

# **RAPPORT DE PROJET CLIENT**

## **APPLICATION DE VENTE DE VOITURE EN LIGNE**

**Rédigé et présenté par :**

**CHARLES PHILIPPE TONYE TONYE**

*Année : 2022- 2023*

## Table des matières

Table des matières .....	2
1. INTRODUCTION .....	4
2. Analyse des besoins.....	4
2.1 Besoins fonctionnels.....	4
2.2 Besoins non fonctionnels .....	5
2.3 Identification des acteurs .....	5
2.3.1 PLANIFICATION DU PROJET .....	6
2.3.2 Financement/contraintes .....	6
2.3.3 Livrables.....	7
2.4 Quelques contraintes .....	8
2.4.1. Contraintes de vérification .....	8
2.4.2. Contraintes de non nullité (NOT NULL) .....	8
2.4.3. Contraintes d'unicité .....	8
2.4.4. Contraintes clés primaires .....	9
2.4.5. Contraintes Clé étrangères.....	9
3. Conception de la base de données.....	9
3.1 Identification des entités.....	10
3.2 Relations et cardinalités .....	10
3.3 Création des tables et des contraintes d'intégrité .....	10
3.3.1 Création des tables.....	10
3.3.2 Contraintes d'intégrités.....	11
4. Quelques requêtes CRUD .....	11
4.1 Requête de création : INSERT INTO.....	11
4.2 Requête de la lecture (Read) : Instruction SELECT .....	11
4.3 Requêtes de modification : Instruction Update .....	12
4.4 Requêtes de suppression : Instruction DELETE .....	12
5. DOSSIER DE CONCEPTION .....	12
5.1 Schéma du modèle conceptuel de données .....	12
5.2 Schéma du modèle logique de données .....	13
6. LES DIAGRAMME DE SEQUENCE .....	14
7. MISE EN ŒUVRE .....	15
7.1 CHOIX DES OUTILS .....	15
7.1.1 Environnement matériel.....	15

7.1.2. Environnement logiciel.....	15
7.1.3. Langage de programmation .....	16
7.2 DIAGRAMME DE DEPLOIEMENT.....	16
7.2.1 L'architecture physique (ou matérielle) .....	16
7.2.2 L'ARCHITECTURE LOGIQUE.....	16
8. IMPLEMENTATION.....	17
8.1 GUIDE D'INSTALLATION.....	17
8.2 GUIDE D'UTILISATEUR .....	18
Interface de connexion Admin .....	18
Interfaces Liste des Produits :.....	18
Interfaces de Produits –Voiture enregistré .....	19
Interfaces Rapport de Produits Commandé :.....	19
Interfaces Commandes statut validés : .....	20
Interfaces connexion client : .....	20
Interfaces choix voiture par catégories .....	21
Interfaces Préférences (Réservation commande).....	21
Interfaces reçu bon commande :.....	22
9. CONCLUSION GENERALE .....	22
10. PERSPECTIVES DE DEVELOPPEMENT .....	22

## 1. INTRODUCTION

Ce rapport présente notre projet de conception et développement d'une application web de gestion des annonces de vente de voitures en ligne. L'objectif principal de ce projet est de fournir aux utilisateurs une plateforme conviviale et efficace pour visualiser, sélectionner et acheter des voitures en ligne.

Le rapport décrit en détail les étapes d'analyse et de conception du projet, en mettant l'accent sur l'identification des besoins fonctionnels et non fonctionnels de l'application, la conception de la base de données et l'implémentation des fonctionnalités CRUD (Create, Read, Update, Delete). Les contraintes d'intégrité mises en place dans la base de données pour garantir la cohérence des données sont également abordées.

L'objectif final de ce rapport est de fournir une documentation complète du processus de développement de notre application web de gestion des annonces de vente de voitures en ligne, en mettant en évidence les choix clés et en offrant des perspectives d'amélioration pour l'avenir.

