilité** : CSS offre de nombreuses options pour contrôler la mise en page et l'apparence des éléments, ce qui permet aux développeurs de créer des designs uniques et personnalisés.
— **Media queries** : CSS prend en charge les media queries, qui permettent de définir des styles différents en fonction des caractéristiques de l'appareil utilisé pour afficher la page (par exemple, écran large, tablette, téléphone) (36).

CSS est utilisé dans une variété de cas, notamment pour :

— Styler des sites web et des applications web
— Réaliser des mises en page responsives pour les appareils mobiles et les tablettes
— Créer des animations et des transitions
— Personnaliser l'apparence des formulaires et des boutons
— Imprimer des pages web de manière adaptée pour l'impression

Exemple de code CSS :

```
/* Style pour le corps du document */
body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
}

/* Style pour les titres h1 */
h1 {
  color: #008080;
}

/* Style pour les liens */
a {
  color: #3366cc;
  text-decoration: none;
}

/* Style pour les boutons */
.button {
  background-color: #ff6600;
  color: #ffffff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

/* Style pour les images */
img {
  max-width: 100%;
  height: auto;
}
```

2.6.3 JavaScript

JavaScript est un langage de programmation de haut niveau, interprété, orienté objet et multiplateforme. Il a été créé par *Brendan Eich* en 1995 alors qu'il travaillait pour *Netscape Communications Corporation*. Le langage a été développé pour être utilisé dans les navigateurs web afin de rendre les pages web plus interactives et dynamiques. Depuis sa création, JavaScript est devenu l'un des langages de programmation les plus populaires et les plus largement utilisés dans le développement web (37).

Caractéristiques de JavaScript :

— **Interprété** : JavaScript est un langage interprété, ce qui signifie qu'il est exécuté directement par le navigateur sans nécessiter de compilation préalable.

— **Orienté objet** : JavaScript est un langage orienté objet, ce qui signifie qu'il permet de définir et de manipuler des objets. Il supporte l'héritage, le polymorphisme et l'encapsulation.
— **Dynamique** : JavaScript est un langage dynamique, ce qui signifie que les types des variables sont déterminés à l'exécution plutôt qu'à la compilation.
— **Multiplateforme** : JavaScript est pris en charge par tous les principaux navigateurs web, ce qui le rend multiplateforme et indépendant du système d'exploitation.

Utilisations de JavaScript :
— **Développement web** : JavaScript est principalement utilisé pour rendre les pages web interactives et dynamiques en ajoutant des fonctionnalités telles que les animations, les formulaires interactifs, les menus déroulants, etc.
— **Développement d'applications web** : JavaScript est utilisé pour développer des applications web côté client, également connues sous le nom d'applications web front-end. Il est souvent utilisé en combinaison avec HTML et CSS.
— **Développement d'applications mobiles** : JavaScript est utilisé pour développer des applications mobiles en utilisant des frameworks tels que React Native et Ionic.
— **Développement de jeux** : JavaScript est utilisé pour développer des jeux en utilisant des bibliothèques et des moteurs de jeu tels que Phaser et Three.js.

Exemple de code JavaScript :

```
// Declaration d'une fonction en JavaScript
function addition(a, b) {
  return a + b;
}

// Appel de la fonction et affichage du r sultat
let resultat = addition(5, 3);
console.log(resultat); // Affiche 8
```

2.6.4 JSON (JavaScript Object Notation)

JSON (**J**ava**S**cript **O**bject **N**otation) est un langage de notation de données léger, largement utilisé pour l'échange de données entre applications. Il a été inventé par *Douglas Crockford* dans les années 2000 et est dérivé de la syntaxe des objets JavaScript.

JSON est basé sur deux structures de données :
— **Objets** : représentés par des paires clé-valeur, où les clés sont des chaînes de caractères et les valeurs peuvent être des nombres, des chaînes, des tableaux, d'autres objets ou des valeurs booléennes (true ou false) et null.

Exemple d'objet JSON :

```
{
  "nom": "John Doe",
  "age": 30,
  "ville": "Paris"
}
```

— **Tableaux** : représentés par une liste ordonnée de valeurs, où chaque valeur peut être de n'importe quel type de données JSON.
Exemple de tableau JSON :

```
["rouge", "vert", "bleu"]
```

JSON est largement utilisé dans le développement Web pour transmettre des données entre les serveurs et les clients. Il est souvent utilisé dans les API REST pour échanger des données structurées entre le serveur et l'application cliente.

Caractéristiques de JSON

— **Léger** : JSON est un format de données léger, facile à lire et à écrire pour les humains et les machines.
— **Syntaxe simple** : JSON utilise une syntaxe simple et intuitive, basée sur les objets et les tableaux.
— **Indépendant du langage** : JSON est indépendant du langage de programmation, ce qui signifie qu'il peut être utilisé avec une variété de langages de programmation.

JSON est devenu un format de données standard dans le développement Web moderne en raison de sa simplicité, de sa lisibilité et de son extensibilité. Il est également utilisé dans d'autres domaines, tels que le stockage de configurations et de métadonnées (38) (39).

2.6.5 XML (Extensible Markup Language)

XML (e**X**tensible **M**arkup **L**anguage) est un langage de balisage conçu pour stocker et transporter des données de manière structurée et lisible par l'homme. Il a été créé dans les années 1990 par un groupe de travail du *W3C* (*World Wide Web Consortium*), comprenant notamment *Jon Bosak*, *Tim Bray* et *Jean Paoli*.

XML a été développé dans le but de fournir un format de données indépendant des plateformes et des applications, permettant un échange de données plus flexible et interopérable entre différentes technologies (40) (41).

Principales caractéristiques de XML

— **Balisage** : Les données XML sont structurées à l'aide de balises qui définissent les éléments et les attributs des données. Les balises sont délimitées par des chevrons (< >).

— **Lisibilité** : XML est conçu pour être lisible par l'homme, ce qui facilite le débogage et la compréhension des données.
— **Extensibilité** : XML permet aux utilisateurs de définir leurs propres balises personnalisées, ce qui permet une grande flexibilité dans la représentation des données.
— **Hiérarchie** : Les données XML sont organisées en une structure hiérarchique d'éléments et de sous-éléments, ce qui facilite la représentation de données complexes.

Utilisations courantes de XML

— **Stockage de données** : XML est largement utilisé pour stocker des données structurées dans des fichiers et des bases de données.
— **Échange de données** : XML est souvent utilisé comme format d'échange de données entre applications et systèmes hétérogènes.
— **Web services** : XML est utilisé dans les services web pour transmettre des données entre les clients et les serveurs.
— **Configuration** : XML est utilisé pour stocker des configurations et des paramètres de logiciels.

Exemple de données XML

```
<bookstore>
    <book>
      <title>Introduction to XML </title>
      <author>John Doe</author>
      <year>2023</year>
    </book>
    <book>
      <title>Advanced XML Techniques</title>
      <author>Jane Smith</author>
      <year>2022</year>
    </book>
</bookstore>
```

2.7 L'architecture MVC

L'architecture MVC (**M**odèle-**V**ue-**C**ontrôleur) est un modèle architectural utilisé dans le développement de logiciels pour organiser et structurer le code de manière à séparer les préoccupations et à faciliter la maintenance et l'extensibilité du code. Il est couramment utilisé dans le développement web et d'autres applications à interface graphique (42).

2.7.1 Composant de l'architecture MVC

L'architecture MVC comprend 3 composants comme suite :

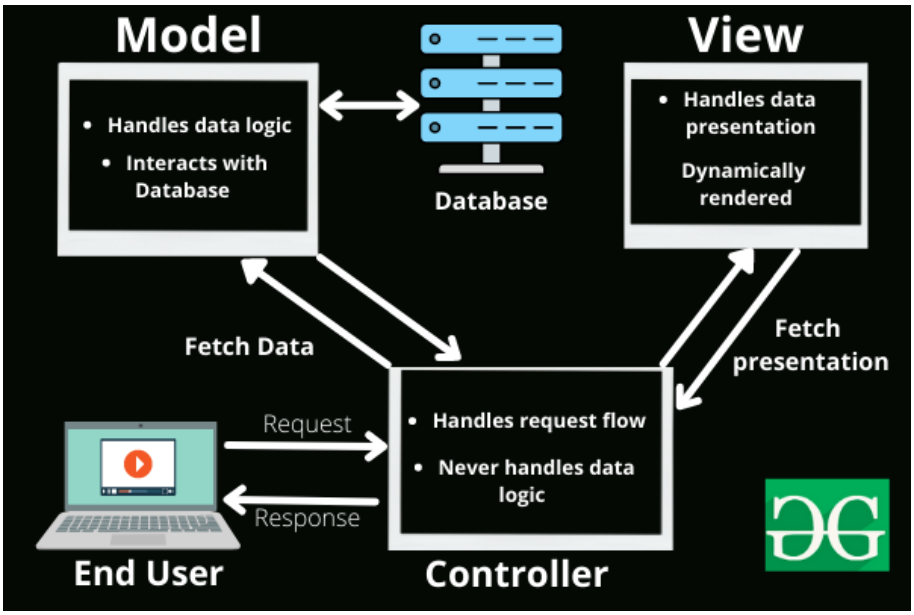

FIGURE 2.6 – Architecture MVC

Modèle (Model)

Le Modèle représente les données et la logique métier de l'application. Il encapsule les données et fournit des méthodes pour interagir avec celles-ci. Dans une application web, le Modèle peut récupérer des données depuis une base de données, effectuer des calculs, et répondre aux requêtes du Contrôleur (43). Il est important de noter que le Modèle est indépendant de l'interface utilisateur ou de la façon dont les données sont présentées.

Vue (View)

La Vue est responsable de la présentation des données à l'utilisateur. C'est le composant d'interface utilisateur qui affiche les informations provenant du Modèle et recueille les saisies de l'utilisateur. Dans le développement web, une Vue pourrait être une page HTML ou un modèle qui est rendu dans le navigateur (43). La Vue ne doit pas contenir de logique métier. Elle doit plutôt recevoir des données du Contrôleur ou directement du Modèle et les présenter à l'utilisateur.

Contrôleur (Controller)

Le Contrôleur agit comme un intermédiaire entre le Modèle et la Vue. Il reçoit l'entrée de l'utilisateur de la Vue, la traite (ce qui peut impliquer d'interagir avec le Modèle) et met à jour la Vue en conséquence. Dans une application web, le Contrôleur reçoit les demandes du client, les traite, et décide quels méthodes du Modèle appeler et quelle Vue afficher en réponse (43).

2.7.2 Fonctionnement du framework MVC avec un exemple

Supposons qu'un utilisateur envoie une demande à un serveur afin d'obtenir une liste des étudiants inscrits dans une classe donnée. Le serveur transmettrait cette requête au contrôleur (Controller) spécifique responsable de la gestion des étudiants. Ce contrôleur (Controller) solliciterait ensuite le modèle (Model) en charge des étudiants pour qu'il fournisse une liste exhaustive des étudiants dans cette classe.

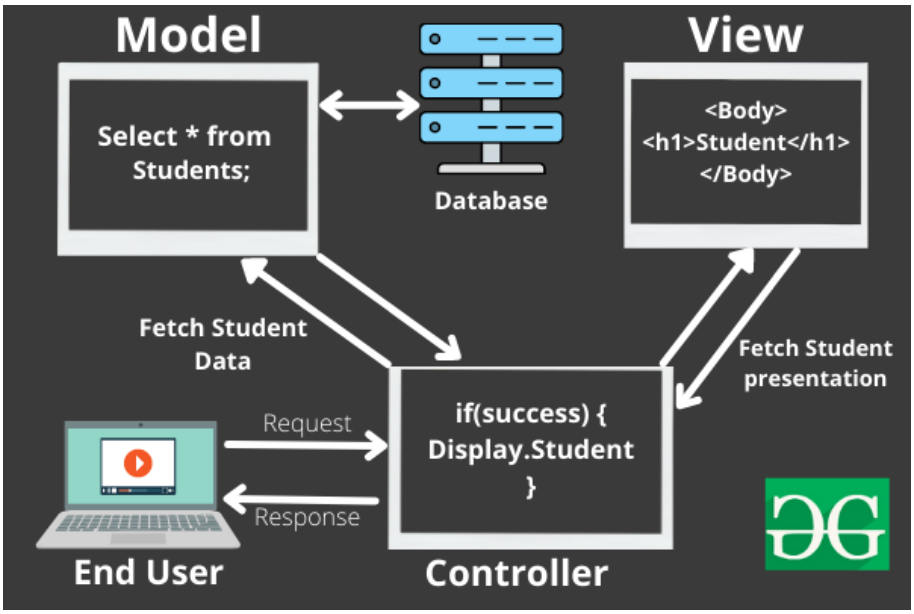

FIGURE 2.7 – flux de données dans l'architecture MVC

Le modèle (Model) interroge la base de données afin de récupérer la liste complète des étudiants, puis transmet cette liste au contrôleur (Controller). Si la requête du modèle est réussie, le contrôleur sollicite la vue associée aux étudiants pour générer une représentation de cette liste. Cette vue (View) prend les informations sur les étudiants du contrôleur et les transforme en un format HTML exploitable par le navigateur.

