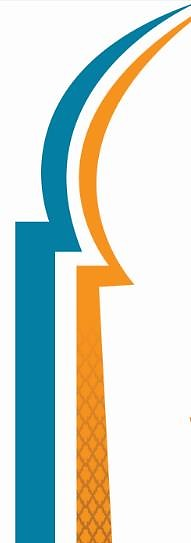
*UNIVERSITÉ* **MOULAY ISMAIL**



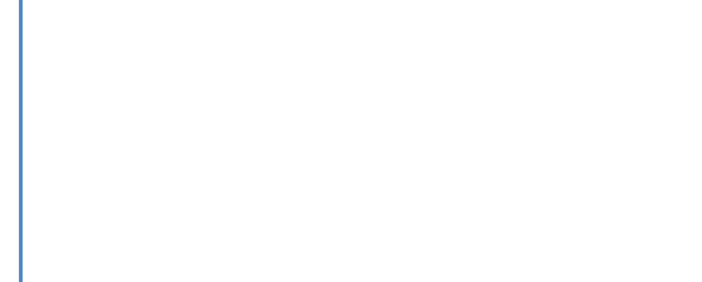
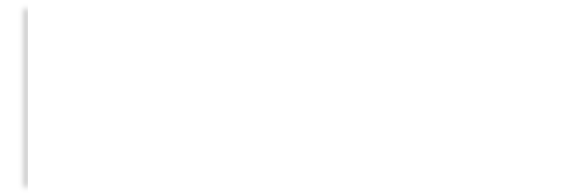
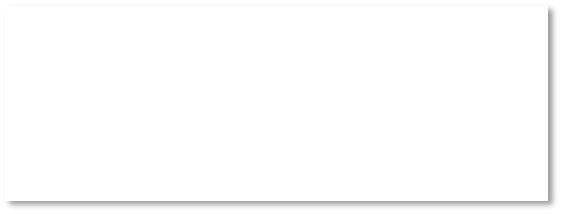
**FACULTÉ DES SCIENCES** ET TECHNIQUE D’ERRACHIDIA

*Département d’informatique* - **LST Génie logiciel**





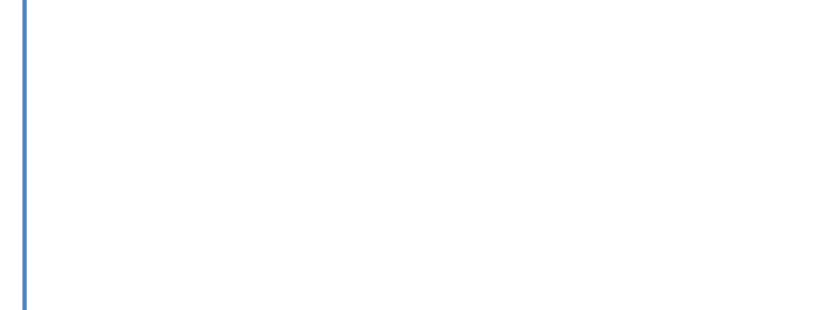
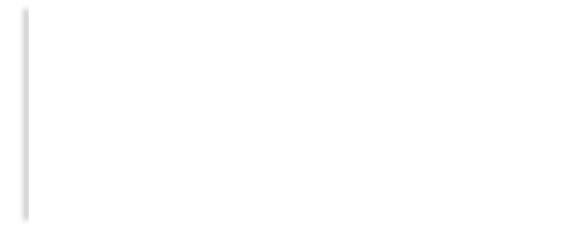
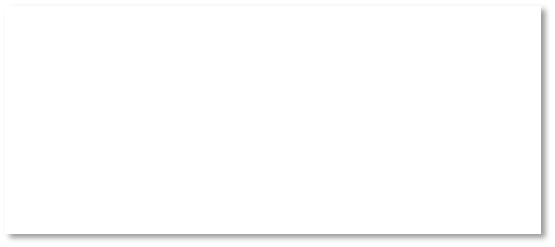
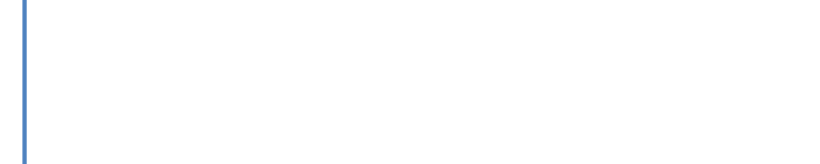
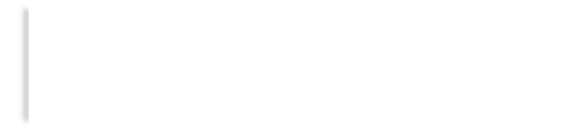
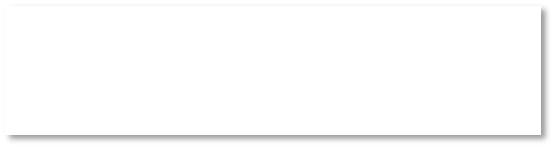
PROJET DE FIN D’ETUDE (PFE) **: Réalisation d’une Application web e-commerce « JOURYA »**



*Réalisation*

**OURYA** AYOUB

**FIADEHOUNDJI** KOKOU JAQCUES MAWUENA



*Encadrant*

Pr **YAAKOUBI** Fouad

*Membres du jury*

Pr Ouchao **Brahim** (Président)

Pr Taifi **Khaddouj** (Membre)

**ANNEE UNIVERSITAIRE : 2022-2023**

DEDICACE



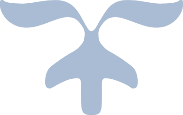
**Une pensée forte,**

***À*** *nos familles et à nos parents, qui ont été les piliers de notre parcours et nous ont soutenus inconditionnellement. Leur croyance en nous et leur encouragement constant nous ont permis de surmonter les obstacles et de devenir les meilleures versions de nous-mêmes.*

***À*** *tous ceux qui ont contribué, de près ou de loin, à notre succès. Que ce soit par des conseils précieux, des encouragements chaleureux ou des opportunités offertes, vous avez joué un rôle essentiel dans notre développement personnel et professionnel.*

***À*** *nos professeurs, nos amis et toutes les personnes qui croisent notre chemin au quotidien, vous nous enseignez la valeur de la persévérance et de l'effort constant. Votre présence nous inspire et nous motive à atteindre de nouveaux sommets.*

***C'est avec une profonde gratitude, un amour sincère et une admiration sans limites que nous vous dédions ces mots. Vous êtes les témoins de notre succès, et nous vous exprimons notre plus profonde reconnaissance et notre plus haute estime !***



REMERCIEMENTS

Tout d'abord, nous exprimons notre profonde gratitude à Allah, le Tout-Puissant, le Très Miséricordieux, qui nous a accordé une bonne santé et nous a permis de mener à bien cette tâche difficile dans les meilleures conditions.

Nous souhaitons remercier nos parents, nos frères et nos sœurs qui ont toujours été à nos côtés, nous soutenant dans nos choix et nous apportant leur soutien lors des moments les plus difficiles.

Nos sincères remerciements vont également à notre professeur encadrant, le **Pr** **Fouad** **Yaakoubi**, qui a consacré de nombreux efforts pour nous accompagner tout au long de la réalisation de notre projet. Ses précieux conseils et son soutien infaillible nous ont permis de surmonter les obstacles et de progresser.

Nous tenons à exprimer notre reconnaissance envers tout le personnel administratif et le corps enseignant de l'Université Moulay Ismaël, en particulier de la Faculté des Sciences et Techniques d'Errachidia, qui se battent chaque jour pour transmettre leur savoir à de jeunes étudiants, indépendamment de leurs origines ou de leurs antécédents.

Nous n'oublions pas de remercier les éminents membres du jury qui ont accepté de consacrer leur temps et leurs compétences pour évaluer notre humble travail.

Enfin, nous adressons nos remerciements les plus chaleureux à tous ceux qui ont honoré notre projet de leur présence. Votre soutien et votre présence nous touchent profondément et nous inspirent à continuer de nous dépasser.

**Sommaire**

[**DEDICACE** 1](#_Toc139802602)

[**REMERCIEMENTS** 2](#_Toc139802603)

[**LISTE DES TABLEAUX** 5](#_Toc139802604)

[**LISTE DES FIGURES** 6](#_Toc139802605)

[**LISTE DES ABREVIATIONS** 7](#_Toc139802606)

[**INTRODUCTION GENERALE** 8](#_Toc139802607)

[**CHAPITRE I : CADRE GENERAL DU PROJET** 9](#_Toc139802608)

[1. Introduction 9](#_Toc139802609)

[2. Etude préliminaire 9](#_Toc139802610)

[2.1 Les sites web 9](#_Toc139802611)

[2.2 Les applications web 10](#_Toc139802612)

[2.3 Les applications web e-commerce 10](#_Toc139802613)

[3. Etude de l’existant 11](#_Toc139802614)

[4. Objectifs et spécifications de besoins 11](#_Toc139802615)

[4.1 Points forts de l’application Web 11](#_Toc139802616)

[4.2 Points faibles de l’application web 12](#_Toc139802617)

[5. Présentation du projet 12](#_Toc139802618)

[5.1 Plan de travail 12](#_Toc139802619)

[5.2 Diagramme de GANTT 13](#_Toc139802620)

[6. Conclusion 14](#_Toc139802621)

[**CHAPITRE II : CONCEPTION DE PROJET** 16](#_Toc139802622)

[1. Introduction 16](#_Toc139802623)

[2. Analyse et conception 16](#_Toc139802624)

[2.1 Cycle de vie 16](#_Toc139802625)

[3. Modélisation avec le langage UML 17](#_Toc139802626)

[3.1 Langage UML 17](#_Toc139802627)

[3.2 Identification des acteurs 18](#_Toc139802628)

[3.3 Diagramme de cas d’utilisation 18](#_Toc139802629)

[3.4 Diagramme de séquence 21](#_Toc139802630)

[3.5 Diagramme de classe 25](#_Toc139802631)

[4. Conclusion 27](#_Toc139802632)

[**CHAPITRE III : REALISATION DU PROJET** 28](#_Toc139802633)

[1. Introduction 28](#_Toc139802634)

[2. Langages et outils de développement 28](#_Toc139802635)

[2.1 Langages et scripts 28](#_Toc139802636)

[a. HTML5 28](#_Toc139802637)

[b. CSS3 28](#_Toc139802638)

[c. JavaScript 28](#_Toc139802639)

[d. PHP 29](#_Toc139802640)

[e. MySQL 29](#_Toc139802641)

[f. SQL 29](#_Toc139802642)

[2.2 Outils de développement 29](#_Toc139802643)

[a. Visual studio code : 29](#_Toc139802644)

[b. WampServer : 29](#_Toc139802645)

[c. StarUML 30](#_Toc139802646)

[d. Microsoft Word, PowerPoint, Project 30](#_Toc139802647)

[e. PicPick 30](#_Toc139802648)

[f. Git 30](#_Toc139802649)

[g. GitHub 31](#_Toc139802650)

[2.3 Frameworks 31](#_Toc139802651)

[2.4 Architecture de l’application web 31](#_Toc139802652)

[2.5 Approche de Programmation : POO 35](#_Toc139802653)

[3. Maquettes de l’application 35](#_Toc139802654)

[3.1 Interface d’accueil d’un Internaute 35](#_Toc139802655)

[3.2 Interface Login 37](#_Toc139802656)

[3.3 Interface d’acceuil d’un client 38](#_Toc139802657)

[3.4 Interface d’une catégorie 38](#_Toc139802658)

[3.5 Interface d’un produit 39](#_Toc139802659)

[3.6 Interface Panier 39](#_Toc139802660)

[3.7 Interface du payement 40](#_Toc139802661)

[3.8 Interface mon compte d’un client 41](#_Toc139802662)

[3.9 Interface mon compte de l’admin 42](#_Toc139802663)

[a. Page des commandes 42](#_Toc139802664)

[b. Page des clients 43](#_Toc139802665)

[c. Page des produits 43](#_Toc139802666)

[d. Page des catégories 44](#_Toc139802667)

[e. Page des informations personnelle 44](#_Toc139802668)

[4. Conclusion 44](#_Toc139802669)

[**CONCLUSION ET PERSPECTIVES** 45](#_Toc139802670)

[**BIBLIOGRAPHIE & WEBOGRAPHIE** 46](#_Toc139802671)

LISTE DES TABLEAUX

[Tableau 1 - Enchainement des tâches du projet 13](#_Toc139714888)

[Tableau 2 - Frise chronologique 13](#_Toc139714889)

LISTE DES FIGURES

[Figure 1 - Transfert avec un site statique 9](#_Toc139714662)

[Figure 2 - Transfert avec un site dynamique 10](#_Toc139714663)

[Figure 3 - Diagramme de GANTT 14](#_Toc139714664)

[Figure 4 - Cycle de vie en V (source img: https://www.manager-go.com/gestion-de-projet/cycle-en-v.htm) 16](#_Toc139714665)

[Figure 5 - Les diagrammes d'UML 2.5 17](#_Toc139714666)

[Figure 6 - diagramme de cas d'utilisation 18](#_Toc139714667)

[Figure 7 - détails sur les cas d'utilisation d’un internaute 19](#_Toc139714668)

[Figure 8 - détail sur les cas d'utilisation de client 19](#_Toc139714669)

[Figure 9 - détail sur les cas d'utilisation de l’admin 20](#_Toc139714670)

[Figure 10 - Diagramme de séquence du cas d’utilisation “se connecter ” 21](#_Toc139714671)

[Figure 11 - Diagramme de séquence du cas d’utilisation “s’inscrire ” 22](#_Toc139714672)

[Figure 12 - Diagramme de séquence du cas d’utilisation “Acheter un produit ” 23](#_Toc139714673)

[Figure 13 - Diagramme de séquence du cas d’utilisation “Ajout d’un produit ” 23](#_Toc139714674)

[Figure 14 - Le diagramme regroupant les classes et les relations entre elles 25](#_Toc139714675)

[Figure 15 - Interface d’accueil - vue1 30](#_Toc139714676)

[Figure 16 - Interface d’accueil - vue2 30](#_Toc139714677)

[Figure 17- Interface d’accueil - vue3 31](#_Toc139714678)

[Figure 18 - Interface login 31](#_Toc139714679)

[Figure 19 - Interface sign up 32](#_Toc139714680)

[Figure 20 - Interface d'acceuil Client 32](#_Toc139714681)

[Figure 21 - Interface d'une catégorie 33](#_Toc139714682)

[Figure 22 - Interface d'un produit 33](#_Toc139714683)

[Figure 23 - Interface Panier 34](#_Toc139714684)

[Figure 24 - Interface payement 34](#_Toc139714685)

[Figure 25 - Interface payement complet 35](#_Toc139714686)

[Figure 26 - Page des informations des commandes 35](#_Toc139714687)

[Figure 27 - Page des informations personnelle 35](#_Toc139714688)

[Figure 28 - Page de gestion des commandes 36](#_Toc139714689)

[Figure 29 - Page de gestion des clients 36](#_Toc139714690)

[Figure 30 - Page de gestion des produits 37](#_Toc139714691)

[Figure 31 - Page de gestion des catégories 37](#_Toc139714692)

[Figure 32 - Page des informations personnelle d'un admin 38](#_Toc139714693)

LISTE DES ABREVIATIONS

**LST :** Licence en Sciences et Techniques

**MS Project :** Microsoft Project

**UML :** Unified Modified Language

**ClassD :** Diagramme de classe

**CU:** Cas d’utilisation

**UseCaseD :** Diagramme de cas d’utilisation

**SequenceD :** Diagramme de séquence

**HTML:** HyperText MarkupLanguage

**CSS:** Cascading Style Sheets

**PHP:** Hypertext Preprocessor

**POO :** Programmation Orientée Objet

**MySQL : My** fait référence au prénom de la fille du Co-créateur Michael Widenius et

**SQL** : Structured Query Language

**SGBD :** Système de Gestion de Base de Données

**IDE :** Integrated Development Environment (environnement de développement intégré)

…

INTRODUCTION GENERALE

Dans un monde de plus en plus numérique, le commerce électronique est en pleine expansion. Les consommateurs recherchent des moyens pratiques et sécurisés pour effectuer leurs achats en ligne, tandis que les entreprises cherche nt à étendre leur portée et à atteindre un public plus large. C'est dans ce contexte dynamique que notre projet de conception de notre site web e-commerce JOURYA prend vie.

Notre plateforme vise à offrir une solution complète pour les achats en ligne, en mettant l'accent sur la convivialité, la variété des produits et la facilité d'utilisation. Que ce soit pour l'achat de vêtements, d'électronique, d'articles ménagers ou de produits spécialisés, notre site web offrira une gamme diversifiée de produits provenant de différents fournisseurs, répondant ainsi aux besoins et aux préférences variés des consommateurs.

Nous mettrons l'accent sur la création d'une interface attrayante et ergonomique,. Les utilisateurs pourront naviguer facilement à travers les catégories de produits, consulter les descriptions détaillées, comparer les prix et effectuer leurs achats en toute confiance.

Nous sommes déterminés à offrir une expérience d'achat en ligne exceptionnelle, en plaçant la satisfaction des utilisateurs au cœur de notre démarche.

Même si comme tous les projets, celui-ci est destiné à évoluer au fil du temps, nous sommes convaincus que notre site web e-commerce apportera une valeur ajoutée aux consommateurs, en créant un environnement favorable aux transactions sécurisées. Nous sommes impatients de vous présenter notre plateforme et de vous accompagner dans vos achats en ligne, tout en explorant les opportunités offertes par le monde passionnant du commerce électronique.

CHAPITRE I : CADRE GENERAL DU PROJET

# Introduction

Aujourd'hui, la digitalisation a transformé en profondeur les pratiques de consommation et les modes de vente à l'échelle mondiale. Dans ce contexte, les entreprises ont vu dans Internet une opportunité de s'implanter sur la toile pour toucher une clientèle de plus en plus connectée. Les sites e-commerce se sont alors imposés comme des outils incontournables pour les entreprises souhaitant élargir leur portée et conquérir de nouveaux marchés.