## 2. Analyse des besoins

Dans cette section, nous allons identifier et décrire les besoins fonctionnels et non fonctionnels de notre application.

### 2.1 Besoins fonctionnels

Dans cette sous-section, énumérons et décrivons les fonctionnalités clés que notre application doit fournir.

- Création d'annonces : Permettre aux vendeurs de créer de nouvelles annonces pour les voitures à vendre. Les informations nécessaires peuvent inclure le modèle, la marque, l'année, le prix, les caractéristiques, etc.
- Recherche et filtrage : Offrir aux utilisateurs la possibilité de rechercher et filtrer les annonces en fonction de critères spécifiques tels que la marque, le modèle, l'année, la gamme de prix, etc.
- Consultation des détails des voitures : Permettre aux utilisateurs de consulter les détails complets des voitures, y compris les caractéristiques techniques, les photos, l'historique d'entretien, etc.
- Mise à jour des informations : Autoriser les vendeurs à mettre à jour les informations des annonces existantes, comme le prix, les caractéristiques, etc.
- Suppression des annonces : Permettre aux vendeurs de supprimer les annonces qui ne sont plus valides ou pertinentes.
- Gestion des utilisateurs : Mettre en place un système d'inscription, de connexion et de gestion des comptes utilisateurs pour les vendeurs et les acheteurs.
- Panier d'achat : Offrir aux utilisateurs la possibilité d'ajouter des voitures à leur panier d'achat et de procéder au paiement.

## 2.2 Besoins non fonctionnels

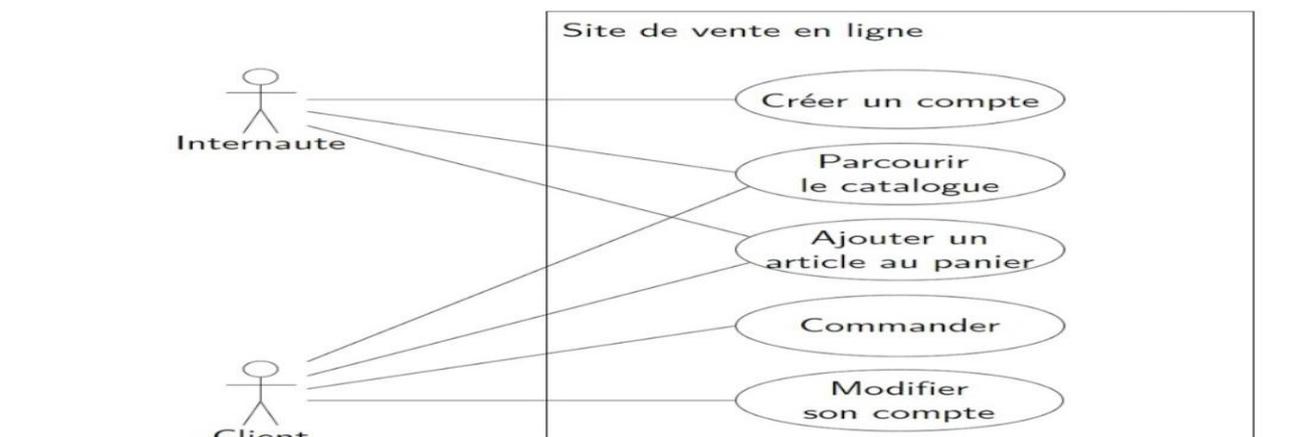
Dans cette sous-section, détaillons les besoins non fonctionnels qui sont essentiels pour garantir la qualité de notre application.