Le contrôleur prendrait alors cette présentation et la renverrait à l'utilisateur. Mettant ainsi fin à la demande. Si auparavant le modèle renvoyait une erreur, le contrôleur gérerait cette erreur en demandant à la vue qui gère les erreurs de restituer une présentation pour cette erreur particulière. Cette présentation d'erreur serait alors renvoyée à l'utilisateur au lieu de la présentation de la liste des étudiants.

Comme nous pouvons le voir dans l'exemple ci-dessus, le modèle gère toutes les données. La vue gère toutes les présentations et le contrôleur indique simplement au modèle et à la vue ce qu'il faut faire. Il s'agit de l'architecture de base et du fonctionnement du framework MVC.

2.7.3 Avantages du MVC

— **Séparation des Préoccupations** Chaque composant (Modèle, Vue, Contrôleur) a un rôle distinct, ce qui rend le code plus facile à comprendre, à maintenir et à étendre.
— **Réutilisabilité du Code** Grâce à la séparation, les composants peuvent être réutilisés dans différentes parties de l'application ou même dans d'autres applications.
— **Testabilité** Chaque composant peut être testé de manière indépendante. Cela facilite les tests unitaires et les tests automatisés.
— **Scalabilité** Il est plus facile de mettre à l'échelle différentes parties de l'application indépendamment. Par exemple, vous pourriez avoir besoin de plus de serveurs pour gérer un trafic accru du côté du Contrôleur, ou vous pourriez optimiser le Modèle pour gérer de grandes quantités de données.
— **Développement Parallèle** Différentes équipes ou développeurs peuvent travailler sur différents composants en même temps sans interférer avec le travail des autres.

2.8 Frameworks MVC populaires

Les frameworks Model-View-Controller (MVC) sont les piliers du développement web moderne. Ils fournissent une structure essentielle pour concevoir des applications web robustes et flexibles. Voici une liste des frameworks MVC les plus populaires et largement utilisés dans le Web 2.0.

2.8.1 Node Js

Node.js est un environnement d'exécution JavaScript côté serveur, construit sur le moteur JavaScript V8 de Google Chrome. Il permet d'exécuter du code JavaScript en dehors du navigateur, ce qui signifie qu'il peut être utilisé pour créer des applications serveur, des API, des outils en ligne de commande, et bien plus encore.

2.8.2 Express Js

Express.js est un framework web minimaliste pour Node.js. Il est conçu pour simplifier le processus de création d'applications web et d'API en fournissant des fonctionnalités de base tout en laissant aux développeurs la flexibilité nécessaire pour étendre et personnaliser leur application.

Caractéristiques d'Express.js

Routage : Express permet de définir des routes qui correspondent à des URL spécifiques. Chaque route peut être associée à une fonction de rappel (callback) qui gère la logique de traitement de la requête.

Middleware : Les middleware sont des fonctions qui peuvent être utilisées pour traiter les requêtes HTTP avant qu'elles n'atteignent la route finale. Ils sont souvent utilisés pour

effectuer des opérations communes comme l'analyse des corps de requête, l'ajout d'entêtes, l'authentification, etc.

Gestion des requêtes et réponses : Express simplifie la gestion des requêtes et réponses HTTP. Il fournit des méthodes telles que res.send(), res.json(), res.render() pour envoyer des réponses au client.

Templates de vue (View Engines) : Bien qu'Express soit minimaliste, il ne contient pas de moteur de modèles intégré. Cependant, il prend en charge l'utilisation de divers moteurs de modèles tels que EJS, Pug, Handlebars, etc., pour faciliter la création de vues dynamiques.

Gestion des erreurs : Express fournit des mécanismes pour gérer les erreurs. Il est possible d'ajouter un middleware d'erreur qui intercepte les erreurs et les traite de manière appropriée.

Gestion des sessions et des cookies : Bien qu'Express ne fournisse pas de gestion de sessions ou de cookies intégrée, il existe des middleware comme express-session et cookie-parser qui permettent de gérer ces fonctionnalités.

API RESTful : Express est souvent utilisé pour créer des API RESTful, en utilisant les méthodes HTTP (GET, POST, PUT, DELETE) pour gérer les ressources.

Simplicité et flexibilité : Express est conçu pour être minimaliste et ne fournit que des fonctionnalités de base. Cela permet aux développeurs de choisir et d'ajouter les fonctionnalités dont ils ont besoin via des middleware ou des bibliothèques tierces.

Grande communauté et écosystème : Comme Node.js, Express a une communauté active et il existe de nombreuses bibliothèques tierces disponibles pour étendre ses fonctionnalités.

Bien adapté à la création d'API et de microservices : En raison de sa simplicité et de sa flexibilité, Express est souvent utilisé pour créer des API et des microservices, en permettant aux développeurs de rapidement construire des services web performants.

2.8.3 Ruby on Rails

Ruby on Rails, souvent appelé simplement "Rails", est un framework open-source de développement web écrit en Ruby, un langage de programmation interprété. Ce framework a été créé par David Heinemeier Hansson et a été rendu public en 2004 (44).

Ruby on Rails suit le modèle de conception MVC (Modèle-Vue-Contrôleur), qui divise une application web en trois composants principaux : le modèle (qui gère les données et la logique métier), la vue (qui gère l'interface utilisateur et l'affichage) et le contrôleur

(qui gère les interactions entre le modèle et la vue) (44).

Les caractéristiques clés de Ruby on Rails comprennent la convention sur la configuration, ce qui signifie que Rails offre des conventions préétablies pour simplifier le développement, une structure de répertoires organisée, une intégration étroite avec la base de données, un système de routage flexible pour gérer les URL, la gestion automatisée de nombreux aspects de la sécurité, et une grande variété de bibliothèques et de modules prêts à l'emploi, appelés "gems", qui facilitent le développement (45).

Ruby on Rails est célèbre pour sa simplicité et sa productivité, ce qui en fait un choix populaire parmi les développeurs pour créer des applications web rapidement et efficacement. Il est utilisé dans de nombreux projets web et a contribué à populariser le langage Ruby.

2.8.4 Laravel

Laravel est un framework de développement web open-source écrit en PHP. Il offre une structure et un ensemble de fonctionnalités puissantes pour simplifier et accélérer le processus de création d'applications web. Laravel suit le modèle de conception MVC (Modèle-Vue-Contrôleur) qui permet de séparer la logique métier, la présentation et le traitement des requêtes (46).

Ce framework offre de nombreuses fonctionnalités prêtes à l'emploi telles que la gestion des bases de données, l'authentification utilisateur, le routage, la gestion des sessions, et bien d'autres. Il intègre également une syntaxe élégante et expressive, ainsi qu'une large gamme de bibliothèques et d'outils qui simplifient le développement (46).

Laravel a gagné en popularité grâce à sa facilité d'utilisation, sa documentation complète, et sa communauté active qui contribue à son évolution et fournit un support important aux développeurs.

2.8.5 Django

Django est un framework web open source écrit en Python. Il fournit une infrastructure solide et complète pour le développement rapide d'applications web sécurisées et évolutives. Créé par Adrian Holovaty et Simon Willison, Django a été initialement publié en 2005 et est depuis devenu l'un des frameworks web les plus populaires et largement utilisés (47).

La philosophie de Django est de suivre le principe du "Don't Repeat Yourself" (DRY) et de privilégier la simplicité et la rapidité de développement. Pour ce faire, il intègre un ensemble de composants prêts à l'emploi, tels qu'un système de gestion de bases de données, un ORM (Object-Relational Mapping), une interface d'administration, un système d'authentification, un système de formulaires, un gestionnaire de URL, et bien d'autres. Ces composants permettent aux développeurs de se concentrer sur la logique métier de

leurs applications plutôt que sur la mise en place d'infrastructures de base (48).

Django suit le modèle de conception MVC (Modèle-Vue-Contrôleur) mais il est souvent appelé "Modèle-Vue-Template" (MVT) dans la documentation de Django, car il utilise des templates pour la vue. Il encourage une approche basée sur la convention plutôt que la configuration, ce qui signifie qu'il offre des structures prédéfinies pour faciliter le développement, tout en laissant la possibilité d'ajuster ces conventions selon les besoins spécifiques du projet (48).

En résumé, Django est un framework web puissant, flexible et modulaire qui simplifie le processus de développement d'applications web complexes en offrant une abondance de fonctionnalités prêtes à l'emploi et en suivant les meilleures pratiques de développement. Il est particulièrement apprécié pour sa capacité à favoriser la productivité des développeurs tout en garantissant la sécurité et la performance des applications, ce qui nous a encouragés et motivés à choisir Django pour le développement de notre plateforme.

2.9 Modes de communication

Introduction

Les modes de communication sur le web font référence aux différentes méthodes utilisées pour échanger des données entre les clients et les serveurs. Les principaux modes de communication sont le mode **synchrone**, où la communication se fait en temps réel et bloque l'exécution du code jusqu'à ce qu'une réponse soit reçue, et le mode **asynchrone**, où la communication se fait de manière non bloquante et permet au code de continuer son exécution pendant que les données sont en cours d'échange. Le mode asynchrone est souvent privilégié car il permet une meilleure performance et une expérience utilisateur plus fluide.

2.9.1 Communication synchrone

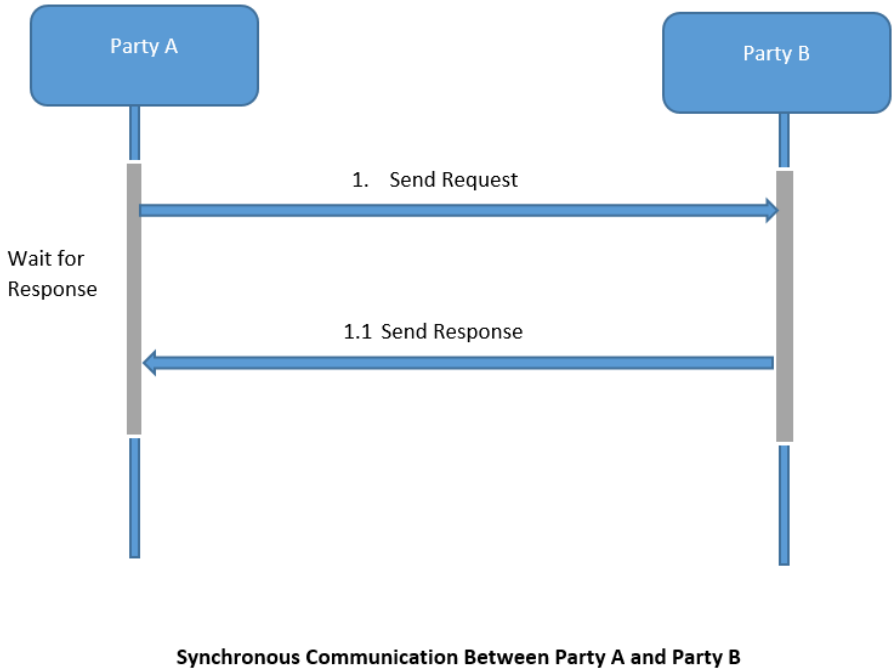

FIGURE 2.8 – Mode de communication synchrone

Le mode de communication **synchrone** est un paradigme dans lequel l'échange d'informations entre les parties impliquées se produit en temps réel et de manière coordonnée. Ce mode a été inventé dans le contexte des communications informatiques pour faciliter la transmission de données entre des systèmes interconnectés.

Dans le mode synchrone, l'émetteur envoie un message et attend une réponse immédiate de la part du destinataire. Le processus se déroule dans un ordre strict, où chaque action est effectuée de manière séquentielle. Cette synchronisation permet un contrôle plus précis du flux de données et une communication plus fiable.

Les cas d'utilisation courants du mode synchrone incluent les appels API, les requêtes de bases de données et les transactions en ligne. Il est particulièrement utile lorsque des réponses instantanées sont nécessaires pour prendre des décisions en temps réel.

Les principales caractéristiques du mode synchrone

— **Temps réel** : Les échanges d'informations se font immédiatement, permettant une communication en temps réel entre les parties.
— **Coordination** : Les parties impliquées sont synchronisées pour garantir que les actions se déroulent de manière ordonnée.

— **Fiabilité** : En raison de la synchronisation stricte, les chances d'erreurs ou de pertes de données sont réduites.

Cependant, le mode synchrone présente également des inconvénients potentiels, tels que la dépendance à une réponse immédiate et le risque de blocage en cas de problèmes de communication (49).