*Revenons-en au sujet…*

Pour réussir un projet de développement d'un site e-commerce, il est primordial d'analyser et de bien spécifier les besoins du client. Notre objectif est de concevoir un site web qui réponde à toutes les exigences d'un bon site e-commerce. Cela inclut des critères tels que l'ergonomie de l'interface utilisateur, la sécurité des transactions, la gestion efficace des produits, la facilité de navigation, les performances du site, et bien d'autres encore. Notre équipe s'engage à réaliser une analyse approfondie des besoins, à suivre une méthodologie rigoureuse et à utiliser les outils appropriés afin de garantir la création d'un logiciel de haute qualité qui répond aux attentes du client et des utilisateurs finaux.

Dans ce rapport, nous allons présenter la réalisation d'un site e-commerce en exposant les différents choix techniques et fonctionnels ainsi que les points forts et les limites de notre système. Nous décrirons également les différentes phases de développement de ce projet, en mettant l'accent sur les étapes clés telles que l'élaboration du cahier des charges, le développement et la mise en production du site.

# Etude préliminaire

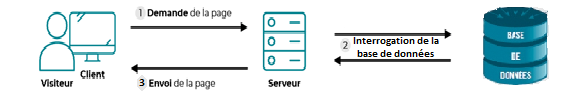
Pour ce projet, et compte tenu du temps, nous avons réalisé un site web (dynamique). Un site web est accessible depuis un navigateur.

## Les sites web

***Qu'est-ce qu'un site web ?***

Un site web est un ensemble de pages et de ressources stockées sur un serveur et accessible via un navigateur web. Il existe deux principaux types de sites web :

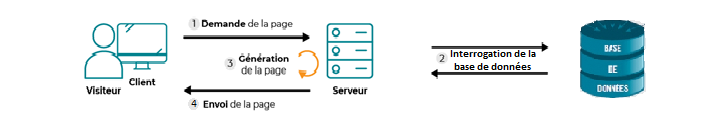
* **Les sites web statiques :** Ce sont des sites constitués de pages qui s'affichent de la même manière pour tous les utilisateurs. Les informations sont préalablement créées et le contenu ne change pas en fonction de l'utilisateur. Les sites web statiques sont généralement adaptés pour présenter des informations qui ne nécessitent pas de mises à jour fréquentes.



**Figure 1 - Transfert avec un site statique**

* **Les sites web dynamiques :** Ces sites sont interactifs et leur contenu peut changer en fonction de l'utilisateur ou d'autres variables. Ils sont généralement liés à une base de données et utilisent des langages serveur tels que PHP, JavaScript (Node.js), Java (J2EE), Python (Django, Flask) pour générer des pages dynamiques. Les sites web dynamiques sont utilisés pour des applications plus complexes telles que les systèmes de gestion, les réseaux sociaux, les plateformes de commerce électronique, etc.

Ces sites web dynamiques sont devenus plus avancés avec le temps, évoluant vers des applications web plus complètes et interactives.



**Figure 2 - Transfert avec un site dynamique**

## Les applications web

***Qu'est-ce qu'une application web ?***

Une application web peut être considérée comme une forme plus évoluée d'un site web. Alors que les sites web fournissent principalement des informations statiques, les applications web offrent des fonctionnalités interactives et une expérience utilisateur plus riche.

Les applications web utilisent des technologies telles que HTML, CSS et JavaScript, et peuvent être développées avec des langages serveur comme PHP, Python (Django, Flask), Ruby (Ruby on Rails), permettant ainsi une interaction avec des bases de données et une personnalisation du contenu en fonction des actions de l'utilisateur.

L'avantage des applications web réside dans leur accessibilité depuis n'importe quel appareil doté d'un navigateur web, sans nécessiter de téléchargement ni d'installation d'une application spécifique. Elles sont également plus faciles à déployer et à mettre à jour, offrant ainsi une expérience utilisateur plus fluide et réactive.

Les applications web peuvent être utilisées dans une variété de domaines, allant des applications de productivité aux systèmes de gestion d'entreprise, en passant par les plateformes de commerce électronique.

## Les applications web e-commerce

***Qu'est-ce qu'une application web e-commerce ?***

En plus des applications web, il existe également des applications web e-commerce, qui sont des plateformes en ligne permettant la vente de produits ou de services. Une application web e-commerce combine les fonctionnalités d'une application web dynamique avec des fonctionnalités spécifiques liées à la vente en ligne.

***Les caractéristiques d'un site e-commerce :***

Le site e-commerce est un outil complexe, qui nécessite une bonne connaissance des besoins et des attentes des clients ainsi que de l’environnement concurrentiel. Nous avons choisi de développer un site e-commerce en raison de sa popularité grandissante et de sa facilité d’accès pour les clients. Les clients peuvent effectuer des achats depuis n'importe quel lieu, à tout moment de la journée.

Il existe différents types de sites e-commerce, notamment :

* **Les sites de vente en ligne pure** **:** ces sites ne disposent pas de magasin physique et commercialisent uniquement des produits en ligne.
* **Les sites de vente en ligne avec magasin physique :** ces sites sont des extensions des magasins physiques et permettent aux clients de consulter et d'acheter les produits en ligne.
* **Les sites de marché en ligne :** ces sites permettent à des vendeurs tiers de vendre leurs produits sur la plateforme en ligne.

Pour notre projet, nous avons opté pour un site de vente en ligne pure. Cela nous permet de proposer un large choix de produits et services, sans les contraintes liées à un magasin physique.

# Etude de l’existant

L'analyse de l'existant est une étape importante dans la conception d'un site web e-commerce. En effet, la concurrence est féroce dans ce domaine, et il est essentiel de comprendre les forces et les faiblesses des sites concurrents pour concevoir une plateforme qui répond aux besoins des utilisateurs et qui se démarque de la concurrence. En étudiant les sites web e-commerce existants, nous avons pu identifier les principales fonctionnalités attendues par les clients, telles que la facilité de navigation, la présentation claire des produits, la sécurité des paiements et la livraison rapide. Nous avons également constaté que les sites les plus performants offrent des fonctionnalités supplémentaires, comme un service client disponible 24h/24 et 7j/7, des recommandations de produits basées sur les préférences des clients et des outils d'analyse de données pour comprendre le comportement des utilisateurs. Nous proposons donc une plateforme de commerce électronique qui intègre toutes ces fonctionnalités pour offrir une expérience utilisateur optimale et une différenciation compétitive sur le marché en ligne.

# Objectifs et spécifications de besoins

## Points forts de l’application Web

Le site e-commerce que nous souhaitons développer est spécifiquement conçu pour notre entreprise afin de faciliter la vente de nos produits en ligne. **Nous avons mis l'accent sur la rapidité et la simplicité du traitement des données pour offrir une expérience fluide et efficace**. Voici quelques points forts de notre solution de site e-commerce :

Interaction client-entreprise : Notre site favorise l'échange direct avec nos clients, leur permettant de naviguer, poser des questions et passer des commandes en ligne.

Gestion efficace : Notre site simplifie la gestion des stocks, des commandes, des livraisons et des paiements, optimisant ainsi notre activité.

Sécurité des données : Nous mettons en place des protocoles pour protéger les informations personnelles et financières de nos clients, assurant ainsi la confidentialité et la sécurité.

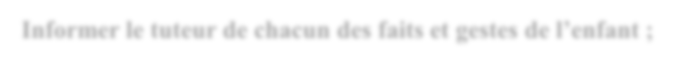
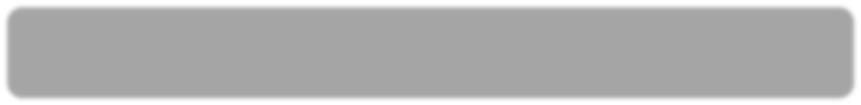
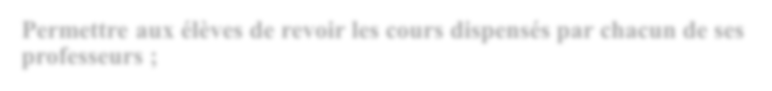
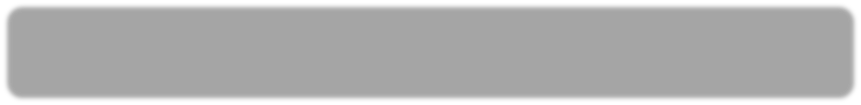
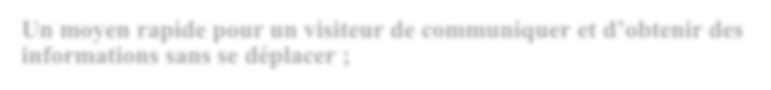
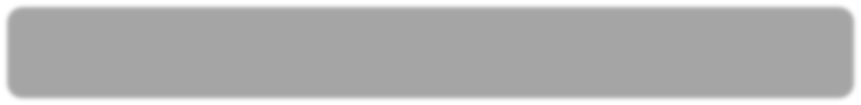
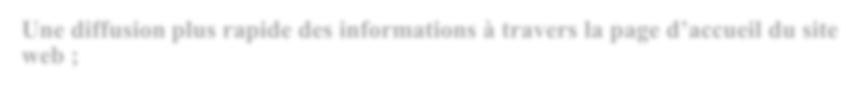
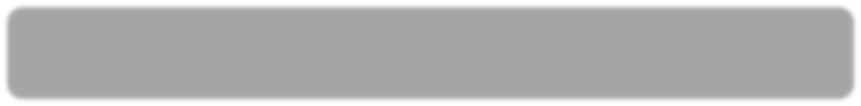
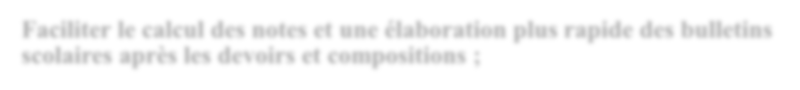
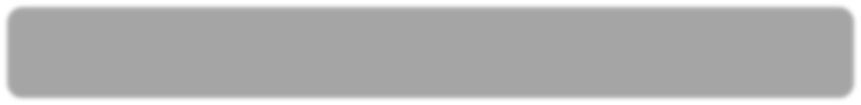
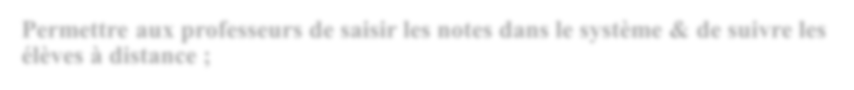
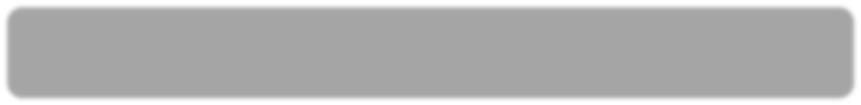
Suivi et communication : Notre solution permet un suivi efficace des commandes, des notifications de livraison et une communication rapide avec nos clients.

Accessibilité et commodité : Notre site offre la possibilité de faire des achats à tout

moment et de n'importe où, avec des options de livraison flexibles.

Affichage des produits tendances : Notre site met en avant les produits les plus populaires

et tendances, permettant aux clients de découvrir facilement les dernières nouveautés.





Promotions : Nous proposons régulièrement des offres spéciales, des réductions et des promotions sur certains produits, offrant ainsi des opportunités d'achat avantageuses à nos clients.



Affichage responsive : Les différentes interfaces de notre site s’affiche et s’adaptent d’une manière intuitive à la taille des différents écrans. Ainsi que ce soit un ordinateur, un smartphone ou d’autres appareils utilisés, une expérience utilisateur notamment la lisibilité optimale est garantie chez le client.

***Et oui, toutes ces fonctionnalités sont offertes par JOURYA…***

## Points faibles de l’application web

Notre site bien que remplissant plusieurs fonctionnalités n’exclut pas une certaine limite. En effet ce système regroupe beaucoup de fonctionnalités, cependant pas toutes les exigences pouvant exister au sein d’un site e-commerce. Par analogie à d’autres sites e-commerces comme **Jumia**, on remarque l’absence de certaines fonctionnalités.

Jumia est une entreprise de commerce en ligne présente sur le marché africain. Fondée en 2012, Jumia vend des produits électroniques, d’hygiène, de l’alimentaire et des services. (<https://fr.wikipedia.org/wiki/Jumia>).

Jumia propose des fonctionnalités plus avancées plus poussés comme :

* Un filtre de recherche puissant qui permet aux utilisateurs de préciser leurs critères de recherche, tels que la marque, la taille, le prix, etc.
* Un processus de remboursement facile et transparent en cas de problème avec une commande. Si un utilisateur rencontre un problème avec un produit, comme un défaut ou une non-conformité, il peut demander un remboursement
* Une fonctionnalité de wishlist, également appelée liste de souhaits, qui permet aux utilisateurs de sauvegarder des produits qu'ils souhaitent acheter ultérieurement.

Les fonctionnalités manquantes dans notre site ne remettent nullement en question l’efficacité de celui-ci car toutes ces fonctionnalités peuvent être ajoutées avec le temps et les moyens nécessaires !

# Présentation du projet

Notre projet de fin d'études consiste à réaliser un site web e-commerce "Jourya" pour permettre aux clients de parcourir une large sélection de produits en ligne et de passer des commandes. Le site sera divisé en deux grandes parties pour faciliter la gestion et l'utilisation pour les utilisateurs :

**• Volet administratif :** Cette partie sera réservée aux administrateurs du site, qui pourront gérer les comptes des utilisateurs, les catégories, les produits disponibles, les commandes, les livraisons.

**• Volet client :** Les clients pourront parcourir les produits disponibles, les ajouter à leur panier, passer des commandes et effectuer des paiements en ligne en toute sécurité. Les clients pourront également gérer leur compte et suivre l'état de leur commande.

## Plan de travail

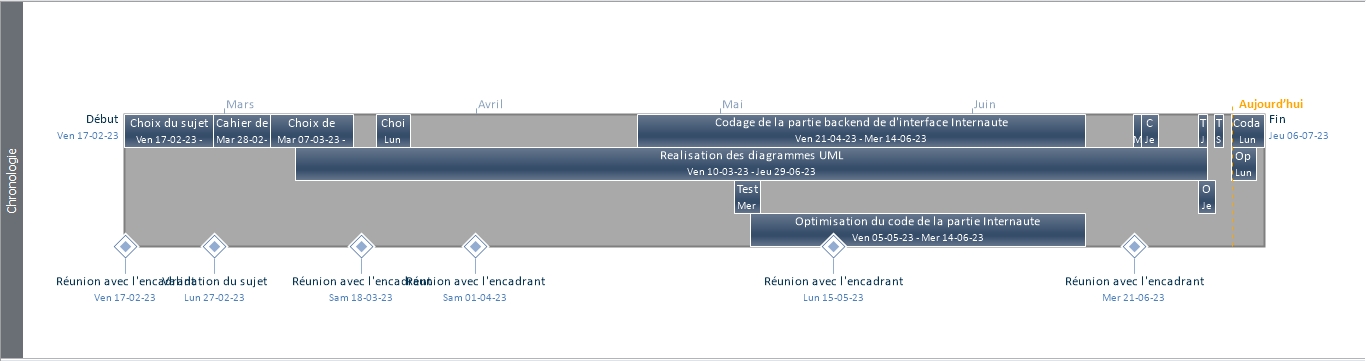
Ce projet a été réalité pendant plusieurs jours. Il a débuté par une phase d’analyse et d’étude de la faisabilité, puis une validation. L’application web en elle-même a été découpée en 2 interfaces (interface des Clients, interface des Admins) qui ont été réalisés l’un après l’autre.

Des réunions avec l’encadrant ont été réalisées des fois pour valider chaque prototype utilisable de la plateforme.

Le tableau suivant regroupe l’enchainement des tâches réalisées avec MS Project (Voir chapitre *Réalisation du projet* pour plus de détails sur ce logiciel).



**Tableau 1 - Enchainement des tâches du projet**

****

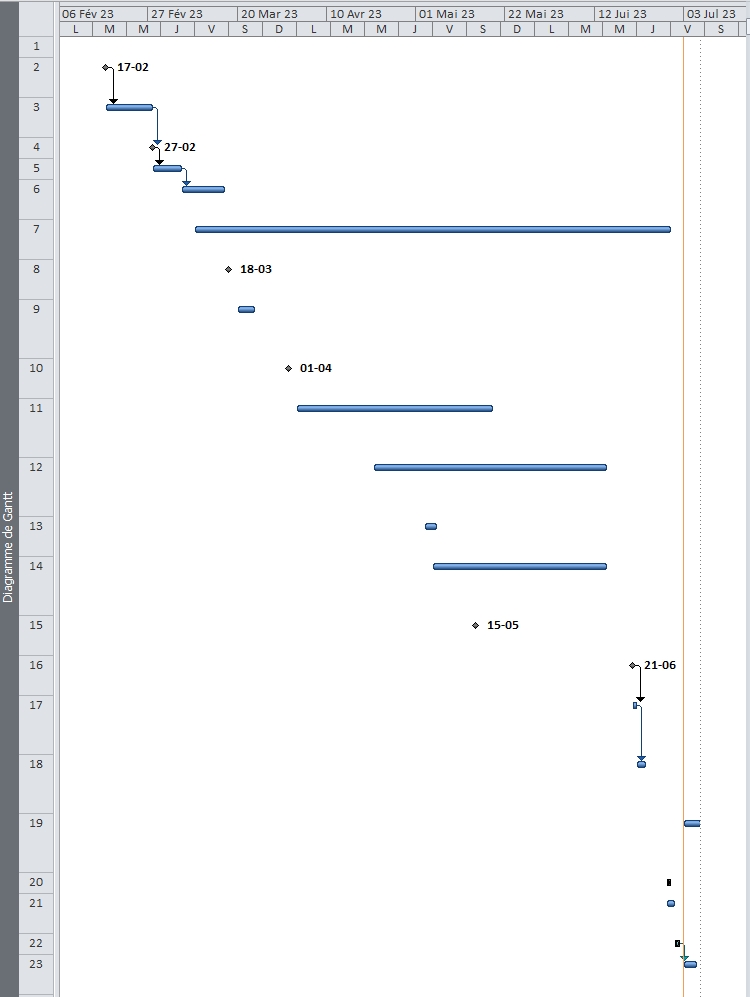
**Tableau 2 - Frise chronologique**

## Diagramme de GANTT

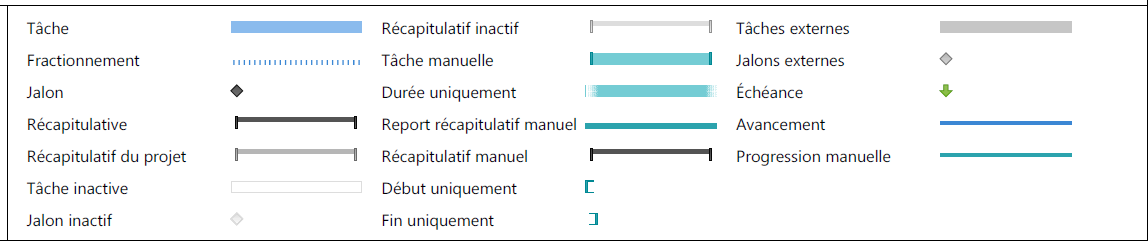
Le diagramme de GANTT est une visualisation graphique qui représente la séquence temporelle des différentes tâches d'un projet. Il offre une représentation visuelle permettant de visualiser plusieurs aspects clés, notamment :

* L'évolution des tâches au fil du temps, mettant en évidence leur progression et leur durée.
* Le suivi du taux d'avancement des tâches, permettant de mesurer le degré de réalisation par rapport au planning prévu.
* L'identification des éventuels retards ou avances dans le projet, facilitant ainsi la gestion des délais et des échéances.
* L'affectation des ressources à chaque tâche, permettant de visualiser les ressources nécessaires pour accomplir chaque activité.
* La représentation d'autres événements importants tels que les réunions et les tests, afin de mieux planifier et coordonner les différentes étapes du projet.