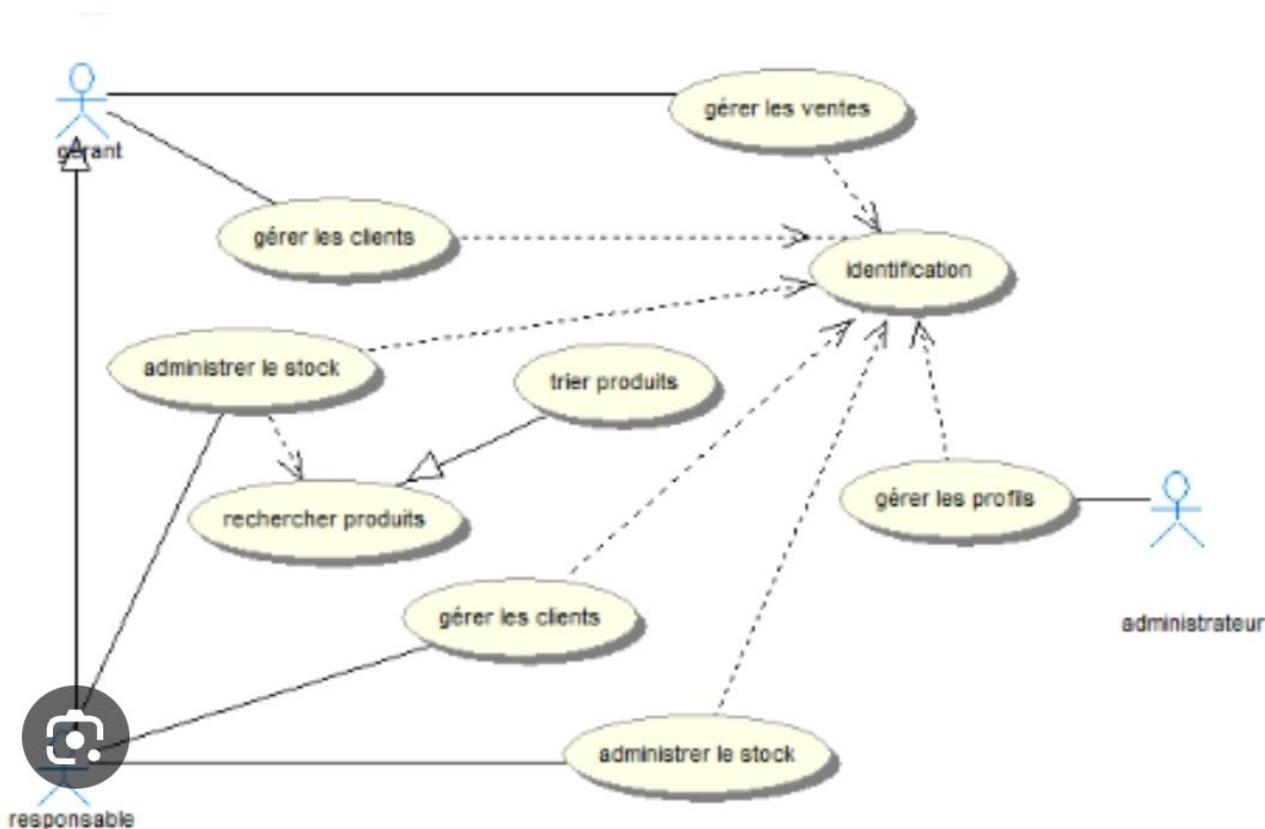
- Performance : L'application doit être rapide et réactive, en fournissant des temps de chargement rapides pour les pages et les résultats de recherche.
- Sécurité : Assure-toi que l'application met en œuvre des mécanismes de sécurité tels que l'authentification des utilisateurs, la protection contre les attaques par injection SQL, etc.
- Disponibilité : L'application doit être disponible et accessible en ligne de manière fiable, sans interruptions majeures de service.

## 2.3 Identification des acteurs

Dans cette sous-section, identifie les différents acteurs qui interagiront avec notre application.

- Utilisateurs anonymes : Les visiteurs qui accèdent au site sans être connectés.
- Vendeurs ou Personnel(Staff) : Les utilisateurs qui souhaitent vendre des voitures en créant des annonces.
- Acheteurs : Les utilisateurs qui recherchent et souhaitent acheter des voitures.
- Administrateurs : Les utilisateurs chargés de gérer et de superviser le bon fonctionnement de l'application.