2.9.2 Communication asynchrone

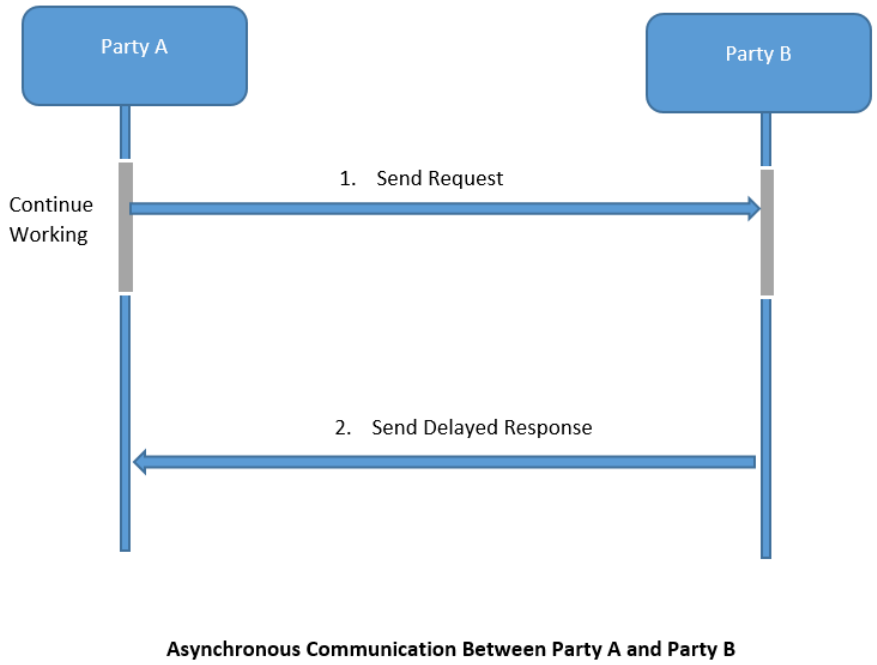

FIGURE 2.9 – Mode de communication asynchrone

Le mode de communication **asynchrone** est un paradigme dans lequel l'échange d'informations entre les parties impliquées se produit sans qu'elles aient besoin de se synchroniser en temps réel. Ce mode a été développé pour résoudre certains des problèmes rencontrés dans les communications synchrones et pour permettre une plus grande flexibilité dans l'échange de données entre les systèmes interconnectés.

Dans le mode asynchrone, l'émetteur envoie un message sans attendre de réponse immédiate de la part du destinataire. Au lieu de cela, il continue ses tâches sans interruption. Le destinataire recevra le message et traitera les données de manière indépendante, sans qu'il soit nécessaire que l'émetteur attende une réponse.

Ce mode de communication est particulièrement utile lorsque des délais variables sont acceptables ou lorsque des tâches peuvent être effectuées de manière asynchrone pour améliorer les performances globales. Il est largement utilisé dans les systèmes distribués,

les applications Web, les services d'API, et les communications réseau.

Les caractéristiques clés du mode asynchrone

- **Flexibilité** : Les parties impliquées peuvent effectuer des tâches de manière indépendante, ce qui permet une plus grande flexibilité dans le flux de données.
- **Performances** : L'utilisation du mode asynchrone peut améliorer les performances globales en évitant les attentes inutiles et en optimisant l'utilisation des ressources.
- **Gestion des erreurs** : Le mode asynchrone permet une meilleure gestion des erreurs et des problèmes de communication en utilisant des mécanismes de rappel ou de promesses.

Cependant, le mode asynchrone peut rendre la gestion du code plus complexe en raison de la nécessité de gérer les rappels ou les promesses pour gérer les réponses.

2.10 Tendances futures

Avec l'évolution rapide du **Web 2.0** et ses technologies révolutionnaires, nous assistons à l'émergence de nouvelles technologies qui redéfiniront l'avenir du web. Le Web 2.0 a permis aux utilisateurs de participer activement en créant et partageant du contenu, tandis que les réseaux sociaux ont transformé notre manière d'interagir en ligne.

Cependant, le futur du web va bien au-delà du Web 2.0. Nous sommes à l'aube d'une nouvelle ère de technologies émergentes qui promettent de bouleverser notre expérience numérique. Parmi elles, le **Web 3.0** prend forme, mettant l'accent sur la décentralisation et l'interopérabilité. Les technologies de la *blockchain* et des *contrats intelligents* permettent une collaboration transparente et sécurisée, ouvrant de nouvelles possibilités dans les domaines des *finances décentralisées (DeFi)*, de l'identité numérique, de la gouvernance et bien plus encore.

D'autre part, **Web Assembly (Wasm)** se profile comme une technologie puissante, permettant l'exécution de code à haute performance dans les navigateurs. Cette virtual machine bas-niveau permet aux développeurs d'écrire des applications web plus rapides et plus efficaces, étendant ainsi les capacités du web moderne.

Parallèlement, l'intelligence artificielle (IA) et l'apprentissage automatique (ML) continuent de progresser, apportant des solutions avancées dans divers domaines, y compris la recommandation personnalisée, la reconnaissance vocale et d'image, l'automatisation des processus, et bien plus encore.

L'Internet des Objets (IoT) se développe également, connectant un nombre croissant d'appareils et générant d'énormes quantités de données. Cette interconnexion d'objets offre un potentiel énorme pour de nouveaux services et applications dans des secteurs

tels que la domotique, la santé, l'industrie et la logistique.

Ces avancées technologiques, tout en promettant des opportunités excitantes, soulèvent également des défis en matière de sécurité, de protection des données personnelles et de gestion des risques.

2.11 Conclusion

Dans ce chapitre, nous avons présenté une vue globale sur le Web 2.0, nous avons commencé par l'historique et l'architecture générale du web. Ensuite, nous avons décrit quelques architectures des applications web et le mode de communication dans ces derniers. Enfin, nous avons brièvement abordé le future du Web.

Dans le prochain chapitre nous allons entamer l'étape de l'analyse et Conception de notre plateforme.

Chapitre 3

Conception

3.1 Introduction

Ce chapitre présentera la processus du conception et mettra en évidence les objectifs de notre plate-forme, ses acteurs ainsi que leurs besoins, qui sont une nécessité incontestable pour se lancer dans la phase de réalisation. L'assurance d'une telle tâche repose principalement sur une représentation graphique de ces besoins et de tous les aspects du système, réalisée à l'aide de diagrammes UML.

3.2 Objectif de notre plateforme

L'objectif principal de notre plate-forme est de faciliter la création, la personnalisation et la vente l'achat de produits imprimés et personnalisés tels que vêtements, accessoires et articles divers.

Cette plateforme permet aux créateurs et aux artistes de concevoir leurs propres designs uniques, puis de les appliquer sur une variété de produits disponibles sur la plateforme. Les clients peuvent ensuite acheter ces produits avec les designs de leur choix, ce qui élimine le besoin de stocker des inventaires à l'avance, la production, livraison, marketing et le service client qui seraient l'objectif et la responsabilité de notre plate-forme.

3.3 Identification des acteurs

Un acteur représente un rôle joué par une entité externe qui interagit directement avec le système étudié. Dans notre cas, nous avons quatre acteurs :

- **Utilisateur :** c'est une personne non authentifié (qui accède au site pour la première fois).
- **Client :** c'est une personne qui veut effectuée une commande ou personnaliser son propre produit.
- **Vendeur :** c'est une personne qui veut lancer sa propre boutique (store) et uploader leur propres designs.
- **Admin :** c'est la personne qui est chargée de la gestion générale de la plateforme (ajout, suppression, modification, support).

Remarque : le vendeur peut être un client en même temps et il peut commander des produits personnalisées.

3.4 Spécification des besoins

Notre plate-forme proposée doit répondre aux besoins fonctionnelles nécessaires qui seront exécutés par le système, ainsi qu'aux besoins non fonctionnelles visant à améliorer la qualité logicielle de celui-ci.

3.4.1 Besoins fonctionnels

Notre application doit satisfaire les besoins fonctionnels suivants :

— Authentification et gestion des rôles.
— Création des stores.
— Création et upload des designs.
— Choisir un produit pour l'imprimer.
— Faire un filtrage des produit selon la catégorie, popularité et le prix ..etc
— Consulter les produits et les designs et effectuer des commandes.

3.4.2 Besoins non fonctionnels

Ce sont les besoins permettant l'amélioration de la qualité des services offerts par une application. Dans notre travail nous avons considéré la liste des besoins non fonctionnel présentés ci-dessous :

— **Sécurité :** l'application doit être extrêmement sécurisée, et les données ne doivent pas être accessibles facilement à tous, mais plutôt au moyen d'un identifiant et d'un mot de passe attribués à l'utilisateur.
— **Convivialité :** l'application doit être facile à utiliser, et elle doit présenter un enchaînement logique entre les interfaces.
— **L'extensibilité :** l'application devra être extensible c'est-à-dire qu'il serait possible d'ajouter ou de modifier des fonctionnalités.
— **La performance :** l'application devra être performante c'est-à-dire que le système doit réagir dans un délai minimal quel que soit l'action de l'utilisateur.

3.5 Les diagrammes UML

Les diagrammes UML (Unified Modeling Language) sont des outils visuels utilisés pour modéliser, concevoir et représenter graphiquement les systèmes logiciels et les processus. Ils fournissent une manière standardisée de visualiser différentes facettes d'un système, de la structure à la dynamique, en passant par les interactions entre les composants(50) (51), qui nous aidé et assuré le bonne réalisation de notre plateforme et de tous ses aspects.

3.5.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est une représentation graphique qui montre comment un système interagit avec les acteurs externes(51). Le diagramme de cas d'utilisation est un outil utile pour définir et comprendre les besoins et les interactions des utilisateurs avec un système logiciel ou un processus.

Diagramme de cas d'utilisation associé à l'Admin

Ce diagramme illustre les cas d'utilisation associé à l'Admin.

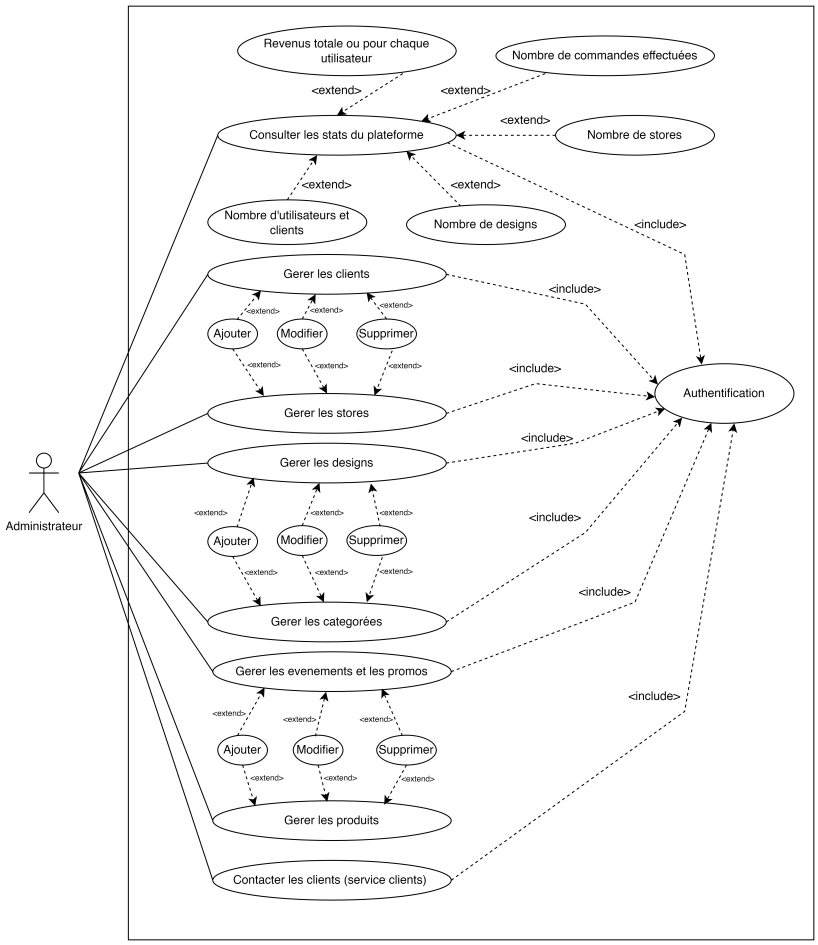

FIGURE 3.1 – Diagramme de cas d'utilisation associé à l'Admin

Diagramme de cas d'utilisation associé à l'utilisateur

Ce diagramme illustre les cas d'utilisation associé à l'utilisateur.