Dans le contexte de notre projet, le diagramme de GANTT se présente sous la forme suivante :



**Figure 3 - Diagramme de GANTT**

**Légende du diagramme de GANTT :**

# Conclusion

En résumé, notre projet de site e-commerce Jourya a été conçu pour répondre aux besoins des consommateurs modernes en matière de shopping en ligne. Nous avons identifié les fonctionnalités essentielles nécessaires pour offrir une expérience utilisateur optimale, telles que la recherche de produits, la gestion de panier d'achat, la gestion de compte utilisateur.

Nous avons également pris en compte les caractéristiques spécifiques d'un site e-commerce, telles que la sécurité des transactions et la gestion des stocks. En utilisant une architecture technique robuste, nous avons pu garantir la disponibilité, la rapidité et la fiabilité de notre site.

CHAPITRE II : CONCEPTION DE PROJET

# Introduction

La conception est une étape cruciale dans la réalisation d'un projet, car elle permet de passer de la définition initiale à la mise en œuvre concrète. Cette phase de préparation et d'organisation consiste à structurer, organiser et planifier les éléments précédemment définis. Elle requiert l'utilisation de méthodes et d'outils adaptés, ainsi qu'une modélisation claire et précise de l'ensemble des composantes du projet. Dans ce chapitre, nous allons présenter la méthode et les outils utilisés pour concevoir notre projet de site e-commerce "Jourya", ainsi que les diagrammes statiques et dynamiques qui ont été élaborés pour modéliser son fonctionnement.

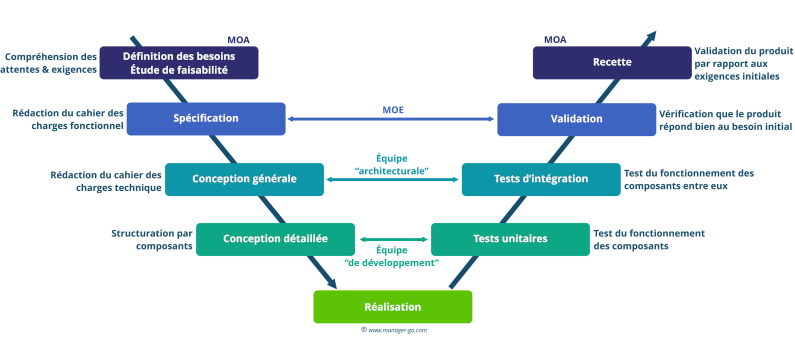
# Analyse et conception

## Cycle de vie

Le cycle **de vie d'un logiciel** (en anglais *software life cycle*), désigne toutes les étapes du développement d'un logiciel, de sa conception à sa disparition. L'objectif d'un tel découpage est de permettre de définir des jalons intermédiaires permettant la **validation** du développement logiciel, c'est-à-dire la conformité du logiciel avec les besoins exprimés, et la **vérification** du processus de développement, c'est-à-dire l'adéquation des méthodes mises en œuvre. Il y’a plusieurs modèles de cycle de vie selon le type du logiciel et les besoins exprimés par le client (la maitrise d’ouvrage).

Pour notre projet, nous avons choisi d'utiliser le modèle de cycle de vie en V en raison de sa pertinence pour notre contexte. Notre projet, bien que de grande envergure, nécessite une approche structurée et rigoureuse pour garantir la qualité du système. Le modèle en V nous permettra de suivre les différentes phases du cycle de développement de manière méthodique, en mettant l'accent sur la validation et la vérification des fonctionnalités. De plus, il nous offre la flexibilité nécessaire pour intégrer de nouvelles fonctionnalités et améliorations tout au long du processus.

Le modèle en V est un modèle de cycle de développement logiciel qui suit les différentes étapes du cycle en forme de "V". Il se caractérise par la mise en place d'un processus itératif, où chaque phase du cycle est revisitée pour proposer un produit de plus en plus abouti. Cependant, contrairement au modèle en spirale, le modèle en V met davantage l'accent sur la validation et la vérification des fonctionnalités plutôt que sur la gestion des risques.



**Figure 4 - Cycle de vie en V (source img: https://www.manager-go.com/gestion-de-projet/cycle-en-v.htm)**

# Modélisation avec le langage UML

## Langage UML

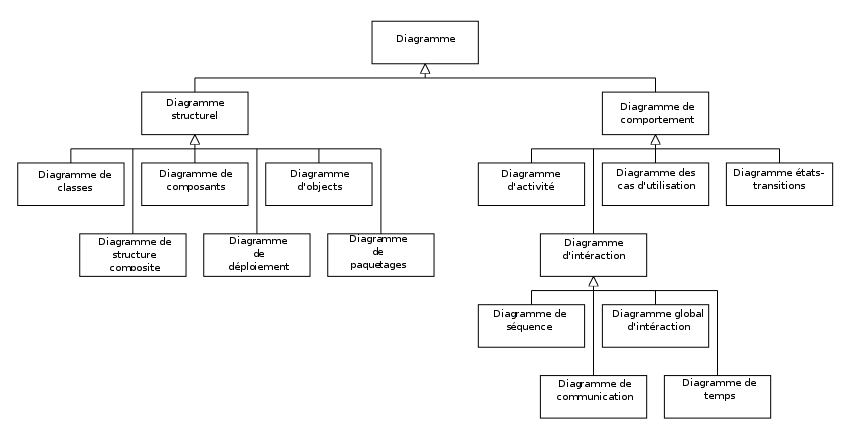
UML signifie « Unified Modeling Language » (langage de modélisation objet unifié). UML est issu de la fusion des méthodes :

• Booch : Grady Booch (1991) ;

• OMT (Object Modelling Technique): James Rumbaugh (1991) et

• OOSE (Object Oriented Software Engineering): Ivar Jacobson (1992).

UML est né dans les années 1990. On assiste à la normalisation d’UML1.0 en janvier 1997 par Object Managment Group (OMG) ; UML 2.0 a été adopté par l'OMG en juillet 2005. La dernière version de la spécification validée par l'OMG est UML 2.5.1 (2017). Le langage de modélisation UML est destiné à faciliter la conception des documents nécessaires au développement d'un logiciel, comme standard de modélisation de l'architecture logicielle. UML 2.0, propose 14 types de diagrammes dont six structurels et huit comportementaux. Les diagrammes sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie. (Source <https://fr.wikipedia.org/wiki>)



**Figure 5 - Les diagrammes d'UML 2.5**

## Identification des acteurs

***Qu’est-ce qu’un acteur*** ?

Un acteur incarne le rôle de l'utilisateur qui engage des interactions avec le système.

L’acteur peut être un utilisateur humain, une organisation, une machine ou un autre système externe.

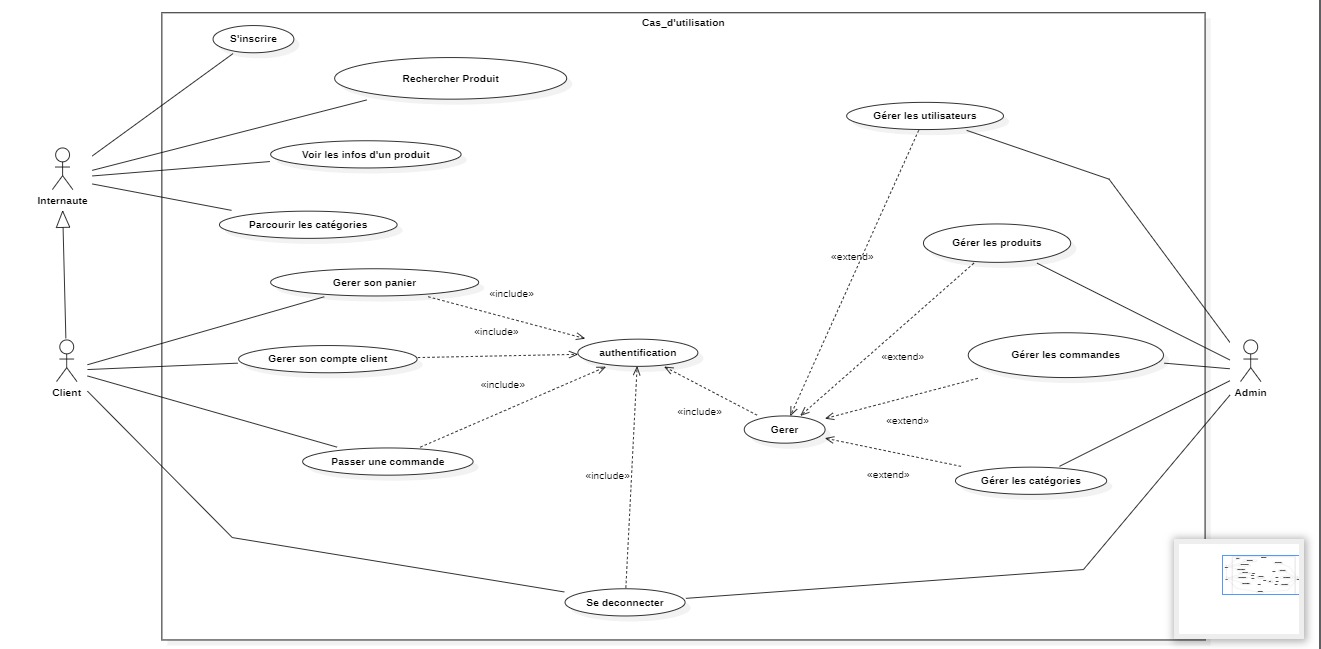
Dans notre système, les acteurs sont :

* **l'administrateur** (ou les administrateurs) joue un rôle clé. L'administrateur a le pouvoir d'effectuer toutes les opérations disponibles sur la plateforme, offrant ainsi un contrôle total sur le système. En plus de cela, l'administrateur a la capacité d'améliorer continuellement les fonctionnalités et en rectifiant d'éventuelles erreurs qui pourraient se produire.
* **Client** représente les utilisateurs finaux qui interagissent avec la plateforme. Le client peut créer un compte, naviguer à travers les produits, effectuer des achats, ajouter des articles à son panier, passer des commandes et effectuer des paiements en ligne de manière sécurisée. Le client bénéficie d'une expérience utilisateur conviviale et peut également gérer son profil, consulter l'historique de ses commandes.
* **Visiteur (internaute)** qui représente les utilisateurs qui consultent le site sans nécessairement effectuer des achats. Les visiteurs peuvent naviguer à travers les différentes catégories de produits, consulter les descriptions et les images. Ils ont la possibilité de rechercher des informations spécifiques sur les produits, d'explorer les fonctionnalités offertes par la plateforme et de se familiariser avec l'offre disponible. Bien que les visiteurs n'aient pas la capacité de passer des commandes ou d'accéder à certaines fonctionnalités réservées aux clients.

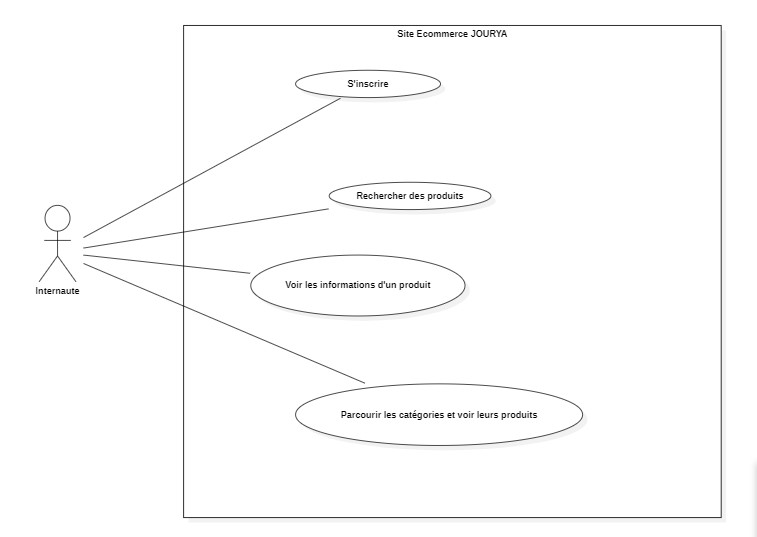
## Diagramme de cas d’utilisation

Un cas d’utilisation est un service offert par le système, à un acteur.

Le diagramme de cas d’utilisation (UseCaseD) est une représentation contextuelle de haut niveau du système modélisé. **Il permet de faire ressortir les grandes fonctionnalités du système**. C’est un des diagrammes UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Le UseCaseD est utile pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisations sont plus appropriés. En effet, un cas d'utilisation (*use case*) représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système (source <https://fr.wikipedia.org/wiki>).

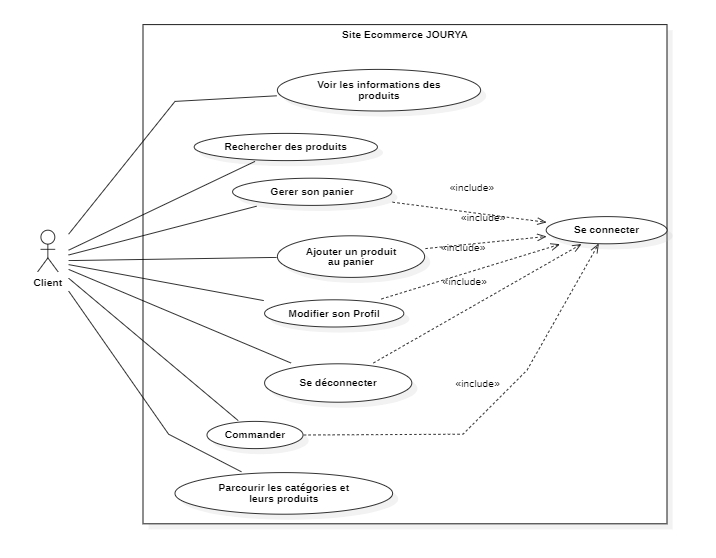
**Figure 6 - diagramme de cas d'utilisation**

***Un peu plus détails sur les CU d’un internaute, client et admin.***

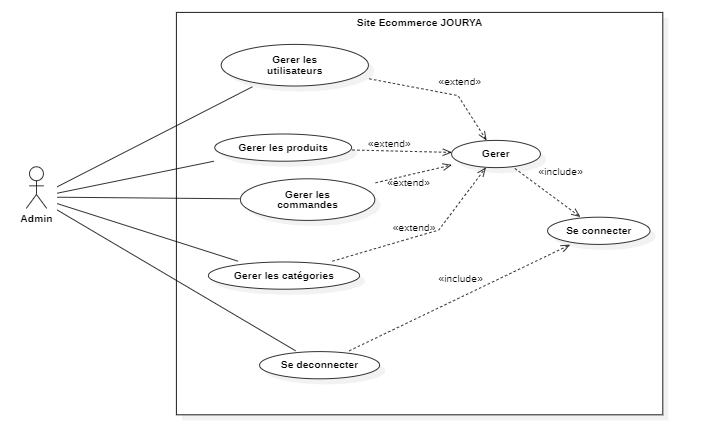


**Figure 7 - détails sur les cas d'utilisation d’un internaute**

Ce diagramme explique de façon un peu plus détaillé les cas d’utilisation d’une personne inscrite. Il met en évidence les relations d'inclusions « include » sur la figure.



**Figure 8 - détail sur les cas d'utilisation de client**

**Figure 9 - détail sur les cas d'utilisation de l’admin**

Comme pour les précédents, on retrouve ici aussi un peu plus détails sur les cas d’utilisation de l’admin. Il met en évidence les relations d’extension

« extend » et les relations d'inclusions « include » sur la figure et de généralisation entre les cas d’utilisation.

## Diagramme de séquence

Un diagramme de séquence comprend un groupe d'objets**1**, représentés par des lignes de vie**2**, et les messages que ces objets échangent lors de l'interaction.

Il permet de montrer les interactions d’objets dans le cadre d’un scénario**3** d’un diagramme de cas d’utilisation.

On distingue les messages synchrones, asynchrones et réflexive.