### 2.3.1 PLANIFICATION DU PROJET

La planification d'un projet est très importante lors de sa réalisation car elle consiste à déterminer et ordonnancer les tâches du projet. Ce qui permettra de déterminer si les objectifs sont réalisés et entre autres suivre et communiquer l'avancement du projet. Nous avons ainsi ressorti la planification de l'évolution de notre projet.

Il se résume par le diagramme de Gantt qui présente les tâches ou activités à réaliser et le temps nécessaires pour leur réalisation.

### 2.3.2 Financement/contraintes

*Tableau : coût des ressources*

Numéro	Tâche	Date début	Date de fin	Durée							
1	Rédaction du rapport	19/06/2023	23/06/2023	05							
2	Dossier d'insertion	19/06/2023	20/06/2023	02							
3	Cahier de charge	21/06/2023	22/06/2023	02							
4	Cahier d'analyse	22/06/2023	23/06/2023	02							
5	Cahier de conception	20/06/2023	22/06/2023	03							
6	Mise en œuvre	20/06/2023	23/06/2023	04							

RESSOURCES	NOMS DES RESSOURCES	CARACTERISTIQUES	PRIX UNITAIRE	PRIX TOTAL
<b>Matérielles</b>	03 ordinateurs portables	HP Corei3 Disque dur 500giga Ram 4giga Processeur 2.30GHZ	300 000FCFA	900 000FCFA
	01modem	Camtel 4G	20 000FCFA	20 000FCFA
	01 Rame de papier	Format A4 double A	3000FCFA	3000FCFA
	04Stylos	Schneider	200FCFA	800FCFA
<b>Logiciels</b>	Server Local	xampp-win32-5.6.38-0-VC11	Gratuit	/
	Système d'exploitation	Windows 10	60 000FCFA	60 000FCFA
	MySQL	Version 8.0.16	Gratuit	/
	Adobe Acrobat Reader DC	Version 21.005.20048	Gratuit	/
	Suite Microsoft Office	2016	20 000FCFA	20 000FCFA
	Antivirus	Kaspersky	72 000FCFA	72 000FCFA

### 2.3.3 Livrables

En gestion de projet, un livrable désigne tout produit fourni pendant la réalisation du projet et nécessaire pour atteindre les objectifs. C'est aussi l'ensemble des composants matérialisant le résultat d'une prestation de réalisation : **c'est le résultat tangible d'une production réelle, appréhendable et mesurable**. Les livrables de notre projet sont :

- Une application fonctionnelle
- Un manuel utilisateur et guide d'installation ;
- Le package de développement sur un CD ;

- Le rapport du projet.

## 2.4 Quelques contraintes

Il s'agit de types de moyen de restreindre la nature de type de données qui peuvent être stockées dans une table. Par exemple une colonne qui contient le prix 'un produit (voiture) ne doit accepter que des valeurs positives. De plus dans une table contenant des informations de produit, il ne peut y avoir qu'une ligne par numéro de produit. Voici quelques exemples de contraintes :

### 2.4.1. Contraintes de vérification

La contrainte de vérification est la plus générique qui soit. Par exemple pour obliger les prix des produits (Voitures) à être positifs, on peut utiliser :

```
ALTER TABLE `tblproduct`  
  ADD PRIMARY KEY (`PROID`);  
COMMIT;
```

```
CREATE TABLE produits (  
  No_produit integer,  
  Nom text,  
  Prix numeric CHECK (prix > 0)  
);
```