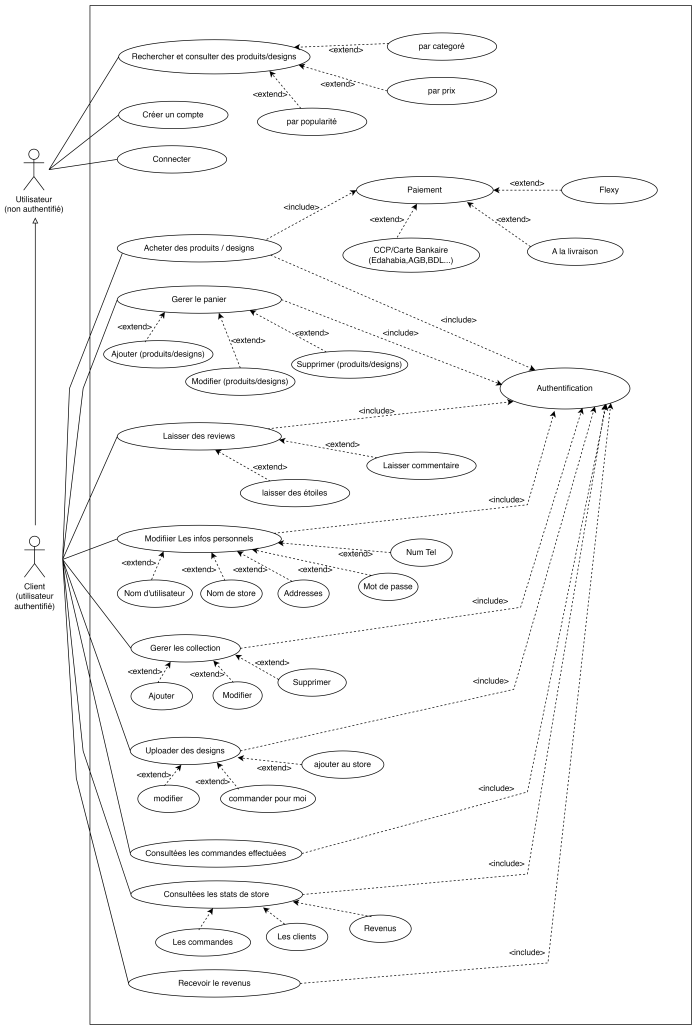

FIGURE 3.2 – Diagramme de cas d'utilisation associé à l'utilisateur

3.5.2 Diagramme de séquence

Un diagramme de séquence est un type de diagramme de modélisation utilisé pour représenter visuellement l'interaction et le flux de contrôle entre différentes entités (comme des objets, des acteurs ou des composants) au sein d'un système logiciel ou d'un processus(51) (52). Il se concentre principalement sur la chronologie des messages échangés entre ces entités pour décrire comment elles collaborent pour accomplir une tâche ou un scénario spécifique.

Diagramme de séquence de la tache "Connexion"

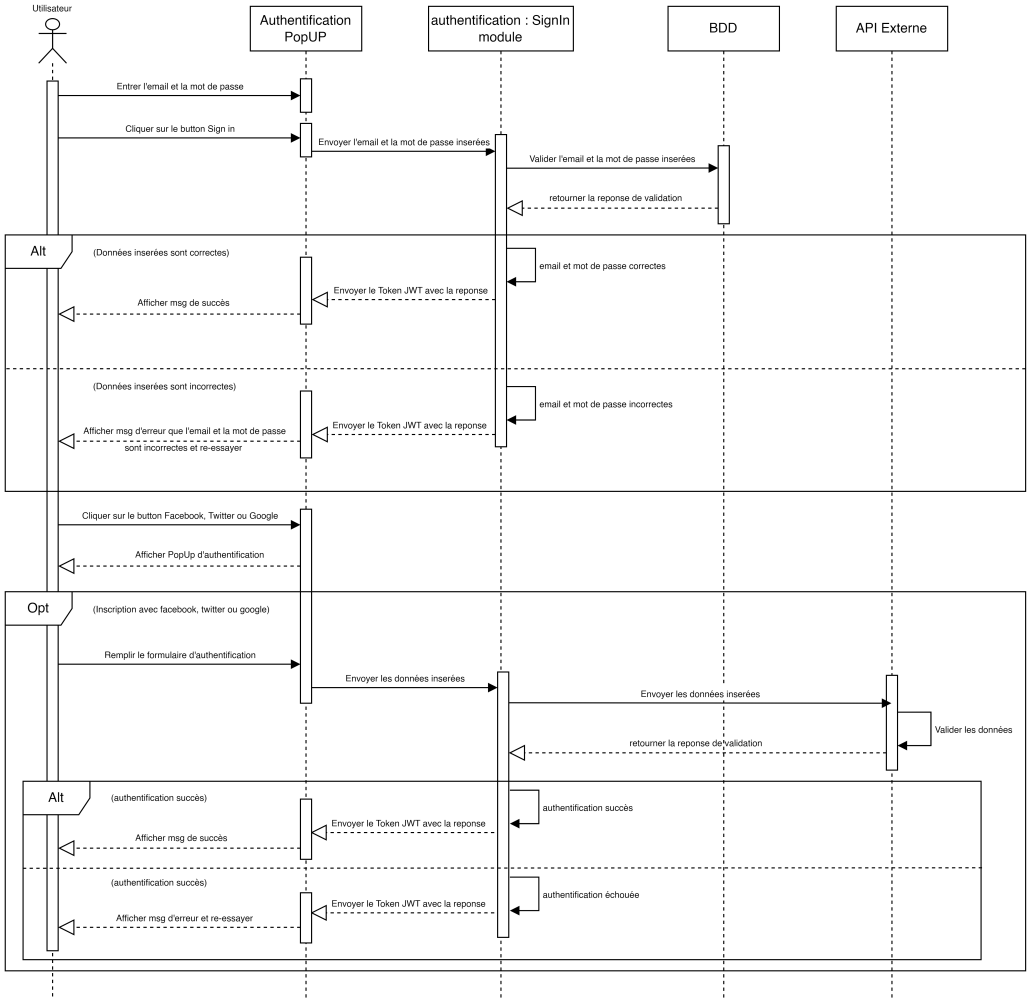

FIGURE 3.3 – Diagramme de séquence associé à la tâche Connexion

Diagramme séquence de la tache "Inscription"

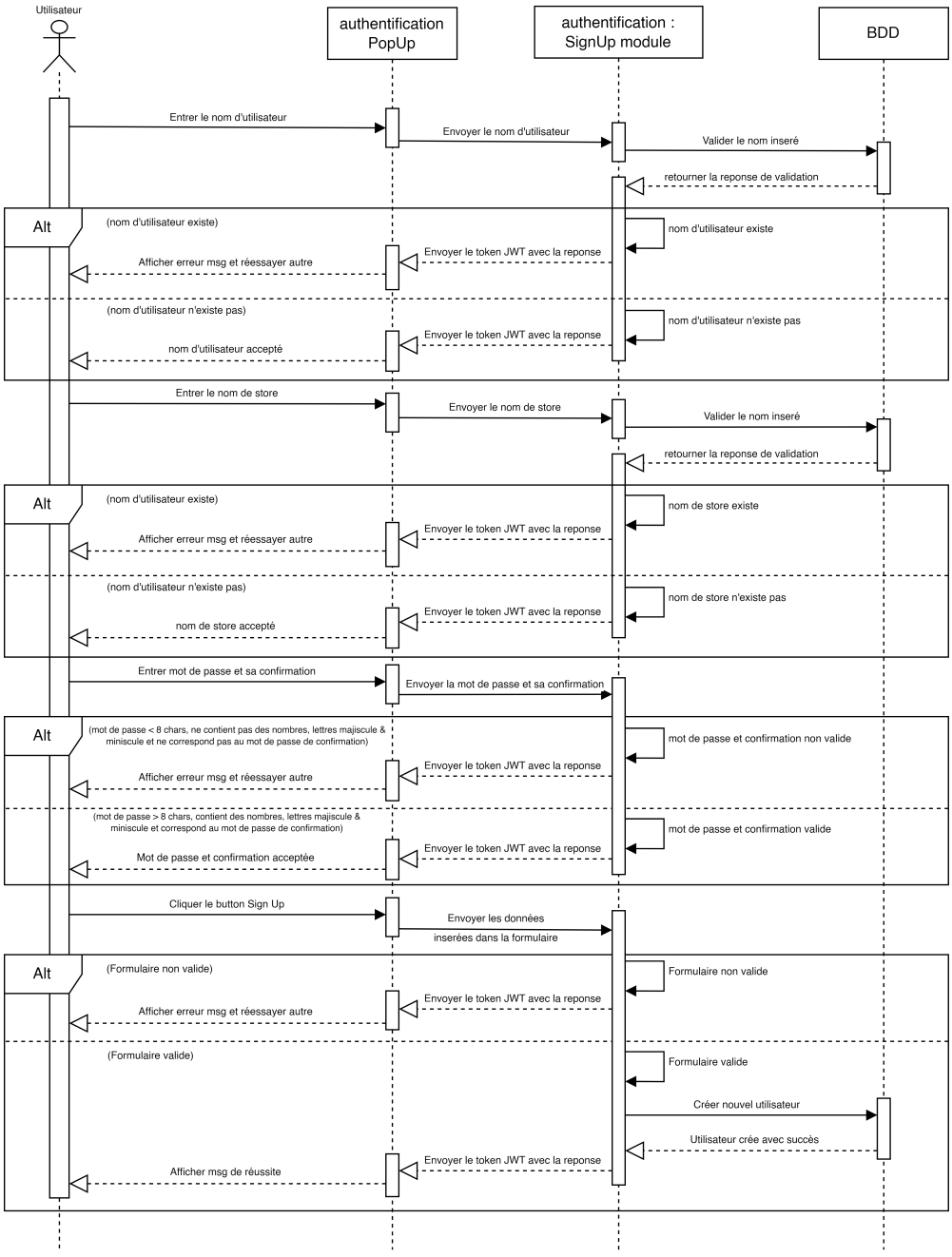

FIGURE 3.4 – Diagramme de séquence associé à la tâche Inscription

Diagramme de séquence de la tache "Effectuer une commande"

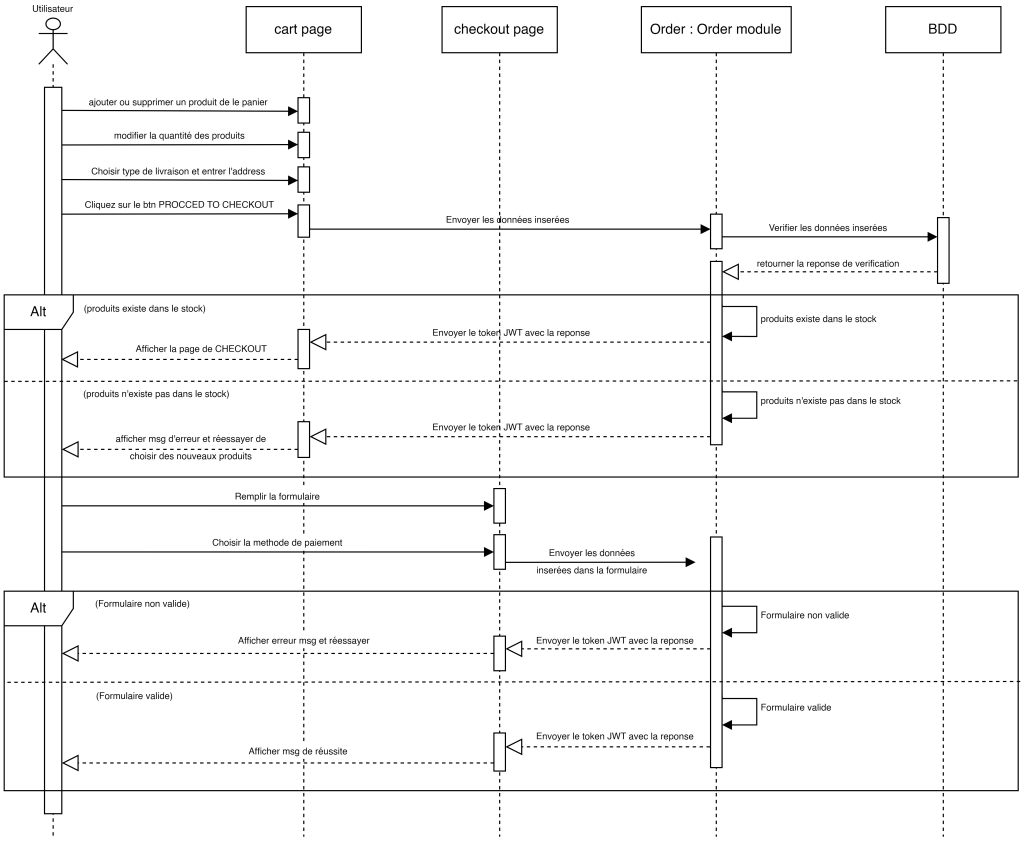

FIGURE 3.5 – Diagramme de séquence associé à la tâche Effectuer une commande

Diagramme de séquence de la tache "Créer / Uploader un design"