***Définitions*** :

• **Message :** définit une communication particulière entre des lignes de vie (objets ou acteurs)

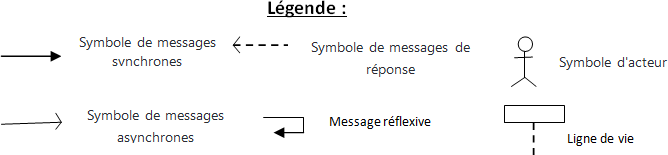
• **Message synchrone :** bloque l’expéditeur du message jusqu’à la réponse du destinataire.

• **Message asynchrone :** l’expéditeur du message ne reste pas bloqué en attendant la réponse du destinataire.

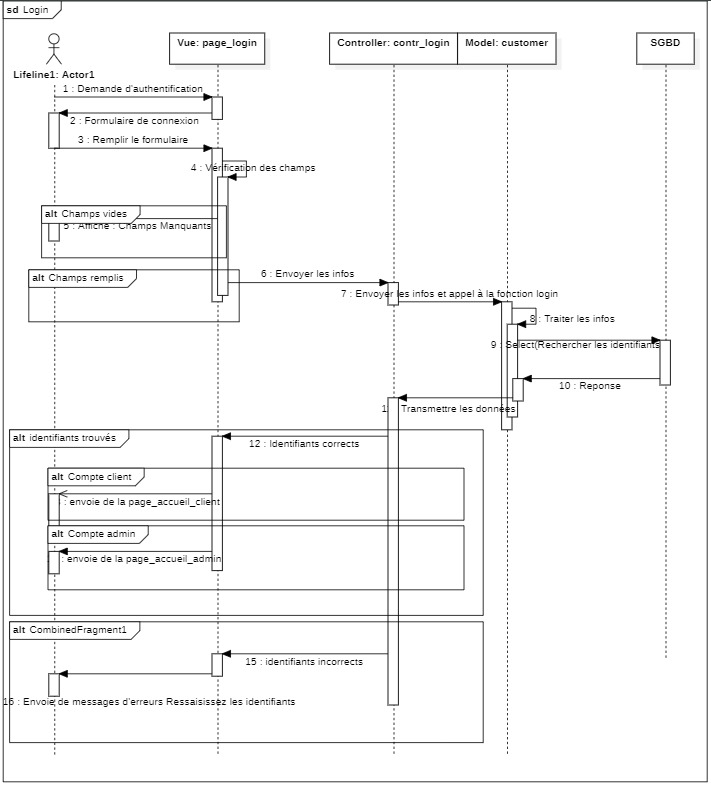
**1 Objet** : c’est une instance d’une classe.

**2 Ligne de vie** : représente un participant à une interaction (objet ou acteur)

**3 Scénario** : c’est une instance d’un cas d’utilisation

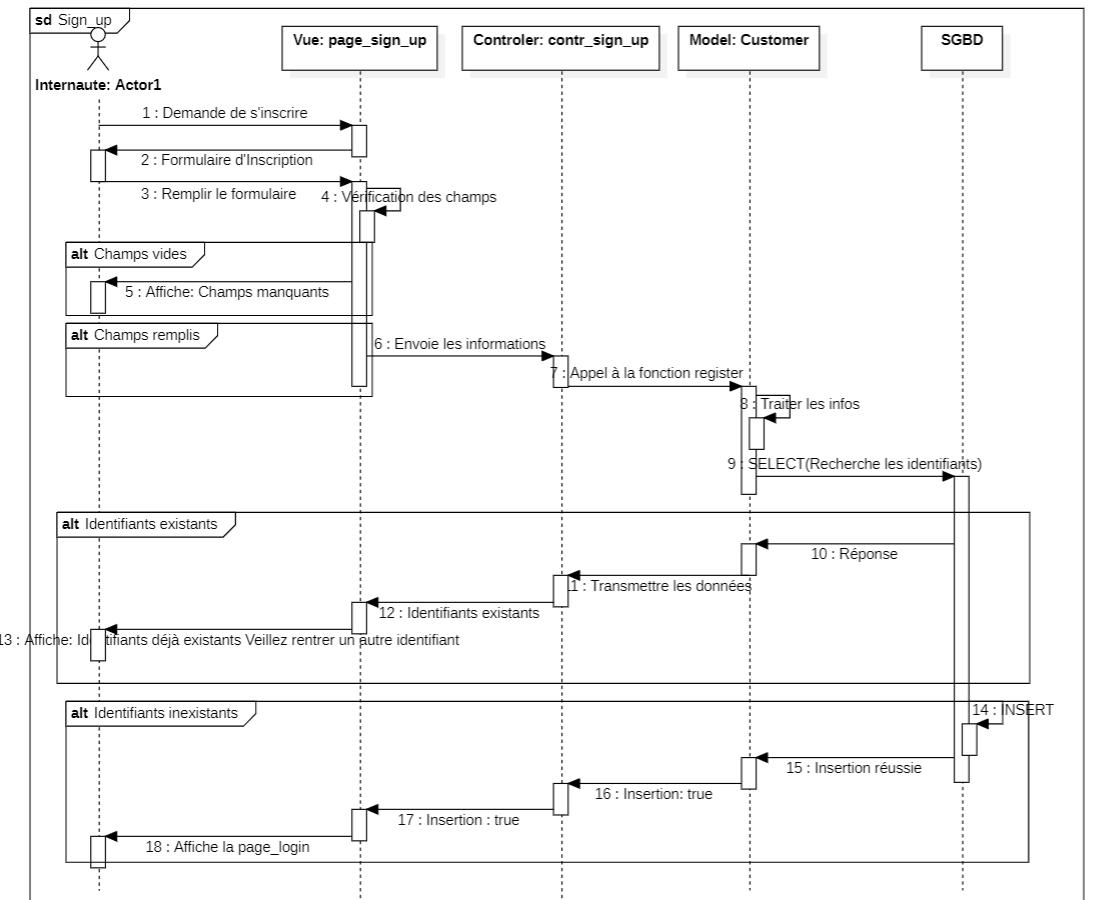
**• Message réflexive :** c’est un message d’un objet vers lui-même.

***Ci-dessous quelques diagrammes de séquences relatifs à notre système :***

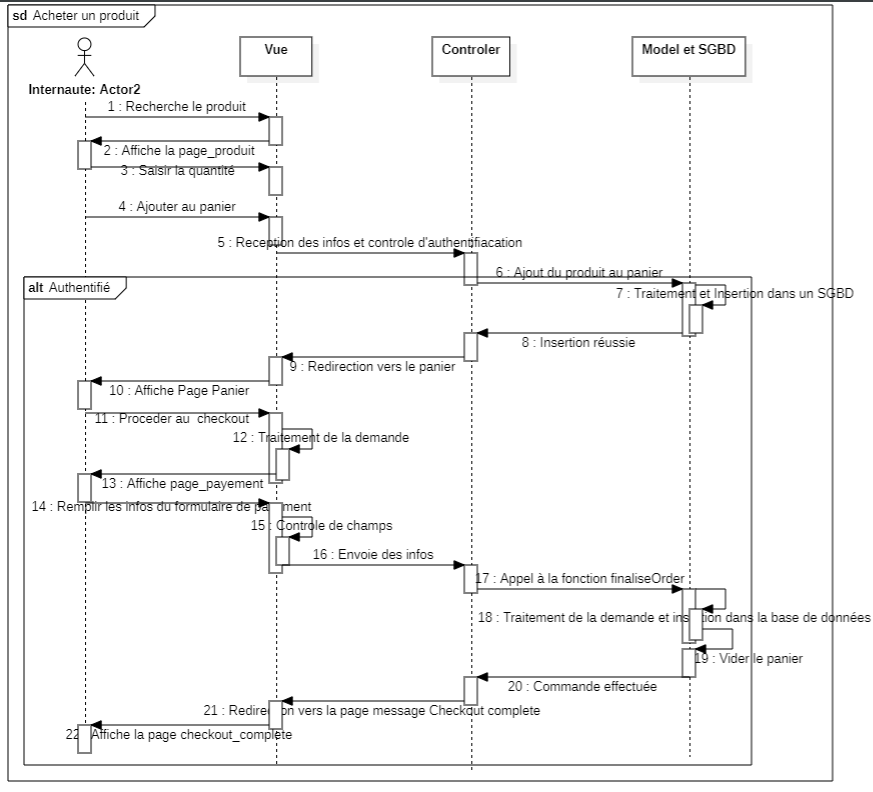


**Figure 10 - Diagramme de séquence du cas d’utilisation “se connecter ”**

Ce diagramme nous présente un scénario du cas d’utilisation “**se connecter**”. Pour se connecter, étant donné que nous travaillons avec le modèle MVC l'utilisateur interagit avec la vue où il rentre ses informations et passe au contrôle des champs. Ensuite les informations rentrées seront dirigées par le contrôleur au niveau du modèle où est appelé la fonction login qui se charge de traiter les informations et si ces derniers sont correctes, renvoie vers la page d'accueil par l'intermédiaire de la vue.

**Figure 11 - Diagramme de séquence du cas d’utilisation “s’inscrire ”**

Pour le cas d'utilisation d'inscription, après avoir interagit avec la vue en demandant et en remplissant le formulaire de la page d'inscription, le contrôle des champs du formulaire est effectué et si tous les champs sont remplis, le contrôleur transmet les informations reçues vers le modèle qui se charge de les traiter. Le modèle effectue une recherche de ces identifiants dans la base de données au cas où ces derniers n'existent pas dans la base de donnée, le modèle procède à inscrire le nouveau utilisateur dans la base de données. Si les identifiants existent déjà, un message s'affiche indiquant de rentrer d'autres identifiants différents des précédents entrés.



**Figure 12 - Diagramme de séquence du cas d’utilisation “Acheter un produit ”**

Pour acheter un produit, après avoir consulter la page du produit, parcouru les informations relatives au produit et saisi la quantité souhaitée auprès de la vue, l'utilisateur procède à l'ajout du produit à son panier, ceci est possible que quand l'utilisateur est authentifié. Après le contrôle d'authentification, si ce dernier est authentifié, les informations sont envoyées au modèle où on les traite et insère ainsi ces derniers sous forme de cart\_item dans la base de donnée. Après cela, la page du panier s'affiche et le client est invité à passer au checkout, ainsi validé le contrôleur envoie vers la vue de la page de paiement où le client rempli le formulaire du paiement. Un contrôle des champs est effectué et les informations sont envoyées par le contrôleur au niveau du modèle où il traite le paiement et l'enregistrer sous forme de commande effectuée. Une page de finalisation de la commande est affichée avec l'identifiant de cette dernière.



**Figure 13 - Diagramme de séquence du cas d’utilisation “Ajout d’un produit ”**

Pour le cas d'utilisation ajouter un produit, l'administrateur interagit avec la page d'ajout de produit fournit par la vue, remplis le formulaire par des informations qui seront dirigées vers le modèle où ces dernières seront traitées et ainsi un nouveau produit sera ajouté à la base de données.

## Diagramme de classe

***Qu’est-ce qu’une classe*** ?

Une classe représente un objet ou un ensemble d'objets qui partagent des attributs et des comportements (méthodes) communs. C’est une abstraction des choses du monde réel.

Une classe possède :

• Un nom (le nom est placé dans la première sur la figure d’une classe) ;

• Des attributs (le(s) attributs dans la deuxième partie sur la figure d’une classe) ;

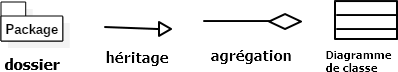
• Des méthodes (elles sont placées dans la troisième partie sur la figure d’une classe)

***Un diagramme de classe*** permet de modéliser les objets qui constituent le système, pour afficher les relations entre les objets, pour décrire ce que ces objets font et les services qu'ils fournissent.

C’est le diagramme le plus importance et indispensable dans la modélisation UML d’un système réalisé avec le concept Orienté Objet.

Il est représenté par un rectangle divisé en trois (3) parties la première contient le nom de la classe, la seconde la liste des attribues avec leurs types et visibilité (public +, privée -, protégée # ou package dans le cas où rien n’est mentionné) et la troisième partie contient les méthodes (ou Opérations).

Les diagrammes qui suivront vont mettre en évidence certaines relations qui peuvent exister entre les différentes classes d’un diagramme de classes telles que l’héritage**4**, l’agrégation et la notion de package l’équivalent des dossiers dans la programmation (java par exemple)

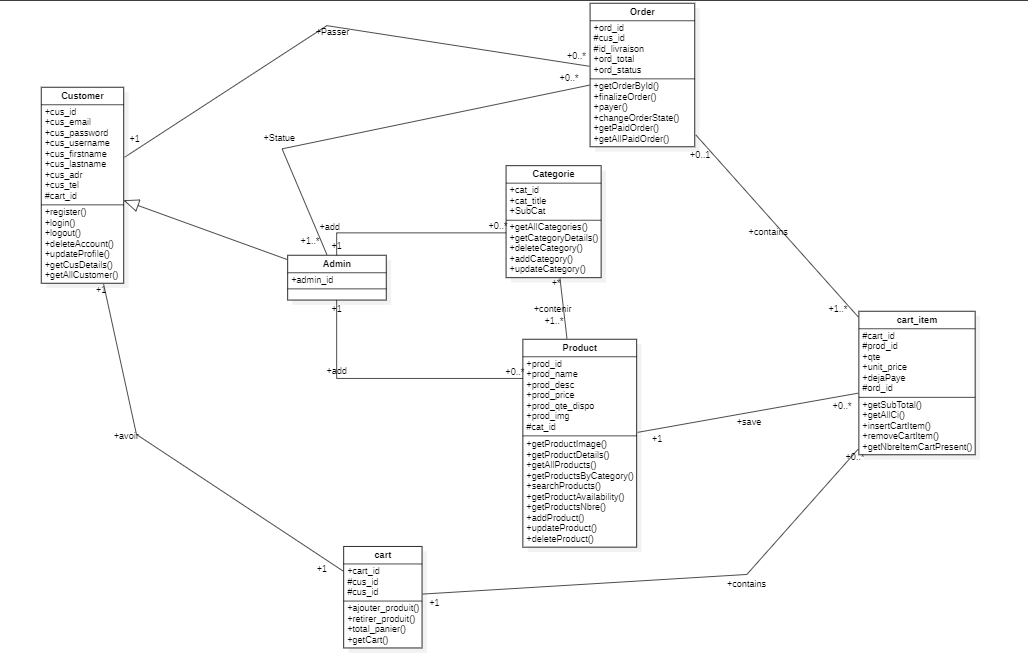


***Légende***

**4 L'héritage :** permet de factoriser les éléments (attributs et méthodes) qui sont communs à un ensemble de classe et de les mettre dans une autre classe plus générale appelée classe mère à partir de laquelle on peut avoir d'autre classe plus spécialisée (Classe fille ou classe dérivée).

**Figure 14 - Le diagramme regroupant les classes et les relations entre elles**

Notre diagramme de classe est constitué de differentes classes en occurence les classes Customer qui représente les clients, Order qui représente less Commandes, admin, Product qui représente les produits, Catégorie qui représente les catégories et Cart item qui représente les élements ajoutés au panier. Ces differentes classes sont composées de divers attributs et de methodes propres à chacune. En exemple la classe Product est composée d'attribut telle que prod\_id qui est son identifiant, prod\_name, prod\_price... et des méthodes telles que getProductImage, getProductDetails qui sont des actions que l'on peut effectuer sur chaque instance.

******Par ailleurs, la classe Customer intéragit avec la classe order où il spécifie les commandes que le client a passé et le client est lié à un seul panier c'est ce qui est notifié dans le diagramme par la relation entre la classe customer et cart.

On a la classe cart\_item qui fait la passerrelle entre la classe cart (le panier auquel l'élement du panier appartient) et la classe Product signifiant qu'un produit est ajouté avec des attributs telle que la quantité à notre panier.

La relation entre le cart\_item et le Order signifie que la commande à la fin sera composée des éléments de panier (cart\_item) qui seront validés par le client au niveau du checkout.

# Conclusion

En résumé, cette section de conception a couvert tous les aspects de la conception de notre projet. Elle a présenté les différents diagrammes qui ont permis de définir la structure globale du système, de visualiser les classes et leurs relations, d'identifier les fonctionnalités clés et les acteurs impliqués, ainsi que de décrire les interactions entre le système et ces acteurs. La phase de conception a joué un rôle essentiel en fournissant une vision claire du système et en permettant aux parties prenantes de se mettre d'accord sur les spécifications et les objectifs du projet. Les diagrammes ont servi de référence précieuse pour guider la mise en œuvre ultérieure du système, en assurant une base solide et une compréhension commune entre toutes les parties concerné

CHAPITRE III : REALISATION DU PROJET

# Introduction

Nous présenterons dans cette partie la description des outils et environnements nécessaires à la réalisation de notre projet. Nous aborderons divers éléments tels que les langages informatiques, les scripts, les bibliothèques, ainsi que les logiciels indispensables tels que les éditeurs de texte, les logiciels de gestion de projets et les logiciels de présentation.

L'objectif est de mettre en évidence les ressources utilisées qui sont soit libres de droits, soit disponibles en version d'essai, ou encore proposant des alternatives libres, le cas échéant. Cette approche garantit à la fois la conformité aux réglementations et l'accessibilité des outils nécessaires à la réalisation de notre projet.

# Langages et outils de développement

## Langages et scripts

#### HTML5

Le terme HTML est l'acronyme de "HyperText Markup Language", qui se traduit par "langage de balisage pour l'hypertexte". Il est utilisé pour la création et la représentation du contenu ainsi que de la structure d'une page web. La dernière révision majeure de HTML est le HTML5, qui a été finalisé le 28 octobre 2014. Cette version apporte de nombreuses nouveautés telles que les formats de date et d'email, ainsi que l'introduction des balises footer, header, section, etc. Cette évolution du HTML offre des fonctionnalités et des possibilités étendues pour la conception et le développement de sites web modernes.