### 2.4.2. Contraintes de non nullité (NOT NULL)

Cette contrainte tester dans notre base de données indique simplement qu'une colonne ne peut pas prendre la valeur NULL. Par exemple dans notre base de données :

```
CREATE TABLE `tblproduct` (  
  `PROID` int(11) NOT NULL,  
  `PRODESC` varchar(255) DEFAULT NULL,  
  `PROQTY` int(11) NOT NULL,  
  `ORIGINALPRICE` double NOT NULL,  
  `PROPRICE` double DEFAULT NULL,  
  `CATEGID` int(11) DEFAULT NULL,  
  `IMAGES` varchar(255) DEFAULT NULL,  
  `PROSTATS` varchar(30) DEFAULT NULL,  
  `OWNERNAME` varchar(90) NOT NULL,  
  `OWNERPHONE` varchar(90) NOT NULL  
); ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Une contrainte NOT NULL est toujours écrite comme contrainte de colonne.

### 2.4.3. Contraintes d'unicité

Elles garantissent l'unicité des données contenues dans une colonne ou un groupe de colonnes par rapport à toutes les lignes de la table. Dans notre base de données on a :

```

CREATE TABLE `tblproduct` (
  `PROID` int(11) NOT NULL,
  `PRODESC` varchar(255) DEFAULT NULL,
  `PROQTY` int(11) NOT NULL,
  `ORIGINALPRICE` double NOT NULL,
  `PROPRICE` double DEFAULT NULL,
  `CATEGID` int(11) DEFAULT NULL,
  `IMAGES` varchar(255) DEFAULT NULL,
  `PROSTATS` varchar(30) DEFAULT NULL,
  `OWNERNAME` varchar(90) NOT NULL,
  `OWNERPHONE` varchar(90) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

#### 2.4.4. Contraintes clés primaires

Elle qu'une colonne, ou un groupe de colonnes, peut être utilisée comme un identifiant unique de ligne pour cette table. Une table a au plus une clé primaire.

```

CREATE TABLE produits (
  No_produit integer PRIMARY KEY,
  Nom text,
  Prix numeric
) ;

```

#### 2.4.5. Contraintes Clé étrangères

Une contrainte clé étrangère stipule que les valeurs d'une colonne (ou groupe de colonnes) doivent correspondre aux valeurs qui apparaissent dans les lignes d'une autre table. On dit que cela maintient l'intégrité référentielle entre les deux tables.

```

CREATE TABLE `tblorder` (
  `ORDERID` int(11) NOT NULL,
  `PROID` int(11) NOT NULL,
  `ORDEREDQTY` int(11) NOT NULL,
  `ORDEREDPRICE` double NOT NULL,
  `ORDEREDNUM` int(11) NOT NULL,
  `USERID` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

### 3. Conception de la base de données

Dans cette section, nous allons concevoir la structure de la base de données en identifiant les entités, les relations entre elles, et en définissant les cardinalités. Voici les étapes clés de la conception :

### 3.1 Identification des entités

Identifions les principales entités impliquées dans notre application.

- Voiture : Représente les voitures disponibles à la vente. Les attributs peuvent inclure le modèle, la marque, l'année, le prix, etc.
- Annonce : Représente les annonces de vente de voitures. Les attributs peuvent inclure la date de création, la description, le statut, etc.
- Utilisateur : Représente les utilisateurs de l'application. Les attributs peuvent inclure le nom, l'adresse e-mail, le mot de passe, etc.
- Panier : Représente le panier d'achat d'un utilisateur. Il peut être lié à l'utilisateur et contenir les voitures sélectionnées pour l'achat.

### 3.2 Relations et cardinalités

Identifions les relations entre les entités et déterminons les cardinalités de ces relations.

- Une voiture peut être associée à plusieurs annonces, mais une annonce ne concerne qu'une seule voiture. Il s'agit d'une relation "plusieurs à un" (many-to-one) entre Voiture et Annonce.
- Un utilisateur peut créer plusieurs annonces, mais une annonce est créée par un seul utilisateur. C'est également une relation "plusieurs à un" entre Utilisateur et Annonce.
- Un utilisateur peut avoir un panier d'achat contenant plusieurs voitures, et une voiture peut être présente dans le panier de plusieurs utilisateurs. Il s'agit d'une relation "plusieurs à plusieurs" (many-to-many) entre Utilisateur et Voiture, qui est modélisée en utilisant une table de jonction.

Détermine les clés primaires et les clés étrangères nécessaires pour maintenir l'intégrité référentielle entre les tables.

### 3.3 Création des tables et des contraintes d'intégrité

#### 3.3.1 Création des tables

Crée les tables dans notre système de gestion de base de données (SGBD) en utilisant les entités identifiées et les relations définies tout en incluant les attributs appropriés pour chaque table. On par exemple la table utilisateur dont la requête de création est identique pour toutes les autres entités :