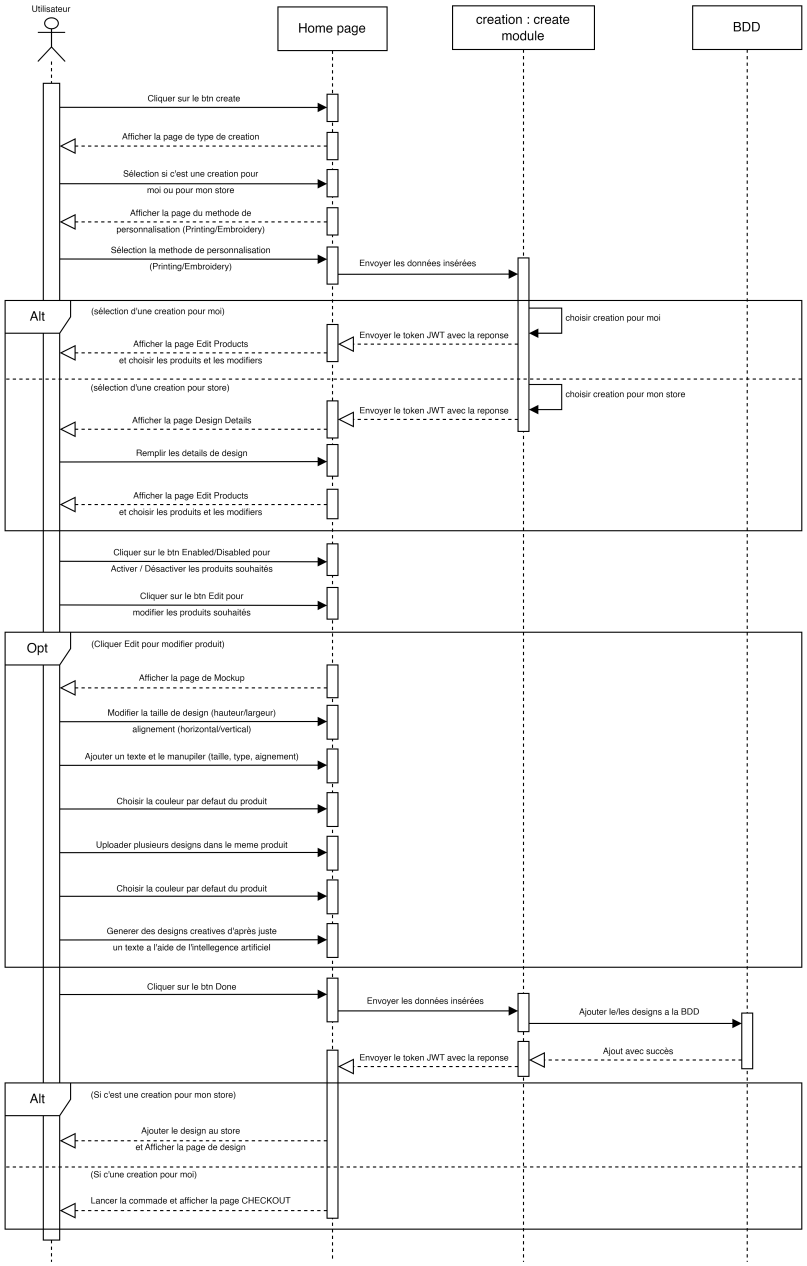

FIGURE 3.6 – Diagramme de séquence associé à la tâche Créer/Uploader un design

3.5.3 Diagramme de Class

Utilisé pour représenter graphiquement la structure statique d'un système logiciel. Il illustre les classes du système, les relations entre ces classes, ainsi que les attributs et les méthodes associés à chaque classe.

Un diagramme de classe fournit une vue statique d'un système, montrant les classes, leurs attributs, méthodes et relations. Cela aide les développeurs à comprendre la structure du système et à concevoir des solutions logicielles efficaces.

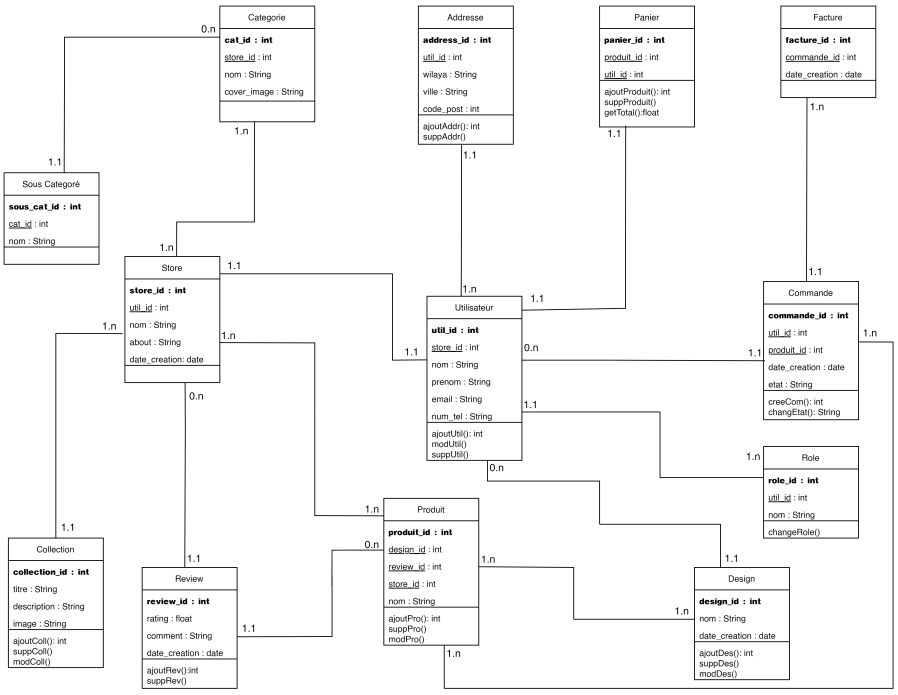

FIGURE 3.7 – Diagramme de class

3.5.4 Diagramme MCD (Model Conceptuel de Données)

MCD est l'acronyme de "Modèle Conceptuel de Données". Il s'agit d'un type de modèle de données utilisé en ingénierie des systèmes d'information pour représenter les concepts, les entités et les relations qui sont pertinents pour une application ou un domaine spécifique (51) (53).

Le MCD est généralement représenté sous forme de diagrammes qui illustrent visuellement les entités, les attributs, les relations et les cardinalités. Il s'agit d'une étape importante dans la conception de bases de données et de systèmes d'information, car il aide à clarifier les besoins en termes de données et à créer une base solide pour la création ultérieure de schémas de bases de données plus détaillés (51).

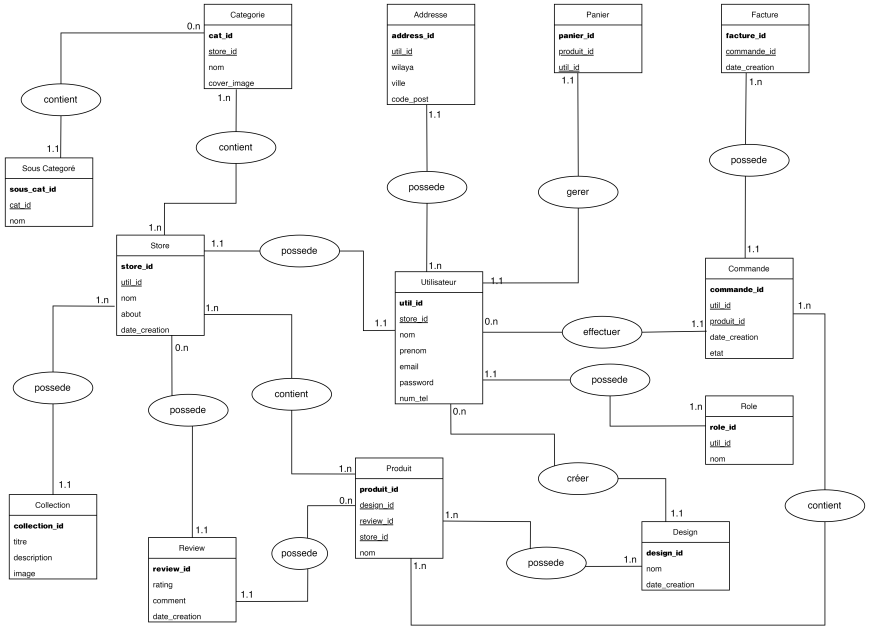

FIGURE 3.8 – Diagramme MCD

3.6 Conclusion

Dans ce chapitre, nous avons abordé l'objectif de notre plateforme, ensuite nous avons identifié nos utilisateurs et déterminer leur besoins et expliqué les interactions qui se produisent entre eux et le système à travers des diagrammes UML montrant le fonctionnement du plateforme. Nous passons maintenant au chapitre suivant qui sera consacré à l'implémentation de la plateforme.

Chapitre 4

L’implémentation

4.1 Introduction

La mise en œuvre d'un projet informatique nécessite la sélection des technologies adéquates à son implémentation. Dans ce chapitre, nous aborderons les différents Outils, Langages, Frameworks et les techniques utilisés dans le développement et la réalisation de notre plateforme. Par la suite, nous allons présenter quelque interfaces de notre plateforme afin de mettre en évidence leurs aspect pratique et intuitive qui nous ont été l'un de nos objectifs principaux.

4.2 Outils de développement

Lors le développement de notre plateforme, vous êtes obligé d'utiliser certains outils qui vous aident et facilitent le processus. Ensuite, nous allons découvrir certains d'entre eux qui nous aident vraiment dans le processus de développement.

4.2.1 Github

GitHub est une plateforme de développement de logiciels qui offre des services de gestion de code source, de collaboration et d'hébergement de projets. Elle est largement utilisée par les développeurs et les équipes de développement de logiciels pour suivre les modifications de code, collaborer sur des projets, gérer des problèmes, et plus encore (54).

4.2.2 Docker

Docker est une technologie de conteneurisation qui repose sur le concept de conteneurs. Les conteneurs sont des environnements d'exécution isolés qui contiennent une application et tous ses composants, tels que les bibliothèques, les dépendances et les configurations. Docker fournit une manière efficace et reproductible de créer, de distribuer et de déployer des applications, en garantissant que l'application fonctionne de la même manière dans tous les environnements, qu'il s'agisse d'un ordinateur portable de développement, d'un serveur de test ou d'un environnement de production (55).

4.2.3 Docker Compose

Docker Compose est un outil open source développé par Docker, Inc. qui permet de définir et de gérer des applications multi-conteneurs. Il simplifie la création, la configuration et le déploiement d'applications composées de plusieurs conteneurs Docker en utilisant un fichier de configuration YAML. Ce fichier décrit l'ensemble de l'architecture de l'application, y compris les images Docker à utiliser, les réseaux nécessaires, les variables d'environnement, les dépendances entre les services, etc (56).

4.2.4 Postman

Postman est un outil de développement d'API qui offre une interface conviviale pour envoyer des requêtes HTTP à des API et pour tester leur fonctionnement. Il offre une interface utilisateur graphique permettant de créer, d'envoyer et de gérer des requêtes HTTP (GET, POST, PUT, DELETE, etc.) vers des services web, des serveurs RESTful et d'autres types d'API. Postman permet également d'automatiser des tests, de gérer des collections de requêtes et de générer de la documentation pour les API (57).

4.3 Langage et Frameworks de développement

4.3.1 Coté Client (Front-end)

HTML

HTML, qui signifie HyperText Markup Language, est le langage de balisage utilisé pour créer des pages web. Il s'agit d'un langage de base pour la création de contenu sur le World Wide Web. HTML permet de structurer et de formater le contenu d'une page web en utilisant des balises (ou des éléments) pour définir la signification et la présentation de différents éléments tels que du texte, des images, des liens, des formulaires, etc (58).

CSS

CSS, ou Cascading Style Sheets, est un langage de feuille de style utilisé pour définir la présentation et la mise en forme d'une page web HTML ou XML. Il permet de séparer la structure d'une page web de sa présentation, ce qui facilite la création de sites web esthétiquement attrayants et cohérents (59).

Javascript

JavaScript est un langage de programmation de haut niveau, orienté objet et interprété. Il est principalement utilisé pour développer des applications web interactives et dynamiques. Contrairement à HTML (HyperText Markup Language) qui sert à la structuration du contenu web et à CSS (Cascading Style Sheets) pour la mise en forme, JavaScript permet de créer des fonctionnalités interactives, de gérer la logique applicative, et de communiquer avec le serveur (60).

React Js

React.js est une bibliothèque JavaScript populaire pour la construction d'interfaces utilisateur interactives et réactives. Elle a été développée par Facebook et est largement utilisée dans l'industrie du développement web. React.js est un choix puissant pour le développement d'applications web modernes en raison de sa performance, de sa modularité, de sa facilité de maintenance et de sa vaste communauté de développeurs. Il est particulièrement adapté aux applications nécessitant des interfaces utilisateur réactives et dynamiques (61).

Redux Toolkit

Redux Toolkit est une bibliothèque open-source qui simplifie la gestion de l'état dans les applications JavaScript, en particulier dans le contexte des applications React. Elle a été créée pour résoudre certains des problèmes courants rencontrés lors de l'utilisation de Redux, une bibliothèque de gestion de l'état très populaire (62).