#### CSS3

Les feuilles de style en cascade, également connues sous le nom de CSS (Cascading Style Sheets) en anglais, sont un langage informatique utilisé pour décrire la présentation des documents HTML. Elles permettent d'ajouter des couleurs, d'améliorer la mise en forme et même d'incorporer de petites animations aux pages web. Les CSS jouent un rôle essentiel dans l'esthétique des sites web en offrant un contrôle précis sur des aspects tels que la mise en page, les polices de caractères, les couleurs et les bordures. Grâce aux CSS, il est possible de créer des designs attrayants et cohérents sur l'ensemble d'un site, tout en simplifiant la gestion des styles

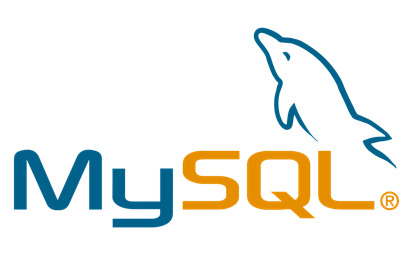
#### JavaScript

JavaScript est un langage de script largement utilisé, initialement conçu pour rendre les pages web plus interactives. Cependant, sa popularité s'est rapidement accrue grâce à de nombreuses bibliothèques telles que Node.js, jQuery et React. Outre son utilisation côté client, le langage JavaScript peut également être utilisé côté serveur, ce qui lui confère une polyvalence et une adaptabilité remarquables. Il offre un large éventail de fonctionnalités et permet de créer des applications web dynamiques et performantes.

#### PHP

Le PHP est un langage de programmation utilisé pour créer des pages web dynamiques. C'est un langage côté serveur, ce qui signifie qu'il permet d'exécuter des scripts sur le serveur avant de renvoyer le contenu au client. Il est convivial, flexible et distribué sous licence libre. Grâce au PHP, il est possible de développer des sites web interactifs en adoptant une approche impérative et/ou orientée objet.

#### MySQL

MSQL est un Système de Gestion de Base de Données (SGBD) parmi les plus populaires au monde. Il est distribué sous double licence, une *licence publique générale GNU* et une *propriétaire* selon l’utilisation qui en est faites. La première version de MySQL est apparue en 1995 et l’outil est régulièrement

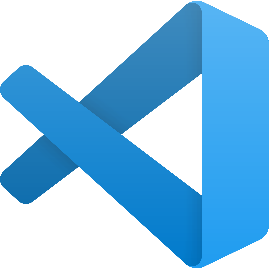
entretenu. SQL fait référence au Structured Query Language, le langage de requête utilisé.

#### SQL

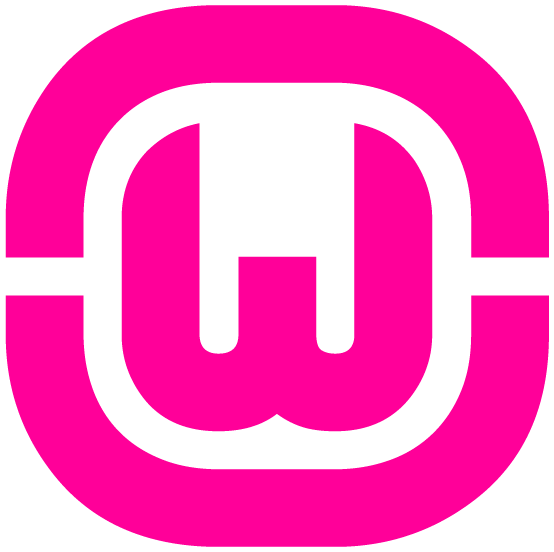
Le **SQL** (Structured Query Language) est un langage permettant de communiquer avec une base de données. Ce langage informatique est notamment très utilisé par les développeurs web pour communiquer avec les données d’un site web.

## Outils de développement

#### Visual studio code :

Visual Studio Code est un éditeur de code source polyvalent utilisable avec divers langages de programmation tels que Java, JavaScript, Go, Node.js et C++. Il est construit sur le framework Electron, qui est largement utilisé pour le développement d'applications Web.

#### WampServer :

  
Le Wamp est un ensemble de logiciels comprenant Apache, le serveur HTTP, MySQL, le système de gestion de base de données, et PHP, l'interpréteur PHP. Ces composants permettent d'exécuter localement des scripts PHP sans avoir besoin de se connecter à un serveur externe.

#### StarUML

StarUML est un outil spécialisé dans la modélisation UML, largement utilisé dans le développement logiciel et la conception orientée objet. La modélisation UML repose sur des diagrammes et des pictogrammes, offrant une représentation visuelle de l'architecture d'un projet en mettant en évidence les acteurs, les processus et les composants. Ce logiciel permet la conception de classes, d'objets et d'acteurs, ainsi que la définition de nombreux attributs. Bien que StarUML soit un logiciel payant, il existe une alternative gratuite et open source appelée UNLet.

#### Microsoft Word, PowerPoint, Project

Word est un logiciel de traitement de texte couramment utilisé pour créer des documents, tandis que PowerPoint est un logiciel dédié à la réalisation de présentations avec un système de diapositives. Ces deux logiciels font partie de la suite Microsoft Office et sont largement utilisés dans les environnements bureautiques. Il est important de noter qu'ils ne sont pas destinés à la programmation. Des alternatives libres telles que la suite OpenOffice offrent des logiciels équivalents.

Quant à Microsoft Project, également connu sous le nom de MS Project, il s'agit d'un logiciel de gestion de projets édité par Microsoft. Il offre aux chefs de projet et aux planificateurs des fonctionnalités telles que la planification, le suivi des projets, la gestion des ressources et du budget, ainsi que l'analyse et la communication des données du projet.

#### PicPick

Picpik est un logiciel de capture d’écran très complet. Il contient un outil complet de capture d'écran, un éditeur d'image intuitif, une pipette de prélèvement des couleurs, une règle à pixels, un rapporteur, un curseur en croix, un tableau blanc.

#### Git

 Git est le système de contrôle de version le plus largement utilisé aujourd'hui. Git est un projet open source avancé, qui est activement maintenu. À l'origine, il a été développé en 2005 par Linus Torvalds, le créateur bien connu du noyau du système d'exploitation Linux. De plus en plus de projets logiciels reposent sur Git pour le contrôle de version, y compris des projets commerciaux et en open source. Les développeurs qui travaillent avec Git sont bien représentés dans le pool de talents disponible, et la solution fonctionne bien sur une vaste gamme de systèmes d'exploitation et d'environnements de développement intégrés (IDE).

#### GitHub

 GitHub est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Ce site est développé en Ruby on Rails et Erlang. GitHub propose des comptes professionnels payants, ainsi que des comptes gratuits pour les projets de logiciels libres.

Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches, et un wiki pour chaque projet. Le site est devenu le plus important dépôt de code au monde, utilisé comme dépôt public de projets libres ou dépôt privé d'entreprises. En 2018, GitHub est acquis par Microsoft.

## Frameworks

****Bootstrap est un framework front-end largement utilisé dans le développement de sites web réactifs et esthétiques. Il offre une gamme d'outils prêts à l'emploi, tels que des classes CSS et des composants JavaScript, qui facilitent la création d'interfaces utilisateur attrayantes et adaptées à différents appareils. Grâce à sa structure flexible et à sa vaste bibliothèque de styles préconçus, Bootstrap permet aux développeurs de gagner du temps en réutilisant des éléments de conception et en appliquant facilement des mises en page responsives. Que ce soit pour créer des sites web simples ou des applications web plus complexes, Bootstrap offre une base solide pour un développement web efficace et est largement adopté par la communauté des développeurs.

## Architecture de l’application web

Dans notre projet, nous avons utilisé le modèle **MVC (Modèle-Vue-Contrôleur).** Le MVC est un design pattern d'architecture logicielle largement utilisé pour concevoir et développer des applications. Il offre une structure organisée et facilite la séparation des préoccupations en divisant l'application en trois composants principaux : le modèle, la vue et le contrôleur.

**Le Modèle (Model)** : Le modèle représente les données qu’on aura à afficher. Il est donc responsable de la manipulation des données, de leur stockage et de leur récupération. Le modèle encapsule également la logique qui permet de traiter les données et de maintenir leur intégrité. Cela peut inclure des opérations telles que la validation des données, les calculs et les mises à jour.

**La Vue (View)** : La vue est responsable de l'affichage des données au sein de l'application. Elle présente les informations à l’utilisateur de manière claire. La vue peut être une interface utilisateur graphique, une page web en soi, tout moyen de représentation visuelle des données. Dans le contexte du modèle MVC, la vue est passive et n'effectue pratiquement aucune opération. Elle se contente de présenter les données fournies par le modèle et de répondre aux interactions de l'utilisateur.

**Le Contrôleur (Controller)** : Le contrôleur gère les interactions de l'utilisateur avec l'application. Il reçoit les entrées de l'utilisateur, les interprète et déclenche les actions appropriées. *Le contrôleur agit comme un intermédiaire entre la vue et le modèle*. Lorsqu'un utilisateur interagit avec la vue, le contrôleur met à jour le modèle en conséquence et rafraîchit la vue pour afficher les changements. Il assure donc la coordination entre les différents composants de l’application.

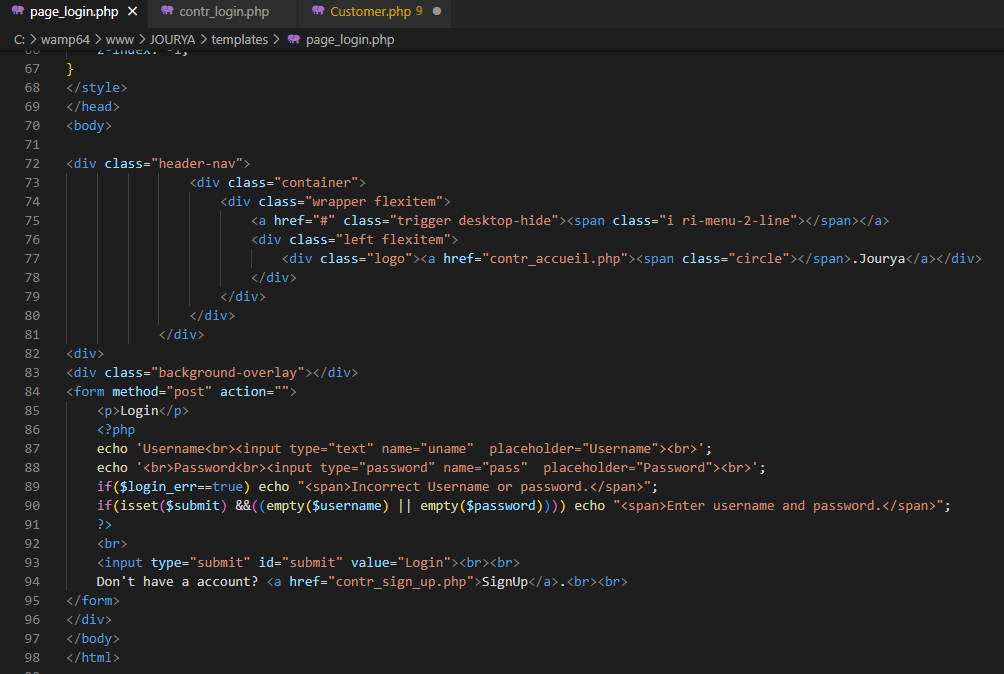
Notre choix de cette architecture est basé sur les divers avantages que le MVC propose, notamment :

* La séparation claire des responsabilités, ce qui facilite la maintenance surtout et l'évolutivité de l'application.
* La réutilisation du code, car les différents composants peuvent être développés indépendamment les uns des autres.
* La facilité de testabilité, car les différents composants peuvent être testés séparément pour vérifier leur bon fonctionnement.

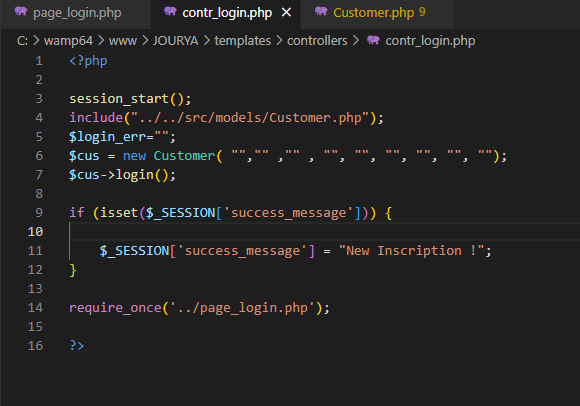


En illustration, nous avons la séquence de connexion qui est divisé en 3 parties :

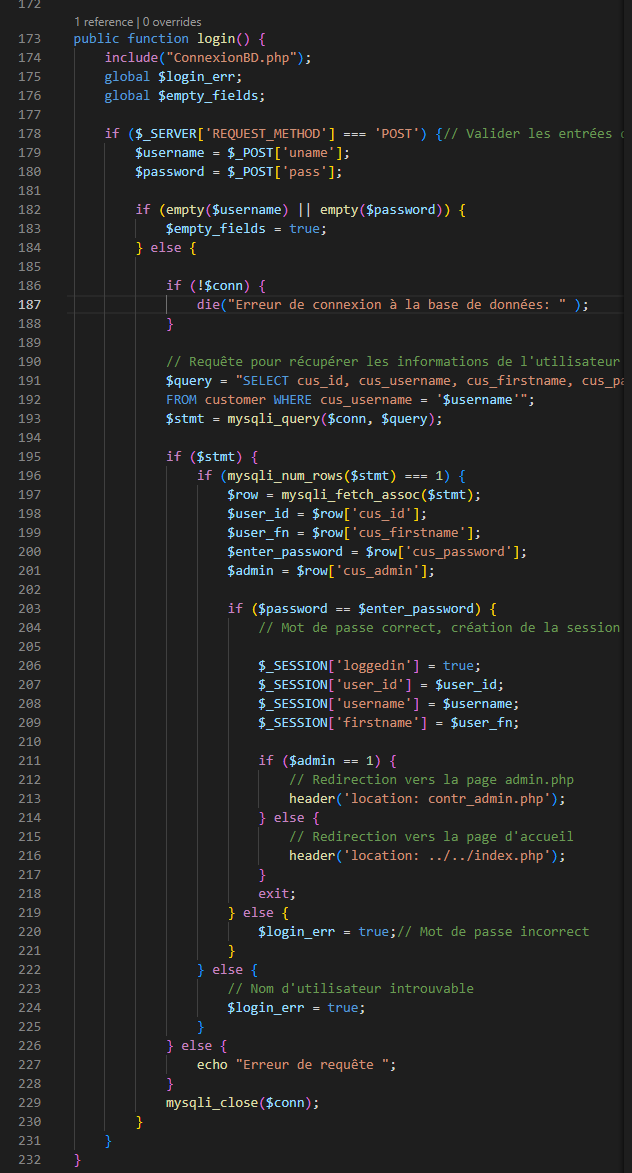
La Vue : nommée (Page\_login) qui désigne la page qui affiche l’interface visuelle de connexion à l’utilisateur en occurrence un formulaire d’entrée du nom d’utilisateur et du mot de passe. Elle est principalement composée de code html, css et javascipt.

La vue : la page qui 

Le contrôleur : nommé ici (contr\_login) est la page intermédiaire entre la vue et le modèle. Elle collecte les informations entrées par l’utilisateur au niveau de la vue et les fournies au Modèle auquel il fait appel afin de traiter ces informations.



Le Modèle : nommé ici (Customer). Le modèle Customer appelé par le contrôleur en particulier est une classe vue que nous utilisons l’approche orientée objet. Le contrôleur après instanciation de cette classe fait appel à la méthode login de la classe afin de traiter les informations reçues et de procéder à la connexion.



## Approche de Programmation : POO

L'architecture MVC offre une approche structurée pour la conception de applications. Cependant, pour tirer pleinement parti de cette architecture, il est essentiel de choisir une méthode de programmation appropriée. Dans le cadre de notre projet de fin d'études, nous avons utilisé la Programmation Orientée Objet (POO) pour développer et implémenter l'architecture MVC de notre application

La Programmation Orientée Objet (POO) est un paradigme de programmation qui permet de structurer et d'organiser le code en utilisant des objets, qui sont des instances de **classes**. La POO encourage la modélisation des concepts du monde réel en utilisant des objets dotés de propriétés (attributs) et de comportements (méthodes). Elle se base sur quatre principes fondamentaux : l'encapsulation, l'héritage, le polymorphisme et l'abstraction.

Dans le contexte de l'architecture MVC, la POO s’implante très facilement. En effet pour tirer parti de cette approche, nous avons développé le côté Modèle sous forme de classes dont les méthodes (fonctions) pourront être appelées par le contrôleur pour le traitement afin de fournir à la vue les données nécessaires pour l’affichage de la page finale.

L'architecture MVC et la POO partagent des concepts communs tels que l'encapsulation et l'abstraction. L'encapsulation permet de cacher les détails internes des composants et de les rendre plus modulaires, tandis que l'abstraction permet de représenter les composants de manière générale, sans se soucier des détails spécifiques de leur implémentation. Ces concepts facilitent la maintenance, l'évolutivité et la réutilisabilité du code.

De plus, la POO favorise la création de classes réutilisables et extensibles, ce qui s'aligne parfaitement avec les avantages de l'architecture MVC. Par exemple, on peut créer des sous-classes spécialisées pour étendre les fonctionnalités du modèle, de la vue ou du contrôleur, sans affecter les autres composants. Cela permet de développer des applications bien structurées, flexibles et évolutives.

En résumé, l'utilisation de la POO dans le cadre de l'architecture MVC de mon projet de fin d'études a permis de structurer efficacement le code et de simplifier les interactions entre les différents composants. La POO a facilité la modélisation des concepts métier, la réutilisation du code et la maintenabilité de l'application. En intégrant ces deux concepts, On a pu développer une architecture solide et flexible qui répond aux besoins de notre projet.