```
CREATE TABLE `tbluseraccount` (
  `USERID` int(11) NOT NULL,
  `U_NAME` varchar(122) NOT NULL,
  `U_USERNAME` varchar(122) NOT NULL,
  `U_PASS` varchar(122) NOT NULL,
  `U_ROLE` varchar(30) NOT NULL,
  `USERIMAGE` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

### 3.3.2 Contraintes d'intégrités

Définir les contraintes d'intégrité, telles que les clés primaires, les clés étrangères et les contraintes de domaine nous permet de garantir l'intégrité des données et maintenir la cohérence de la base de données.

## 4. Quelques requêtes CRUD

### 4.1 Requête de création : INSERT INTO

```
public function create() {
    global $mydb;
    // Don't forget your SQL syntax and good habits:
    // - INSERT INTO table (key, key) VALUES ('value', 'value')
    // - single-quotes around all values
    // - escape all values to prevent SQL injection
    $attributes = $this->sanitized_attributes();
    $sql = "INSERT INTO ".$this->tblname." (";
    $sql .= join(", ", array_keys($attributes));
    $sql .= ") VALUES (";
    $sql .= join("'", array_values($attributes));
    $sql .= "'')";
    echo $mydb->setQuery($sql);

    if($mydb->executeQuery()) {
        $this->id = $mydb->insert_id();
        return true;
    } else {
        return false;
    }
}
```

### 4.2 Requête de la lecture (Read) : Instruction SELECT

```
<?php
$mydb->setQuery("SELECT * FROM `tblcategory`");
$cur = $mydb->loadResultList();

foreach ($cur as $result) {
    echo '<tr>';
    // echo '<td width="5%" align="center"></td>';
    // echo '<td>
    //     <input type="checkbox" name="selector[]" id="sel-
    //     ' . $result->CATEGORIES.'</a></td>';
    echo '<td>' . $result->CATEGORIES.'</td>';
    echo '<td align="center"><a title="Edit" href="index.php
    <a title="Delete" href="controller.php?action=delete
    // echo '<td></td>';
    echo '</tr>';
}
?>
```

### 4.3 Requetes de modification : Instruction Update

```
public function update($id=0) {
    global $mydb;
    $attributes = $this->sanitized_attributes();
    $attribute_pairs = array();
    foreach($attributes as $key => $value) {
        $attribute_pairs[] = "{$key}='{$value}'";
    }
    $sql = "UPDATE ".self::$tblname." SET ";
    $sql .= join(", ", $attribute_pairs);
    $sql .= " WHERE CATEGID=". $id;
    $mydb->setQuery($sql);
    if(!$mydb->executeQuery()) return false;
}
```