Next Js

Next.js est un framework JavaScript populaire, puissant et flexible pour le développement web qui offre de nombreux avantages, notamment le rendu côté serveur, le pré-rendu statique, la gestion des routes, la prise en charge de React, et bien plus encore. Il est particulièrement adapté aux projets nécessitant une performance élevée, un bon référencement et une expérience utilisateur optimale (63).

Axios Js

Axios est une bibliothèque JavaScript simple et puissante pour effectuer des requêtes HTTP depuis un navigateur web ou depuis Node.js, offrant une API conviviale, une gestion efficace des promesses, une gestion améliorée des erreurs et une grande polyvalence pour les développeurs web et Node.js. C'est pourquoi il est largement utilisé dans le développement d'applications modernes (64).

Tailwind Css

Tailwind CSS est un framework CSS puissant qui offre une productivité accrue, une personnalisation facile, une maintenance simplifiée et une grande performance. Il est devenu un choix populaire parmi les développeurs web pour la création d'interfaces utilisateur modernes et réactives (65).

4.3.2 Coté Serveur (Back-end)

Python

Python est un langage de programmation puissant et populaire pour le développement web en raison de sa simplicité, de sa lisibilité, de ses frameworks solides et de sa communauté active. Il offre de nombreux avantages pour les développeurs web, qu'ils travaillent sur des projets de petite ou de grande envergure (66).

Bash

Bash est un outil puissant et polyvalent pour les développeurs web. Il permet d'automatiser des tâches, de gérer des fichiers, d'interagir avec des serveurs, de personnaliser l'environnement de travail et de travailler efficacement avec de nombreux outils en ligne de commande essentiels au développement web. Son utilisation est largement répandue dans la communauté des développeurs web pour améliorer la productivité et la gestion de projets (67).

Yaml

YAML est un langage de sérialisation de données qui permet de représenter des données sous forme de texte lisible par l'homme. Il est principalement utilisé pour configurer des applications, stocker des données structurées, et échanger des données entre différents programmes. YAML est souvent utilisé dans les domaines du développement logiciel, de la gestion de configuration, et de l'automatisation (68).

Django

Django est un framework web open-source écrit en Python qui permet de développer rapidement des applications web robustes et évolutives. Il a été créé pour simplifier le processus de développement web en fournissant des composants réutilisables et en mettant en place des conventions de développement (69).

Django Storage

Le stockage dans Django simplifie la gestion des fichiers dans une application web en fournissant une interface abstraite pour le stockage de fichiers, ce qui permet une configuration flexible et une gestion efficace des fichiers statiques et téléchargés (70).

Django Rest Framwork

Django REST framework est une bibliothèque open source qui simplifie la création d'API Web RESTful en utilisant Django, un framework web Python bien établi. Il fournit des outils et des classes pour gérer les opérations courantes d'une API REST, telles que la sérialisation et la désérialisation des données, l'authentification, l'autorisation, la gestion des vues, la pagination, la gestion des relations, etc (71).

4.3.3 Base De Données

Postgresql

PostgreSQL, souvent appelé simplement "Postgres", est un système de gestion de base de données relationnelle open source. Il est basé sur le modèle de données relationnelles, où les données sont organisées en tables avec des lignes et des colonnes. PostgreSQL prend en charge le langage SQL (Structured Query Language) pour interagir avec les données, et il offre de nombreuses fonctionnalités avancées pour la gestion des données (72).

Langage SQL

SQL (Structured Query Language) est un langage de programmation spécialement conçu pour interagir avec les bases de données relationnelles. SQL est un langage de requête qui permet de manipuler et de gérer des bases de données relationnelles. Il a été développé dans les années 1970 par IBM, et depuis lors, il est devenu un standard industriel pour les systèmes de gestion de bases de données (SGBD). Il est utilisé pour effectuer

diverses opérations sur les données stockées dans une base de données, notamment la création, la mise à jour, la suppression et la récupération (73).

4.4 Les interfaces

Dans ce qui suit, nous allons présenter quelques captures d'écran des interfaces graphiques développé de notre plateforme.

4.4.1 Interface d' accueil

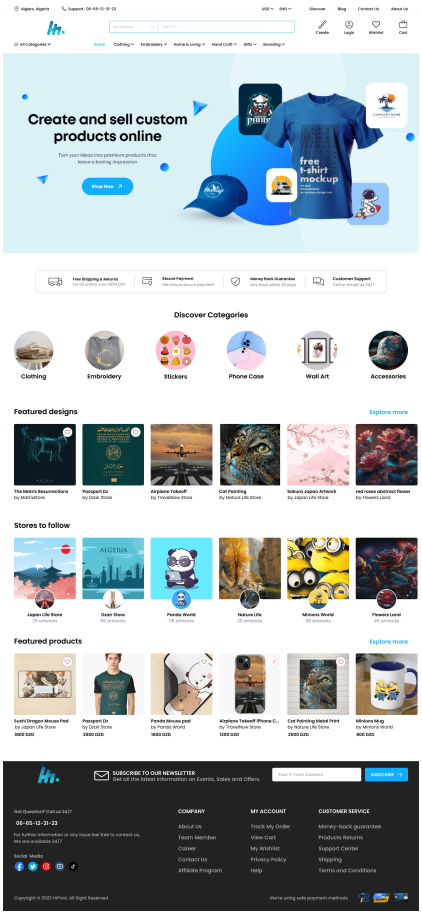

FIGURE 4.1 – Interface d' accueil

La figure représente la page d'accueil qui est l'interface principal de la plateforme. Depuis cette page, l'utilisateur pourra :

— Authentifie (inscription/connexion).
— Consulter les produits, les designs et les boutiques disponibles.
— Rechercher des produits selon les catégories.
— Accéder a le panier et la liste des favoris.
— Créer/Uploader des designs.

— Accéder a le pied de page qui contient plusieurs champs d'aides et d'informations (Contact, service clientèle, programme d'affiliation, livraison...etc).

4.4.2 Interface d'authentification

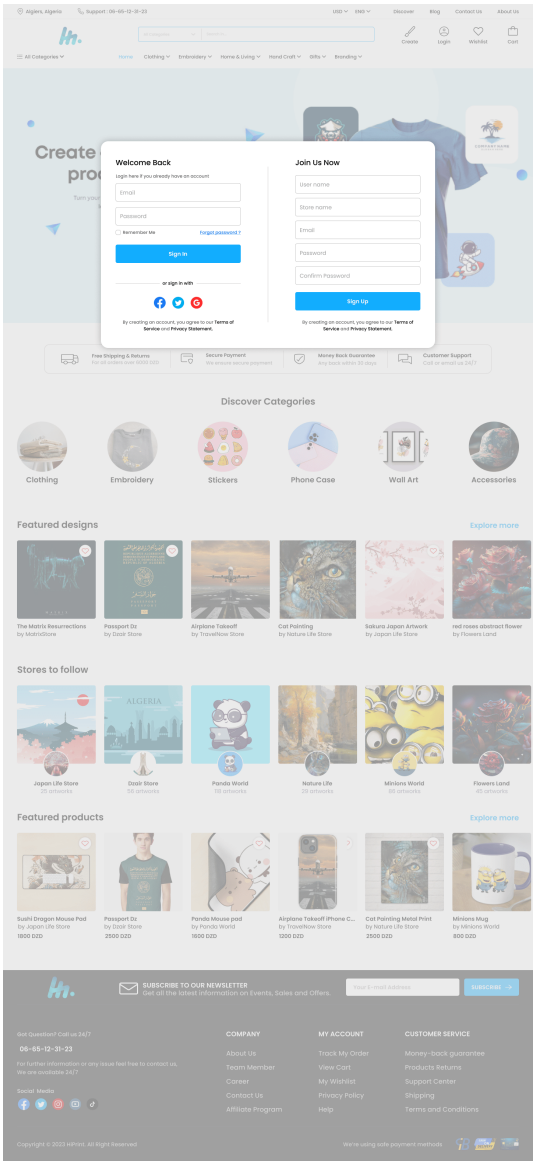

FIGURE 4.2 – Interface d'authentification

La figure représente le PopUp d'authentification qui permet a l'utilisateur de se connecter a la plateforme et accéder a son compte, ou de Créer un nouveau compte si la première visite pour l'utilisateur sur la plateforme.

4.4.3 Interface de Produit

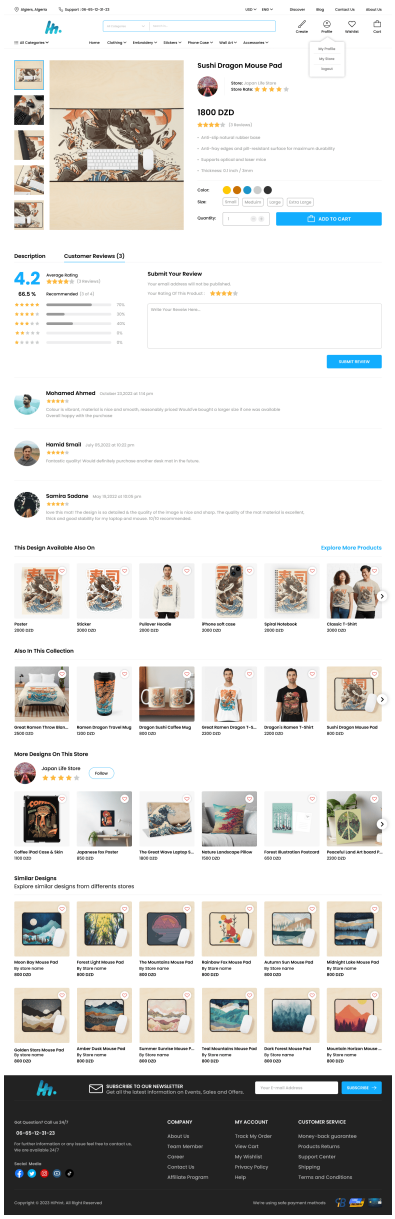

FIGURE 4.3 – Interface de Produit

La figure représente la page de produit et son spécifications. Depuis cette page, l'utilisateur pourra :

— Consulter le produit de différents aspects d'une façon réelle.
— Consulter la description de produit et son spécifications (Couleur, Taille, Quantité).
— Ajouter et Consulter les avis et l'évaluation de produit.
— Ajouter le produit au panier après de choisir les spécification approprié.
— Consulter les designs et les produits similaires.
— Accéder a la boutique propriétaire du produit.

4.4.4 Interface du Boutique (store)

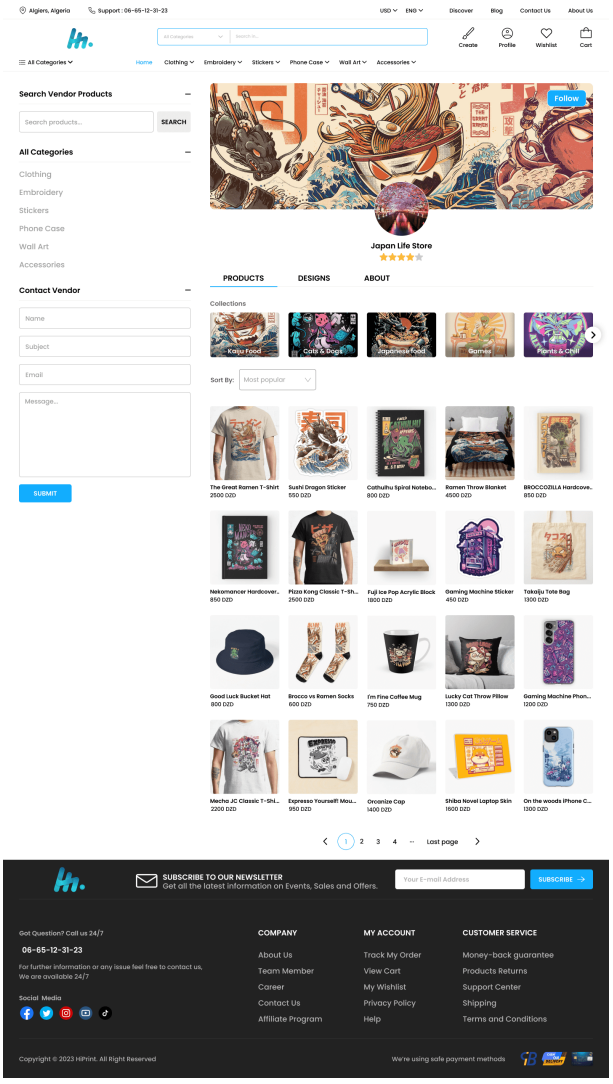

FIGURE 4.4 – Interface du boutique (store)

La figure représente la page du boutique (store), Les produits et designs disponibles. Depuis cette page, l'utilisateur pourra :

— Consulter les produits et les collections disponibles sur la boutique.
— organiser et filtrer les produits par (ventes, prix, catégorie...).
— Rechercher sur la boutique et Contacter le vendeur.