# Maquettes de l’application

Dans cette partie, nous présentons notre travail en montrant des captures d’écran des interfaces

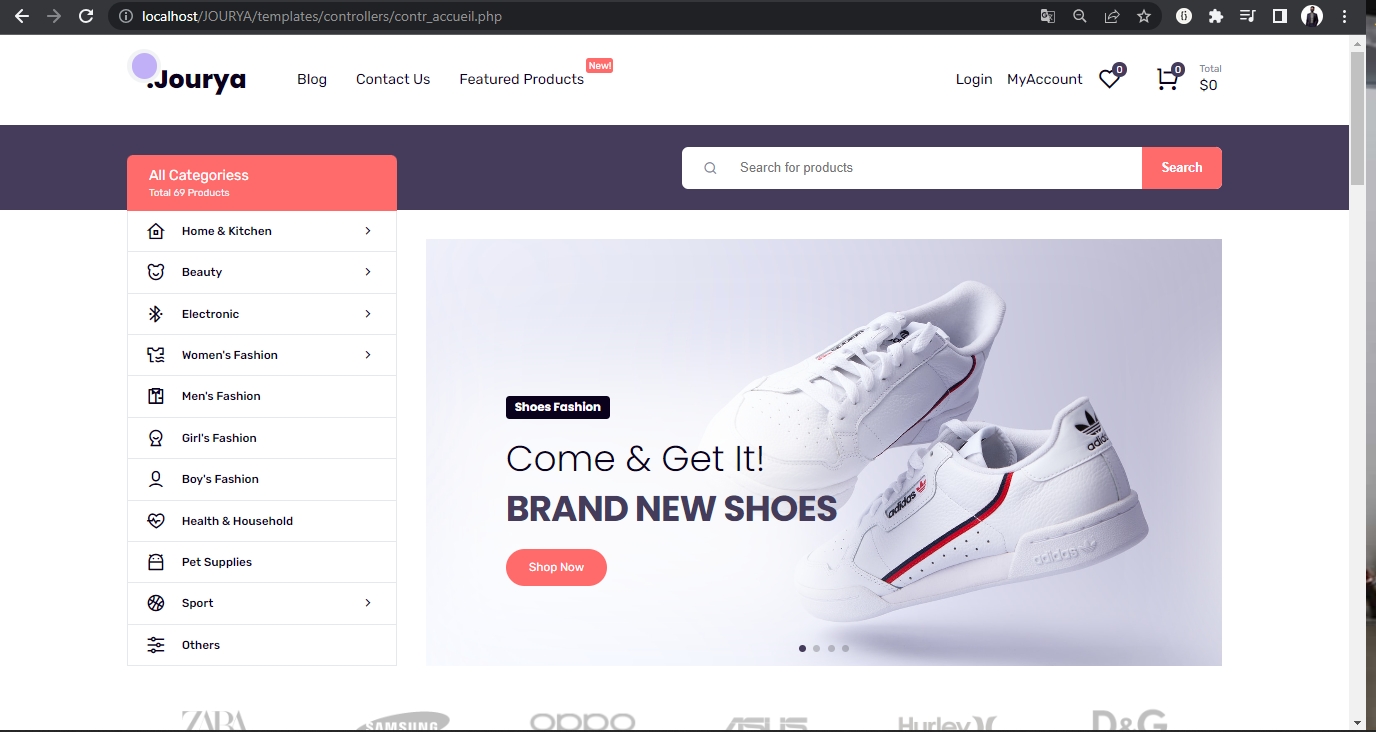
les plus importantes de notre site.

## Interface d’accueil d’un Internaute

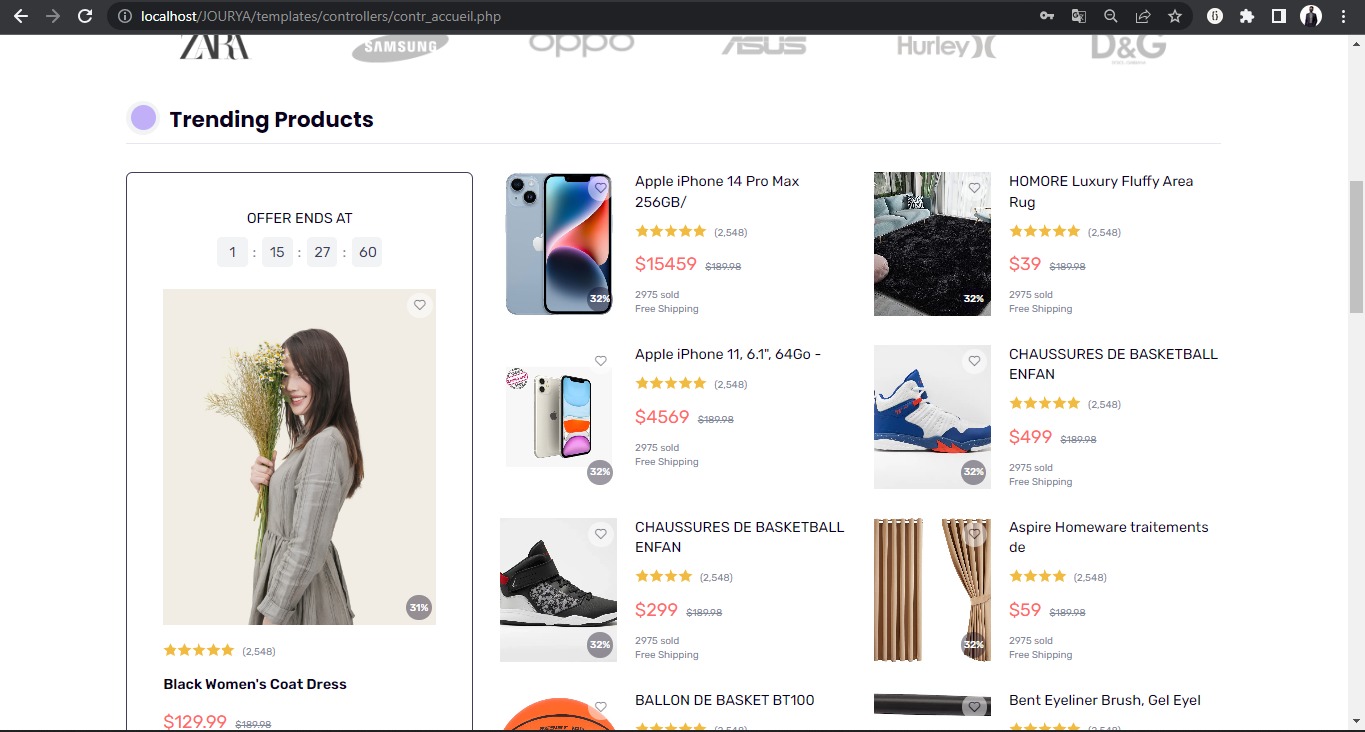
L'interface de la page d'accueil de votre site e-commerce est composée d'un menu principal clairement visible pour faciliter la navigation. Un module de recherche permet aux utilisateurs de trouver des produits spécifiques en utilisant des mots-clés pertinents. Un menu des catégories leur permet également de filtrer leur recherche par type de produit. Enfin, un slider accrocheur met en avant des messages clés tels que les promotions, les offres spéciales et les lancements de nouveaux produits.

Haut du formulaire

Regenerate responseBas du formulaire

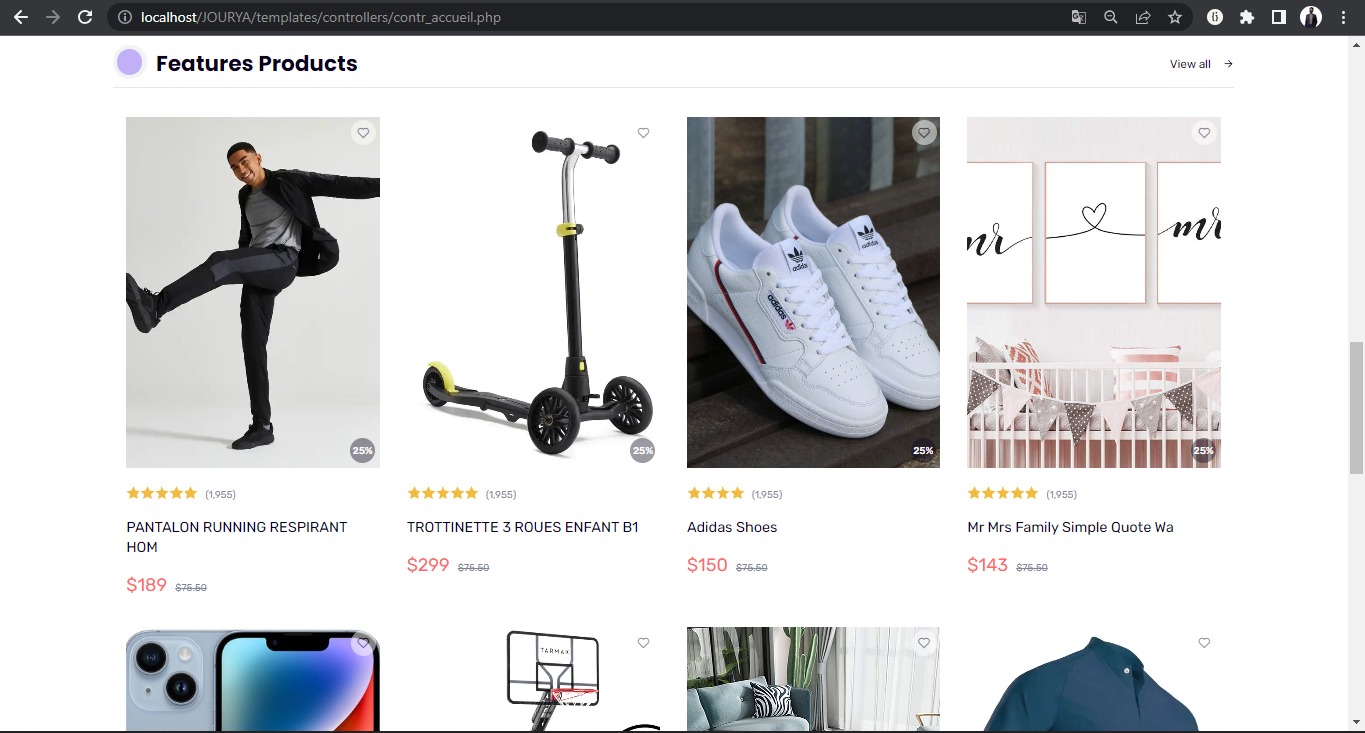


**Figure 15 - Interface d’accueil - vue1**

En explorant la page d'accueil, les utilisateurs auront accès à une sélection de produits tendance, reflétant les dernières tendances du marché. De plus, une offre spéciale à durée limitée sera mise en avant, offrant

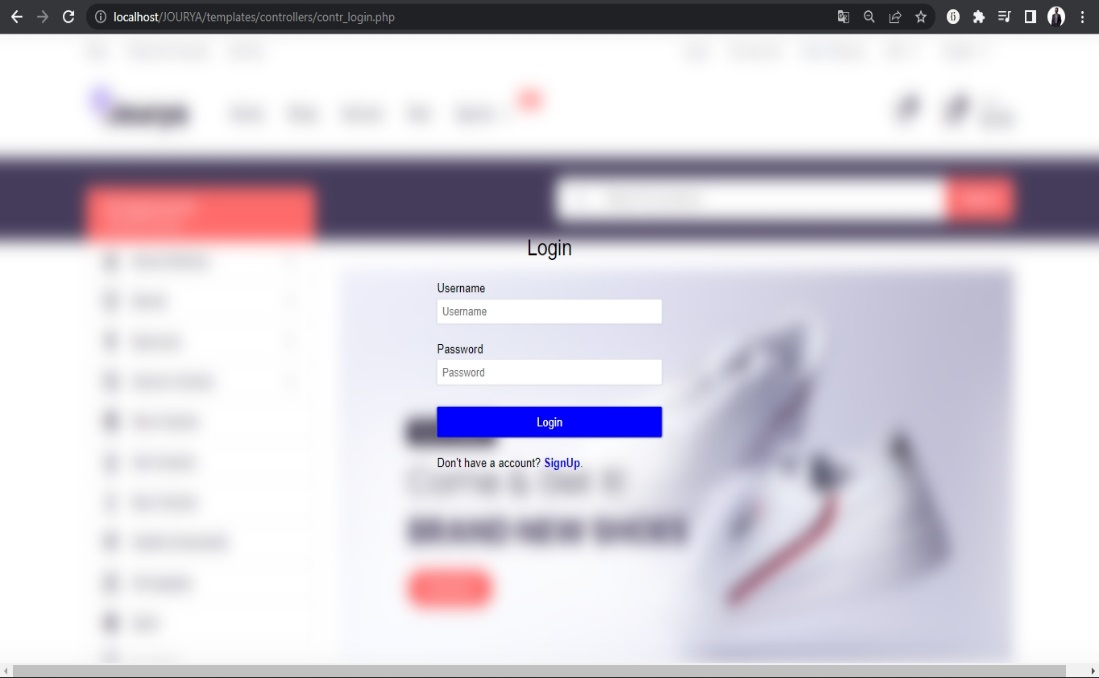
**Figure 16 - Interface d’accueil - vue2**

des avantages exclusifs. En outre, une section dédiée présentera des produits populaires, permettant aux utilisateurs d'avoir un aperçu des articles les plus appréciés par les clients.

**Figure 17- Interface d’accueil - vue3**

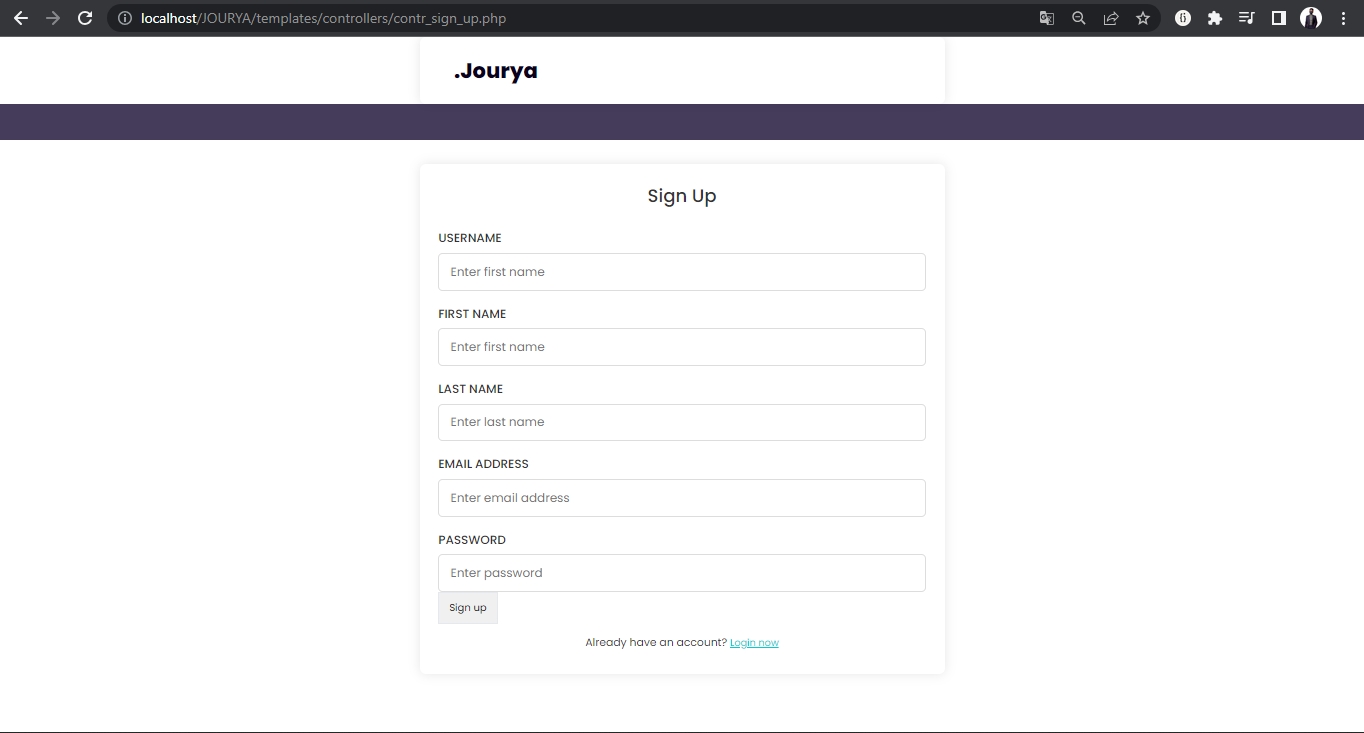
## Interface Login

Lorsque l’utilisateur clique sur le bouton “Login” le bloc suivant s’affiche :



**Figure 18 - Interface login**

L'interface permet aux utilisateurs de se connecter à leur compte existant en utilisant leurs informations valides. En cas d'erreur dans les informations saisies, un message d'erreur est affiché pour informer l'utilisateur de l'erreur. Pour les utilisateurs qui n'ont pas de compte, ils ont la possibilité de s'inscrire en cliquant sur le lien d'inscription disponible.