4.4.5 Interface du Design

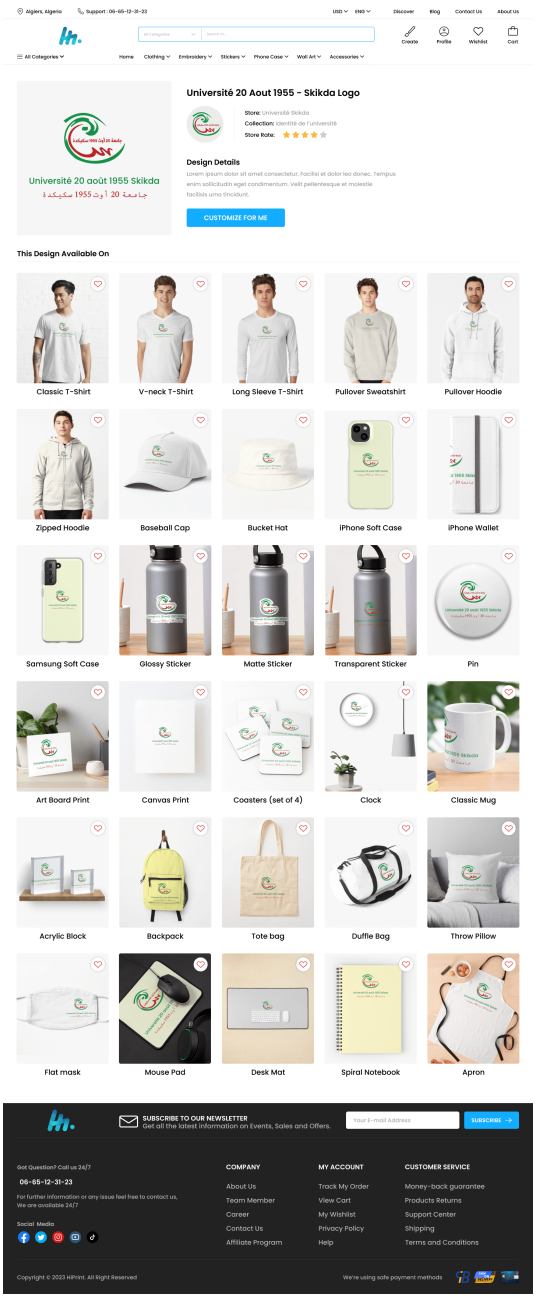

FIGURE 4.5 – Interface du Design

La figure représente la page du Design, Qui représente tous les produits disponibles avec ce design. Depuis cette page, l'utilisateur pourra :

— Consulter le design et les produits disponibles avec ce design.
— Accéder a la boutique qui propose ce design et consulter son évaluation et détails (description).
— Personnaliser et modifier le design.

4.4.6 Interface de liste des favoris

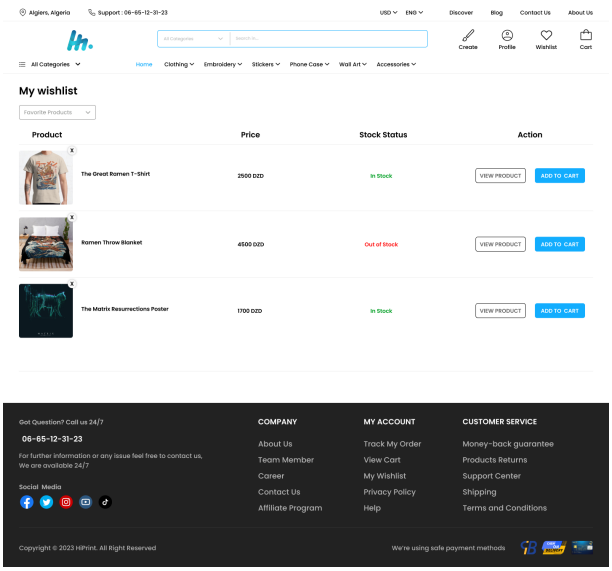

FIGURE 4.6 – Interface de liste des favoris

La figure représente la Page de liste des favoris, Qui représente les produits, les designs et les boutiques (store) préférer. Depuis cette page, l'utilisateur pourra :

— Consulter les produits, les designs et les stores favoris.
— Ajouter les produits au panier.

4.4.7 Interface de panier

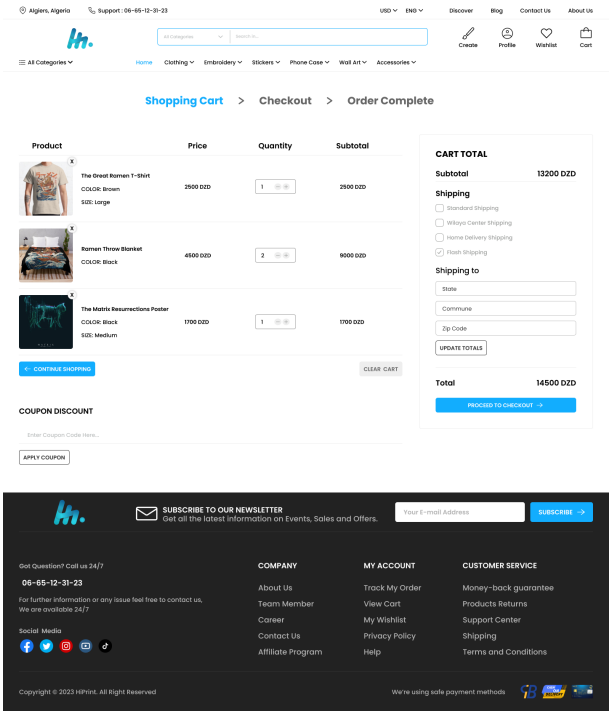

FIGURE 4.7 – Interface de panier

La figure représente la Page de panier, Qui représente les produits qui seront acheter pour procéder a l'étape du check-out du commande et le paiement. Depuis cette page, l'utilisateur pourra :

— Consulter et gérer (ajouter/supprimer) les produits ajouter au panier.
— modifier les spécification de produit (quantité, taille, couleur...etc).
— Appliquer des coupons pour bénéficier des réductions sur la somme totale.
— Choisir le type et l'adresse de livraison pour estimer les frais de livraison.

4.4.8 Interface de validation du commande et paiement

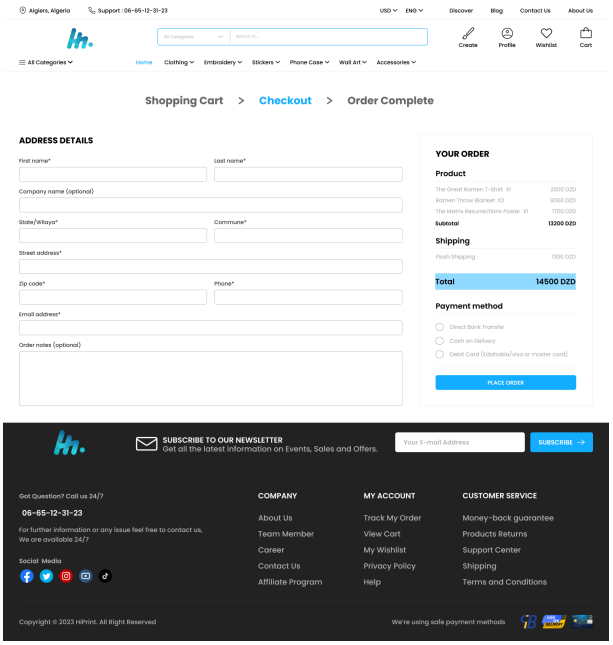

FIGURE 4.8 – Interface de validation du commande et paiement

La figure représente la Page de validation du commande et paiement. Depuis cette page, l'utilisateur pourra :

— Consulter tous les produits qui seront achetées avec son quantité et son coût.
— Consulter le type de livraison et son coût.
— Ajouter les informations personnels pour confirmer la commande et la livraison.
— Choisir le type de paiement (a la livraison, transfert bancaire ou par carte bancaire).

4.4.9 Interfaces associées au processus de création du design

Comme suit, on va montrer la processus pour effectuer une création du design et personnaliser les produits d'une façon réaliste et visuelle.

Interface du détails de design

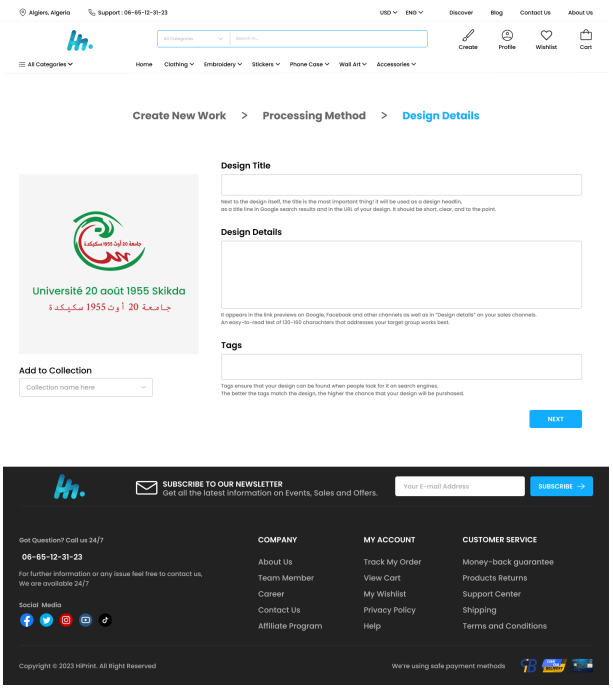

FIGURE 4.9 – Interface du détails de design

La figure représente la page du détails de design, Cette page permet au propriétaire du boutique de saisir les informations de design (le logo de l'Université de Skikda ici comme exemple), qui jouent un rôle essentiel dans l'organisation du design au niveau de la plateforme et dans le classement dans les moteurs de recherche.

Il est recommandé de choisir un titre court et expressif pour le design. En termes de description du design, il est conseillé d'inclure diverses spécifications de design afin de clarifier l'utilisateur. Enfin, en ce qui concerne les mots-clés (Tags), il est conseillé de choisir des mots-clés distinctifs et expressifs qui sont largement utilisés par les utilisateurs pour aider le design à se classer en haut des résultats de recherche et à réussir ses ventes.

Remarque : Cette interface apparaît uniquement si l'utilisateur ajoute un design à la boutique.

Interface du sélection et personnalisation des produits

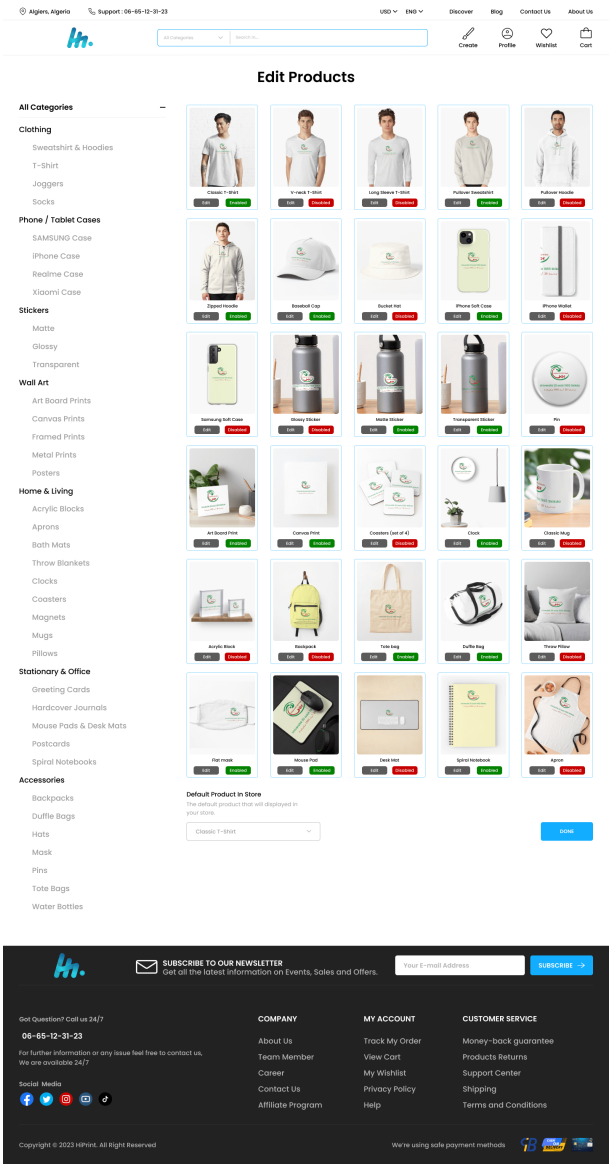

FIGURE 4.10 – Interface de sélection et de personnalisation des produits

La figure représente la page de sélection et de personnalisation des produits, Cette page permet de choisir les produits à personnaliser et à demander pour un usage personnel ou de les ajouter à la boutique personnelle sur la plateforme, où de nombreux produits apparaissent avec le design uploadé et appliqué de manière réaliste sur les produits disponibles.