**Figure 19 - Interface sign up**

## Interface d’acceuil d’un client

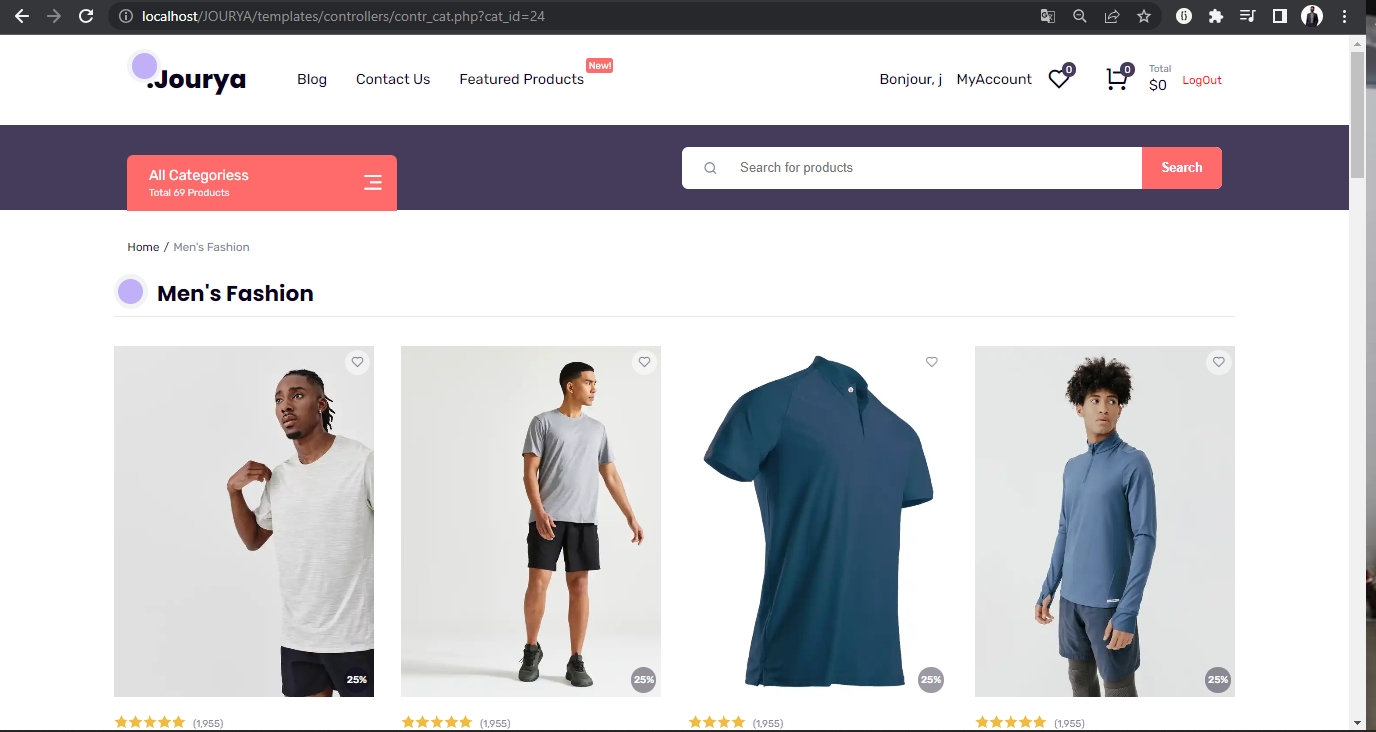
Une fois que le client a correctement saisi ses informations dans l'interface de connexion, il peut accéder à son espace personnel, tel qu'illustré ci-dessous.

**Figure 20 - Interface d'acceuil Client**

Cette interface contient un menu principal qui se compose d'un menu principal qui permet aux utilisateurs de naviguer facilement entre les différentes catégories de produits disponibles. Un module de recherche est également présent en haut de la page, offrant aux utilisateurs la possibilité de trouver rapidement des produits spécifiques.

## Interface d’une catégorie

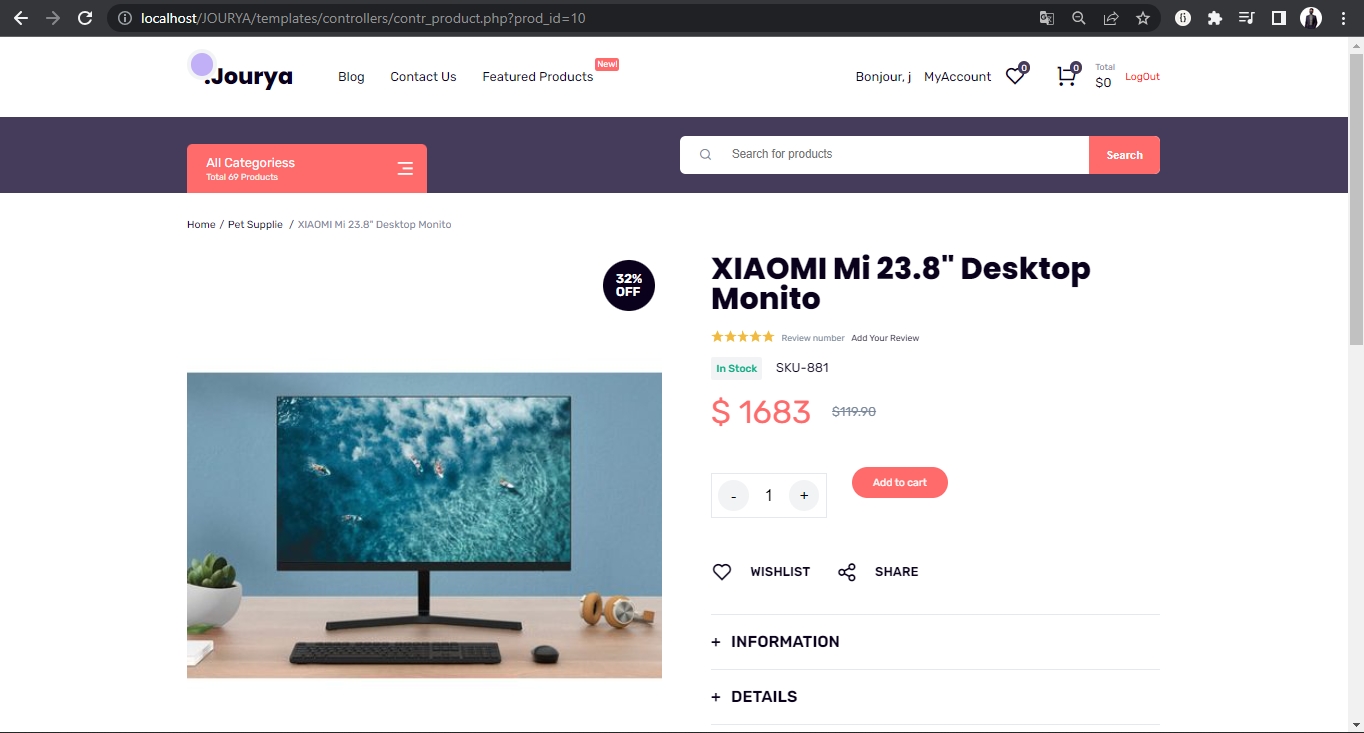
L'interface de la catégorie des produits présente une sélection variée d'articles correspondant à cette catégorie spécifique. Chaque produit est accompagné d'une description, d'une image et d'informations telles que le prix, la disponibilité et les avis des clients.



**Figure 21 - Interface d'une catégorie**

## Interface d’un produit

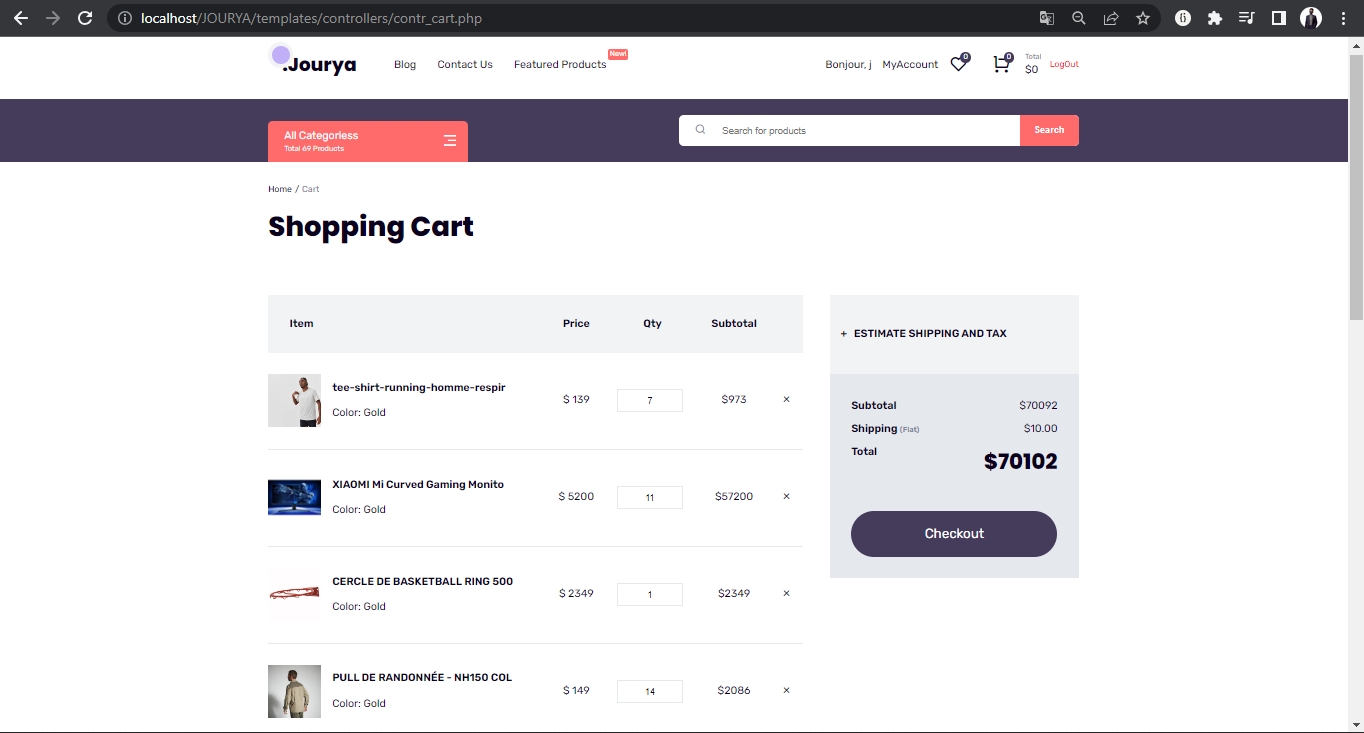
La page du produit sélectionné présente une description détaillée, des avis des clients, le prix et une photo. Cela permet aux utilisateurs d'obtenir toutes les informations essentielles sur le produit avant de prendre une décision d'achat ou de l'ajouter à leur liste de souhaits.



**Figure 22 - Interface d'un produit**

## Interface Panier

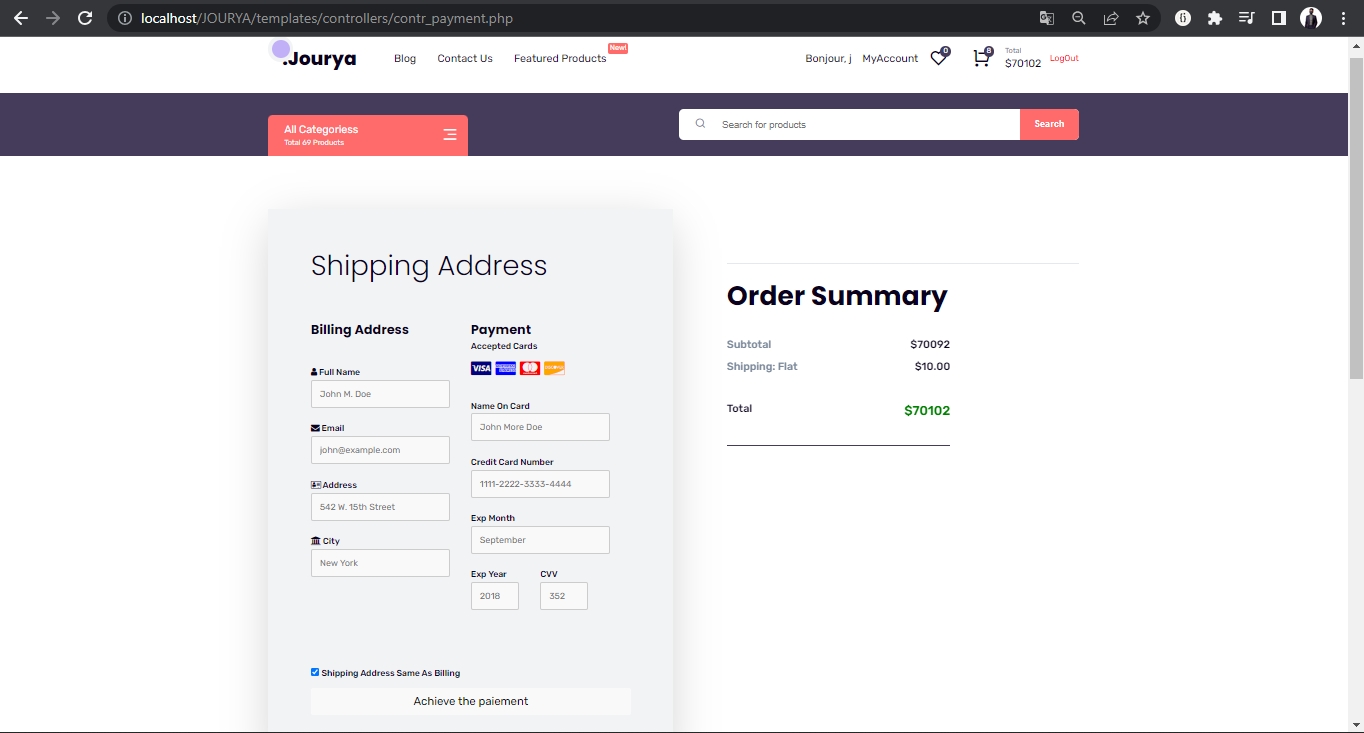
Lorsque l'utilisateur clique sur le panier, la page affiche tous les produits sélectionnés avec leur quantité respective, leur prix et la possibilité pour l'utilisateur de retirer des produits selon ses préférences. De plus, la page calcule automatiquement la somme totale de tous les produits présents dans le panier, offrant ainsi une vue d'ensemble du montant à payer.



**Figure 23 - Interface Panier**

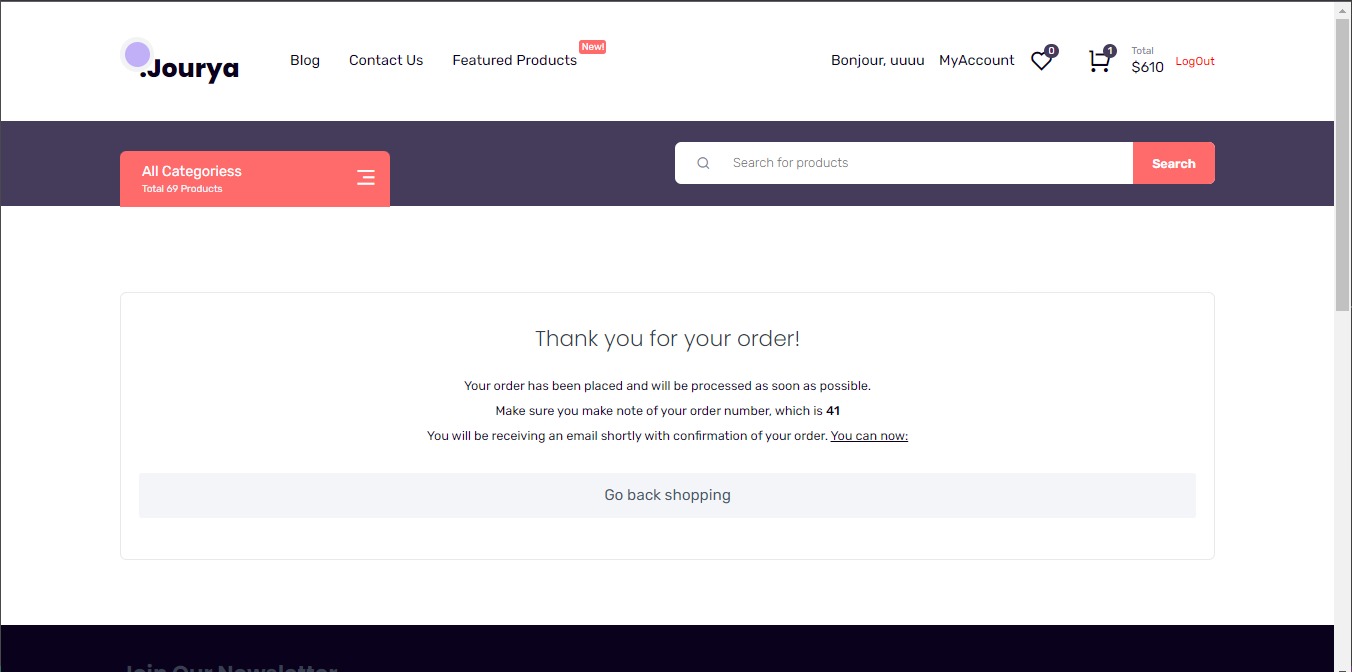
## Interface du payement

La page de paiement est l'étape suivante après avoir cliqué sur "Checkout". Elle permet à l'utilisateur de finaliser son achat en fournissant les informations nécessaires pour le paiement.



**Figure 24 - Interface payement**

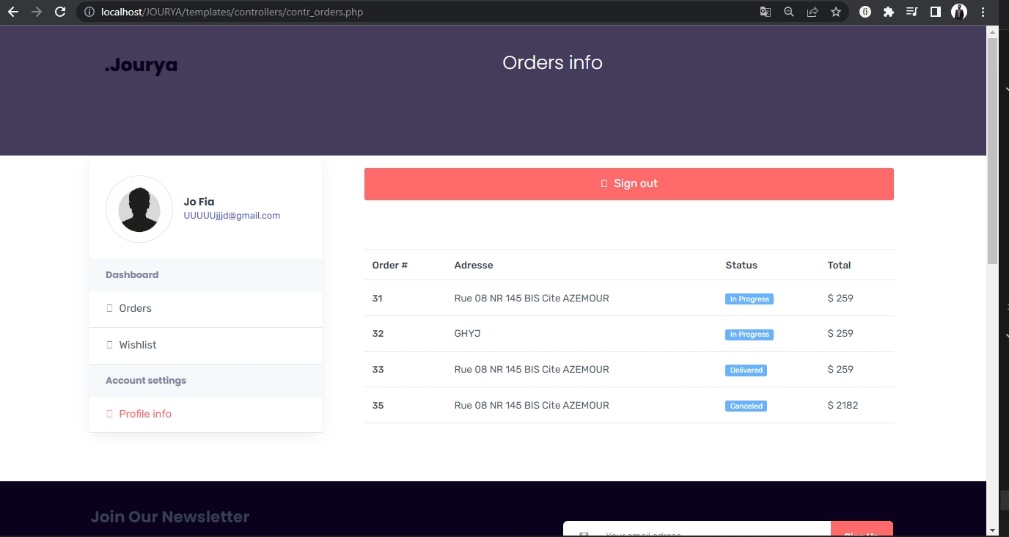
Une fois que l'utilisateur a saisi toutes les informations nécessaires, une interface s'affiche pour confirmer que le paiement a été validé avec succès. À ce stade, l'utilisateur a le choix de retourner à la page d'accueil pour poursuivre ses achats.