Interface de création et modification des designs

Il s'agit d'un ensemble des interfaces qui permettent d'afficher le produit de manière tridimensionnelle réaliste afin de créer des designs compatibles avec le produit et d'afficher le résultat final devant l'utilisateur.

Remarque : Nous avons pris comme exemple le logo de l'Université de Skikda comme design et le T-shirt comme produit.

1 - Interface des propriétés de design

FIGURE 4.11 – Interface des propriétés de design

Cette interface vous permet de modifier diverses propriétés de design (taille, hauteur, largeur, alignement, etc.), ainsi que d'ajouter des textes et également de les contrôler ainsi que leurs propriétés.

2 - Interface des couleurs disponibles

FIGURE 4.12 – Interface des couleurs disponibles

Cette interface permet d'afficher le produit final dans les différentes couleurs disponibles pour aider à choisir le meilleur design qui correspond aux différentes choix.

3 - Interface d'emplacement de design

FIGURE 4.13 – Interface d'emplacement de design

Cette interface permet d'ajouter plusieurs designs, de les modifier et de les afficher sur le produit d'une façon réelle, en fonction des différentes dimensions du produit et de ses différents angles (dans notre exemple, le produit est un t-shirt, les designs peuvent y être ajoutés sur l'avant et l'arrière de produit uniquement, comme indiqué ci-dessus).

4 - Interface des produits disponibles

FIGURE 4.14 – Interface des produits disponibles

Cette interface permet un accès rapide aux produits disponibles sur la plateforme et de les modifier (personnaliser) à souhait.

5 - Interface de l'intelligence artificielle

FIGURE 4.15 – Interface de l'intelligence artificielle

Cette interface permet l'utilisation de l'intelligence artificielle pour faciliter le processus de création de designs en convertissant le texte en image ou en design, ce qui facilite le processus de personnalisation pour l'utilisateur moyen sans avoir besoin d'une expérience préalable en design.

4.5 Conclusion

Ce dernier chapitre décrit tout d'abord les outils de développement, les langages et les frameworks de développement que ce soit côté serveur ou côté utilisateur qui nous avons utilisé pour la réalisation de notre plateforme.

Ensuite, nous avons présenté les interfaces les plus essentiels de notre application, qui nous pensons que cette dernière répond aux besoins définit dans le chapitre précédent et aux exigences principales des utilisateurs.

Conclusion Générale

En conclusion, ce mémoire a exploré le monde en constante évolution du commerce électronique et du web, en mettant l'accent sur la conception et l'implémentation d'une plateforme e-commerce innovante dédiée à la personnalisation des produits.

À travers les différents chapitres, nous avons pu examiner en profondeur les concepts fondamentaux du e-commerce, les aspects techniques du web, et les défis et opportunités liés à la création d'une telle plateforme. Ce projet démontre l'importance de l'innovation dans le domaine du commerce en ligne, où la personnalisation est devenue un élément essentiel de l'expérience client. La plateforme que nous avons développée a le potentiel de répondre aux besoins changeants des consommateurs et des créateurs tout en offrant une solution efficace pour la gestion des transactions en ligne.

Parmi les améliorations envisagées figurent des optimisations de l'interface utilisateur, des options de personnalisation avancées pour les produits, ainsi que l'intégration de fonctionnalités de pré-visualisation en temps réel.

La sécurité des données utilisateur demeurera une priorité absolue, avec une constante mise à jour des protocoles et des dispositifs de protection. En fin de compte, notre engagement envers l'innovation et l'amélioration continuera à guider notre évolution pour répondre aux besoins changeants du marché et offrir une expérience de commerce électronique personnalisée de premier plan.

Nous sommes impatients de voir la plateforme croître et prospérer, et nous espérons qu'elle continuera à inspirer les créateurs et à satisfaire les consommateurs du monde entier.

Bibliographie

[1] I. bit, "client-server architecture diagram." [Online]. Available : https://www.interviewbit.com/blog/client-server-architecture/

[2] Amazon, "What is a dns ?" [Online]. Available : https://aws.amazon.com/route53/what-is-dns/

[3] wallarm, "Websocket protocol." [Online]. Available : https://www.wallarm.com/what/a-simple-explanation-of-what-a-websocket-is

[4] Q. Wu, K. He, and X. Chen, "From an analysis of e-services definitions and classifications to the proposal of new e-service classification," *Procedia Economics and Finance 39 (2016) 192 – 196*, 2016.

[5] Amazon, "What is ecommerce." [Online]. Available : https://sell.amazon.com/learn/what-is-ecommerce

[6] R. Goel, *E-commerce*. New Age International, 2008.

[7] D. K. E. Turban, J. Lee and H. Chung, *Electronic Commerce : A Managerial Perspective*. Prentice Hall, 1999.

[8] A. Manzoor, *E-Commerce an introduction*. LAP LAMBERT Academic Publishing, 2010.

[9] P. V. Henri Isaac, *E-Commerce de la stratégie à la mise en oeuvre opérationnelle*. Pearson Education France, 2008.

[10] theastrologypage, "Qu'est-ce que le service à la demande ? - définition de techopedia." [Online]. Available : https://fr.theastrologypage.com/demand-service

[11] H. W. G. B. Robbert-Jan van der Burg, Kees Ahaus, *Investigating the on-demand service characteristics : an empirical study Robbert*. Jouranl of Service Management, 2019.

[12] techopedia, "On-demand service." [Online]. Available : https://www.techopedia.com/definition/26711/on-demand-service

[13] Printful, "Service de print on demand." [Online]. Available : https://www.printful.com/fr/print-on-demand

[14] printful, “Guide complet pour réussir en print on demand en 2023.” [Online]. Available : https://www.printful.com/fr/blog/guide-complet-print-on-demand#h-1-1-d-finition

[15] Wizishop, “Guide du print on demand : définition, avantages, fournisseurs et tops produits.” [Online]. Available : https://www.wizishop.fr/blog/print-on-demand

[16] W3, “History of the world wide web.” [Online]. Available : https://www.w3.org/about/history/

[17] w3 org, “Hypertext proposal.” [Online]. Available : https://www.w3.org/History/1989/proposal.html

[18] wikipedia, “Hypertext types.” [Online]. Available : https://en.wikipedia.org/wiki/Hypertext

[19] M. S. A. A. S. R. David Gourley, Brian Totty, *HTTP : The Definitive Guide*, September 2002.

[20] mozilla, “Http protocol overview.” [Online]. Available : https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

[21] RFC, “Hypertext transfer protocol – http/1.1.” [Online]. Available : https://datatracker.ietf.org/doc/html/rfc2616

[22] mozilla, “Http verbs.” [Online]. Available : https://developer.mozilla.org/fr/docs/Web/HTTP/Methods

[23] Amazon, “Difference between tls and ssl.” [Online]. Available : https://aws.amazon.com/compare/the-difference-between-ssl-and-tls/

[24] RFC, “The transport layer security (tls) protocol version 1.3.” [Online]. Available : https://datatracker.ietf.org/doc/html/rfc8446

[25] O. C. S. Series, “Transport layer protection cheat sheet.” [Online]. Available : https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html

[26] S. Iniodu, “The anatomy of a url.” [Online]. Available : https://www.linkedin.com/pulse/anatomy-url-solomon-iniodu/

[27] mozilla, “What is a url ?” [Online]. Available : https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_URL

[28] RFC-3986, “Uniform resource identifier (uri) : Generic syntax.” [Online]. Available : https://datatracker.ietf.org/doc/html/rfc3986

[29] Mozilla, “Dns.” [Online]. Available : https://developer.mozilla.org/en-US/docs/Glossary/DNS

[30] RFC, “Internet protocol, darpa internet program, protocol specification.” [Online]. Available : https://datatracker.ietf.org/doc/html/rfc791

[31] D. Cortuk, “Tcp/ip overview.” [Online]. Available : https://medium.com/@derya.cortuk/tcp-ip-overview-768314d7a4c9

[32] Mozilla, “Rest.” [Online]. Available : https://developer.mozilla.org/en-US/docs/Glossary/REST

[33] S. R. Leonard Richardson, Mike Amundsen, *RESTful Web APIs*, 2008.

[34] “Graphql official website.” [Online]. Available : https://graphql.org/

[35] “Html (hypertext markup language).” [Online]. Available : https://developer.mozilla.org/fr/docs/Web/HTML

[36] Mozilla, “Css : Feuilles de style en cascade.” [Online]. Available : https://developer.mozilla.org/fr/docs/Web/CSS

[37] mozilla dev, “Javascript.” [Online]. Available : https://developer.mozilla.org/fr/docs/Web/JavaScript

[38] S. JSON, “Présentation de json.” [Online]. Available : https://www.json.org/json-fr.html

[39] Mozilla, “Manipuler des données json.” [Online]. Available : https://developer.mozilla.org/fr/docs/Learn/JavaScript/Objects/JSON

[40] W3C, “Extensible markup language (xml) 1.0 (fifth edition).” [Online]. Available : https://www.w3.org/TR/xml/

[41] Mozilla, “Introduction à xml.” [Online]. Available : https://developer.mozilla.org/fr/docs/Web/XML/XML_introduction

[42] geeksforgeeks, “mvc framework introduction.” [Online]. Available : https://www.geeksforgeeks.org/mvc-framework-introduction/

[43] developer mozilla, “composants mvc.” [Online]. Available : https://developer.mozilla.org/en-US/docs/Glossary/MVC

[44] codecademy, “ruby introduction.” [Online]. Available : https://www.codecademy.com/resources/blog/what-is-ruby-on-rails/

[45] M. Hartl, *Ruby on Rails Tutorial : Learn Web Development with Rails*, 2020.

[46] M. Stauffer, *Laravel : Up Running*, 2019.

[47] developer mozilla, “django inroduction.” [Online]. Available : https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction

[48] J. K. M. Adrian Holovaty, *The Definitive Guide to Django : Web Development Done Right*.

[49] Mozilla, "Synchronous and asynchronous requests." [Online]. Available : https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest/Synchronous_and_Asynchronous_Requests

[50] Lucidchart, "what is uml." [Online]. Available : https://www.lucidchart.com/pages/what-is-UML-unified-modeling-language

[51] L. Audibert, *Uml2 : de l'apprentissage a la pratique : cours et exercices 2e ed*, 2009.

[52] R. M. Kim Hamilton, *Learning UML 2.0*. O'Reilly Pub, 2006.

[53] A. M. Francis Buekenhout, *Diagram Geometry*, 2013.

[54] developer mozilla, "Github." [Online]. Available : https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/GitHub

[55] S. Goasguen, *Docker Cookbook : Solutions and Examples for Building Distributed Applications*. O'Reilly Media, 2016.

[56] docker docs, "docker compose." [Online]. Available : https://docs.docker.com/compose/

[57] learning postman, "postman." [Online]. Available : https://learning.postman.com/docs/getting-started/first-steps/sending-the-first-request/#:~:text=Postman%20enables%20you%20to%20create,response%20appears%20right%20inside%20Postman.

[58] developer mozilla, "html." [Online]. Available : https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics

[59] mozilla developer, "css." [Online]. Available : https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics

[60] mozilla dev, "Javascript." [Online]. Available : https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics

[61] R. Wieruch, *The Road to React : Your journey to master plain yet pragmatic React.js*, 2023.

[62] geeksforgeeks, "redux toolkit." [Online]. Available : https://www.geeksforgeeks.org/redux-toolkit/

[63] M. Riva, *Real-World Next.js : Build scalable, high-performance, and modern web applications using Next.js, the React framework for production*. Packt, 2022.

[64] reflectoring, "axios js." [Online]. Available : https://reflectoring.io/tutorial-guide-axios/

[65] N. Rappin, *Modern CSS with Tailwind : Flexible Styling without the Fuss 1st Edition*. Pragmatic Bookshelf, 2021.

[66] D. C. R. Severance, *Python for Everybody : Exploring Data in Python 3*, 2016.

[67] gnu org, "bash." [Online]. Available : https://www.gnu.org/software/bash/manual/bash.html

[68] redhat, "yaml." [Online]. Available : https://www.redhat.com/en/topics/automation/what-is-yaml#yaml-syntax

[69] W. S. Vincent, *Django for Beginners : Build websites with Python and Django*, 2023.

[70] scaler, "django storage." [Online]. Available : https://www.scaler.com/topics/django/django-storages/

[71] W. S. Vincent, *REST APIs with Django : Build powerful web APIs with Python and Django*, 2018.

[72] N. M. Richard Stones, *Beginning Databases with PostgreSQL : From Novice to Professional (Beginning From Novice to Professional) 2nd Edition*. Apress, 2005.

[73] J. C. LCF Publishing, *SQL : Learn SQL (using MySQL) in One Day and Learn It Well*, 2018.