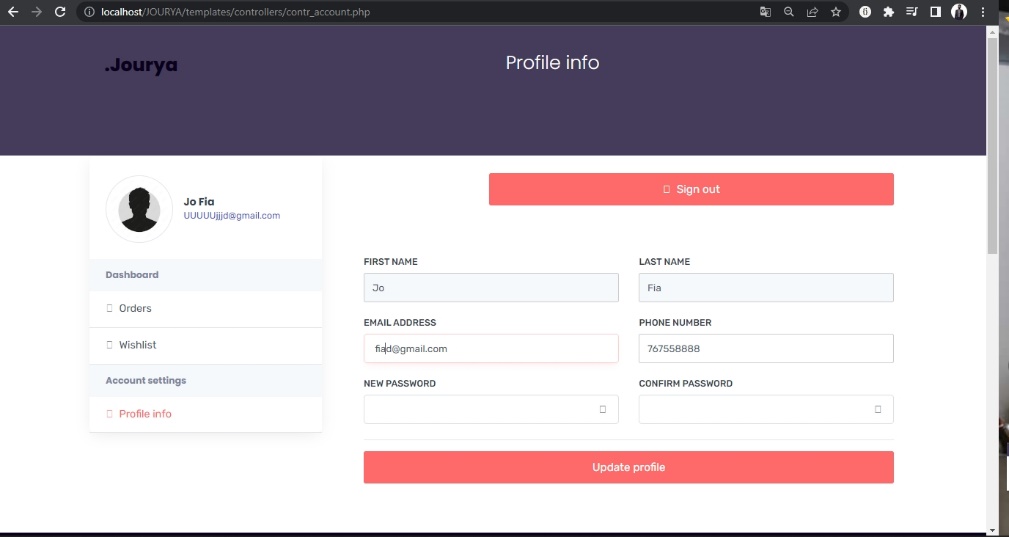
**Figure 25 - Interface payement complet**

## Interface mon compte d’un client

La page "Mon compte" offre aux utilisateurs la possibilité de consulter tous leurs commandes passées et de les suivre en temps réel. Chaque commande est accompagné d'un statut indiquant s'il a été livré, s'il est en cours de traitement ou s'il est en attente. Enfin, ils peuvent également consulter la page "Informations du profil" pour mettre à jour leurs données personnelles et s'assurer que leurs informations sont à jour.



**Figure 26 - Page des informations des commandes**



**Figure 27 - Page des informations personnelle**

## Interface mon compte de l’admin

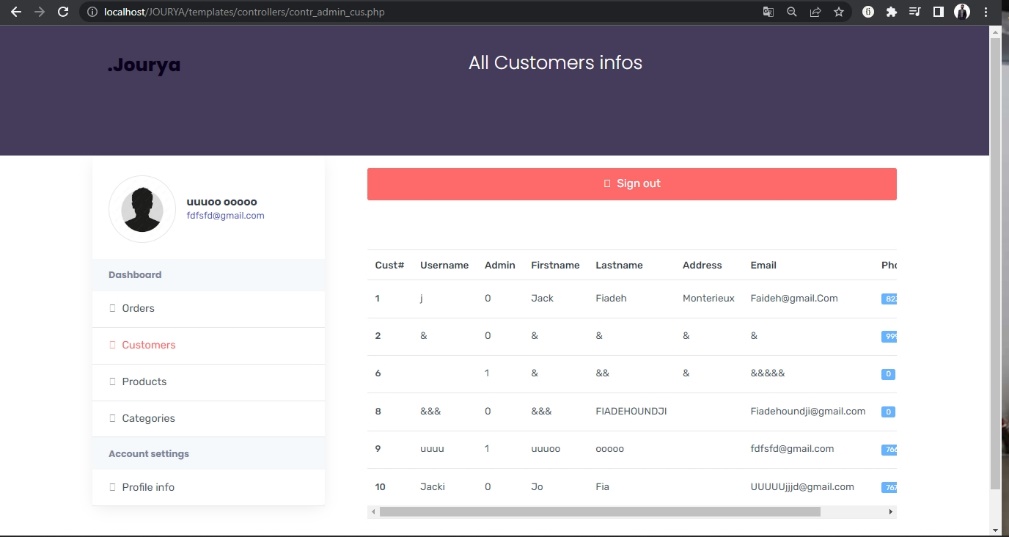
Dans la page Mon Compte réservée à l'administrateur, plusieurs options sont disponibles pour gérer efficacement le site. L'administrateur peut accéder aux sections suivantes :

#### Page des commandes

**Figure 28 - Page de gestion des commandes**

Cette section permet à l'administrateur de visualiser et de gérer les commandes passées par les clients. Il peut consulter les détails des commandes, suivre leur statut, mettre à jour les informations de livraison et gérer les éventuels problèmes liés aux commandes.

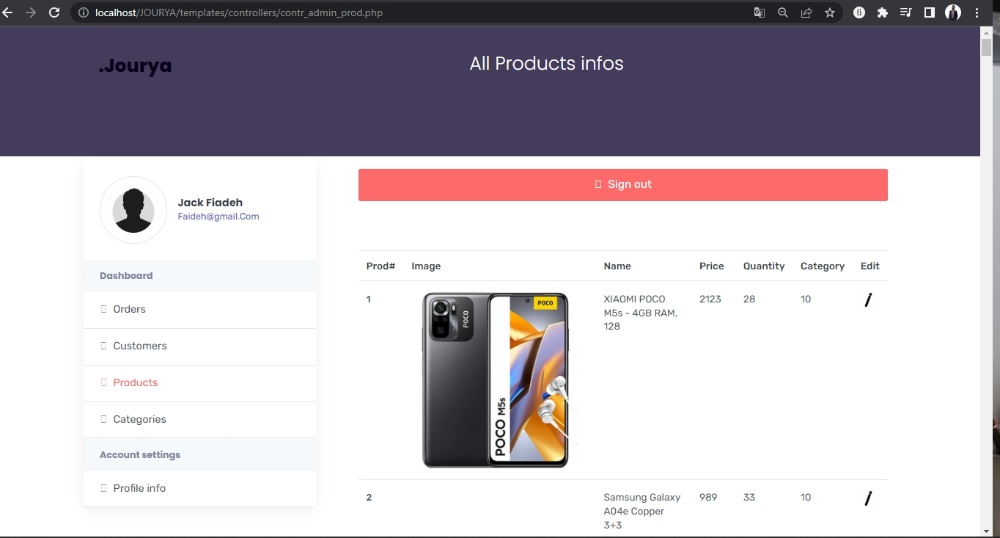
#### Page des clients



**Figure 29 - Page de gestion des clients**

Dans cette section, l'administrateur peut gérer les informations relatives aux clients enregistrés sur le site. Il peut consulter les profils des clients, gérer les informations personnelles, les adresses de livraison et prendre des mesures nécessaires pour assurer une expérience client optimale.

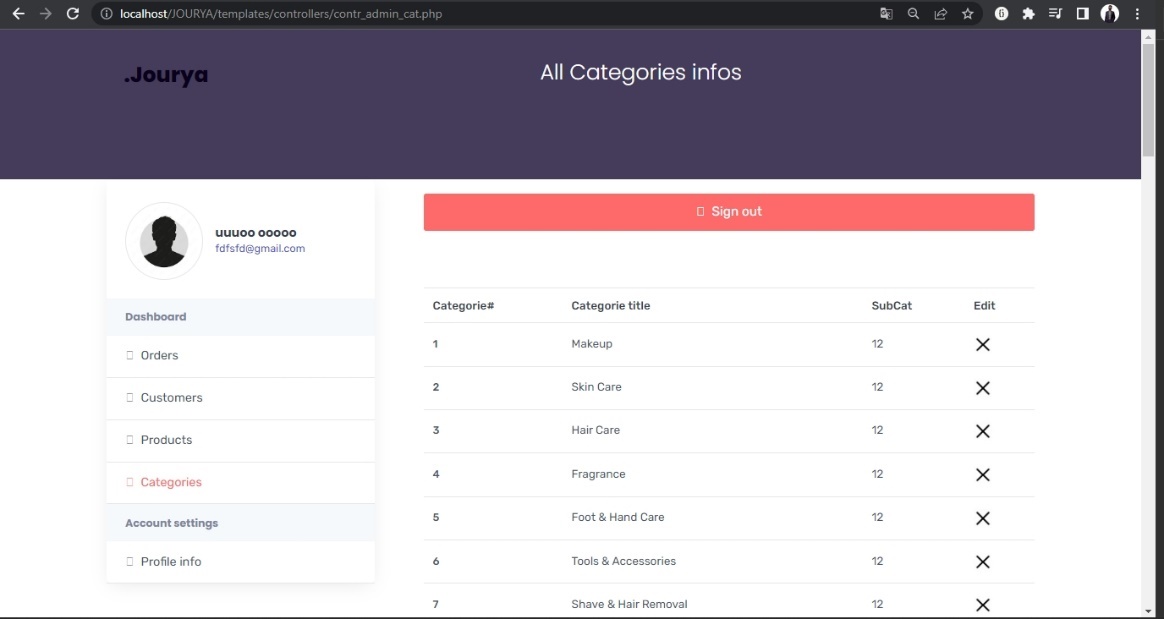
#### Page des produits



**Figure 30 - Page de gestion des produits**

Cette section donne à l'administrateur un contrôle total sur la gestion des produits proposés sur le site. Il peut ajouter de nouveaux produits, mettre à jour les informations existantes, gérer les stocks, les prix et les promotions, et supprimer des produits si nécessaire.

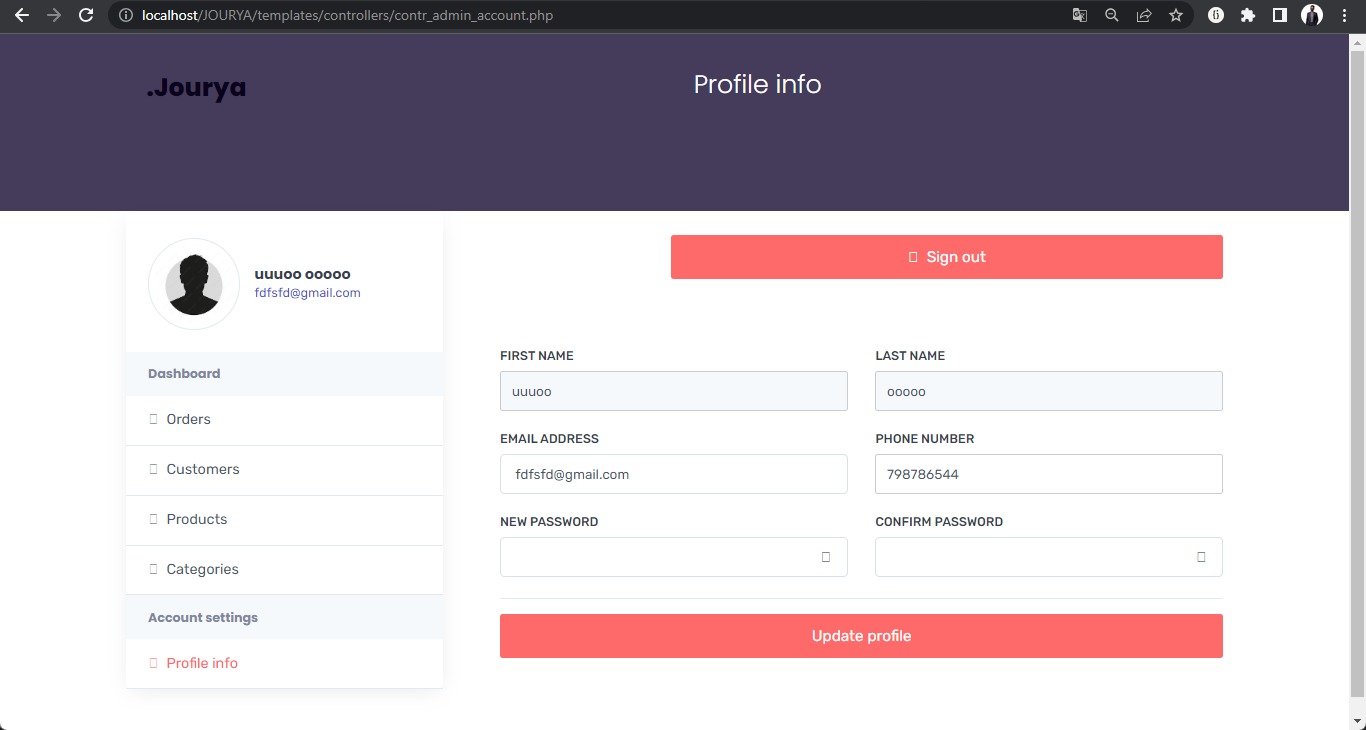
#### Page des catégories



**Figure 31 - Page de gestion des catégories**

Dans cette section, l'administrateur peut gérer les catégories de produits disponibles sur le site. Il peut créer de nouvelles catégories, organiser les produits dans des catégories existantes, apporter des modifications aux catégories existantes et effectuer d'autres tâches liées à la gestion des catégories.

#### Page des informations personnelle



**Figure 32 - Page des informations personnelle d'un admin**

Cette section permet à l'administrateur de mettre à jour ses propres informations de profil, y compris le nom, l'adresse e-mail, le mot de passe, etc. Cela permet à l'administrateur de maintenir ses informations à jour et de garantir la sécurité de son compte.

# Conclusion

Dans cette partie nous avons présenté les différents outils de développement, langages, les schémas de navigation et les maquettes de l’application. Ces outils et langages nous ont permis de mettre au point notre projet tant sur la partie théorique que pratique.

CONCLUSION ET PERSPECTIVES

Dans le cadre de notre projet de fin d’étude de notre cursus en licence, nous avons été amenés à développer un site e-commerce qui est un site de vente en ligne et cela conformément à des exigences stipulées dans le cahier de charge. Ceci nous a fait découvrir de nouvelle architecture, implémenter une réalisation suivant d’autres approches et à enrichir notre savoir et notre expérience.

Ce projet s’inscrit dans le cadre d’une licence d’informatique en Sciences et Techniques option Génie Logiciel au sein de l’UMI, Faculté des Sciences et Techniques d’Errachidia.

C’était vraiment l’occasion pour nous d’approfondir les différentes connaissances acquises durant notre formation. En effet au cours de notre réalisation, nous avons implémenté les fonctionnalités essentielles permettant aux utilisateurs de s'inscrire, de parcourir les produits et les catégories, d'ajouter des articles au panier, de procéder au paiement et de gérer les commandes depuis une interface d'administration. Nous avons mis en place une architecture MVC en utilisant la programmation orientée objet avec PHP. On s’est attelé à rendre notre rendu (nos différentes pages) responsif par rapport à la taille des différents écrans sur lequel l’utilisateur pourrait se trouver. Tout au long de ce travail, nous avons réalisé la création des différentes classes nécessaires pour gérer les utilisateurs, les produits, les catégories et les commandes.

Cependant, notre projet étant assez complet peut être revue et améliorer. Compte tenu du temps et du nombre de ressources très limité que nous avions à notre disponibilité, nous nous sommes attelés à réaliser les fonctionnalités les plus importantes. Notre projet offre donc certaines perspectives d'amélioration et d'extension pour répondre aux besoins des utilisateurs de manière plus avancée. Parmi les perspectives envisageables, nous pouvons mentionner :

* Intégration de modules de machine learning : L'utilisation de techniques de machine learning permettrait de proposer aux utilisateurs une recherche guidée plus efficace et des recommandations personnalisées en fonction de leurs préférences d'achat.
* Utilisation de frameworks plus avancés : Que ce soit sur ce projet ou d’autres, pour améliorer la qualité du code, la performance et la maintenabilité de l'application, il serait judicieux d'explorer des frameworks populaires tels que React, Laravel ou Symfony. Ces frameworks étant plus professionnels, offrent des fonctionnalités avancées et facilitent le développement d'applications web robustes et évolutives.
* De même que les frameworks, sur d’autres projets nous privilégierons l’utilisation d’autres langages plus sécurisés comme J2EE, spring…
* Intégrer un système de paiement comme Stripe

En résumé, ce projet nous a permis de mettre en pratique nos compétences en PHP, en programmation orientée objet et en développement web dans son intégralité. Nous avons acquis une compréhension approfondie de l'architecture MVC et des fonctionnalités d'un site e-commerce. Cependant, nous sommes conscients des possibilités d'amélioration et des nouvelles technologies à explorer pour créer des sites e-commerce plus avancés et adaptés aux besoins du marché. Nous sommes enthousiastes à l'idée de continuer à apprendre et à développer de nouveaux projets en utilisant des technologies innovantes et des bonnes pratiques de développement.

BIBLIOGRAPHIE & WEBOGRAPHIE

* 1. **Bibliographie**

[1] : Pr. Fouad YAKOUBI cours programmation web 2022/2023

[2] : Pr. Brahim OUCHAO cours génie logiciel 2021/2022.

[3] : Pr. Abdeslam JAKIMI cours UML 2022/2023.

[4] : Pr. Mohamed OUALLA cours programmation orientée objet avec Java 2022/2023.

1. **Webographie**

[1] : https://www.ibm.com/docs/fr/

[2] : https://openclassrooms.com/

[3] : https://www.w3schools.com

[4] : https://fr.wikipedia.org/wiki/

[5] : https://github.com/