### 4.4 Requetes de suppression : Instruction DELETE

```
public function delete($id=0) {
    global $mydb;
    $sql = "DELETE FROM ".self::$tblname;
    $sql .= " WHERE CATEGID=". $id;
    $sql .= " LIMIT 1 ";
    $mydb->setQuery($sql);

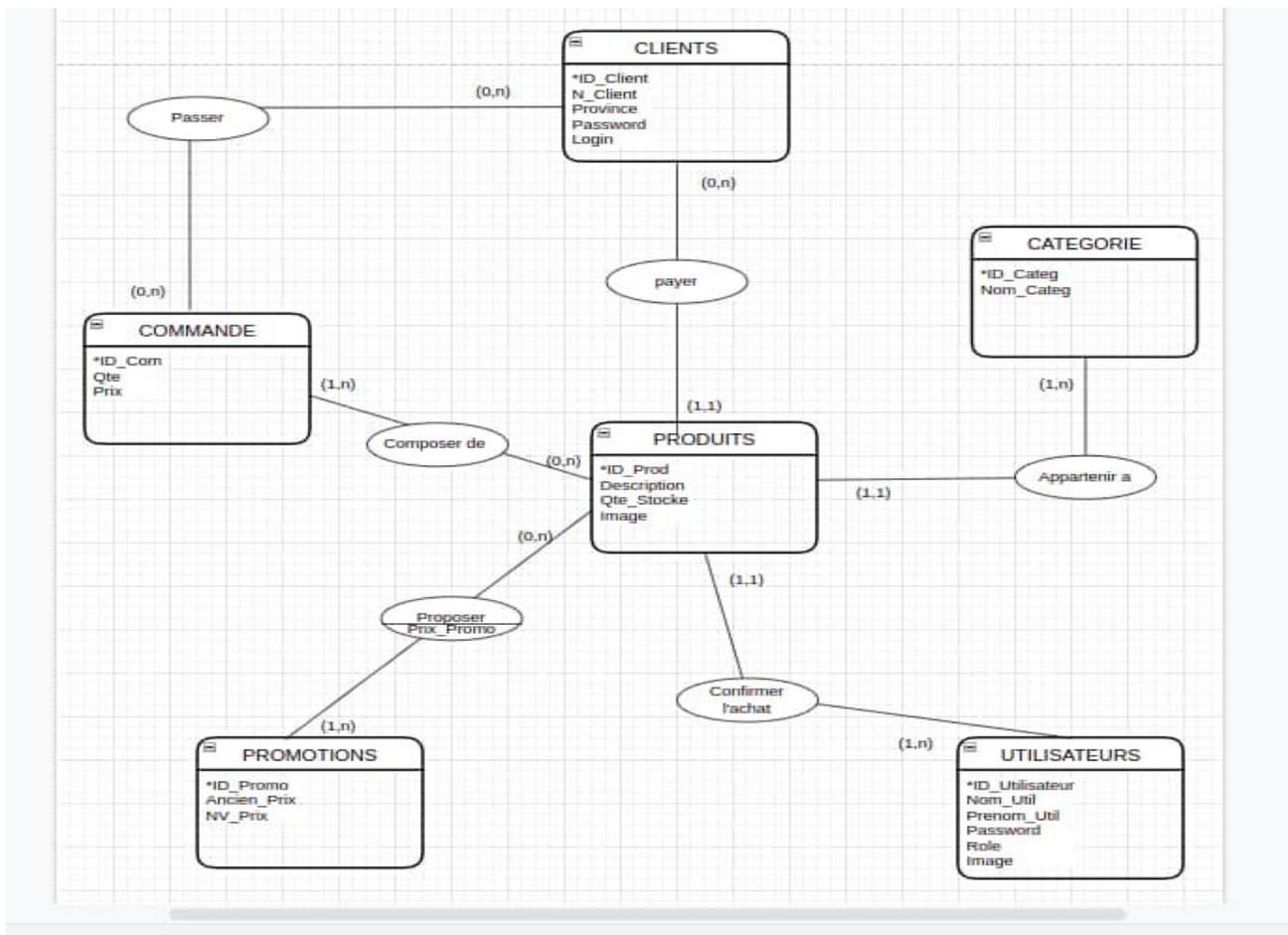
    if(!$mydb->executeQuery()) return false;
}
```

## 5. DOSSIER DE CONCEPTION

En gestion de projet, le dossier de conception constitue une étape incontournable. Il nous permet de visualiser et décrire les exigences du système et présenter la structure du futur système.

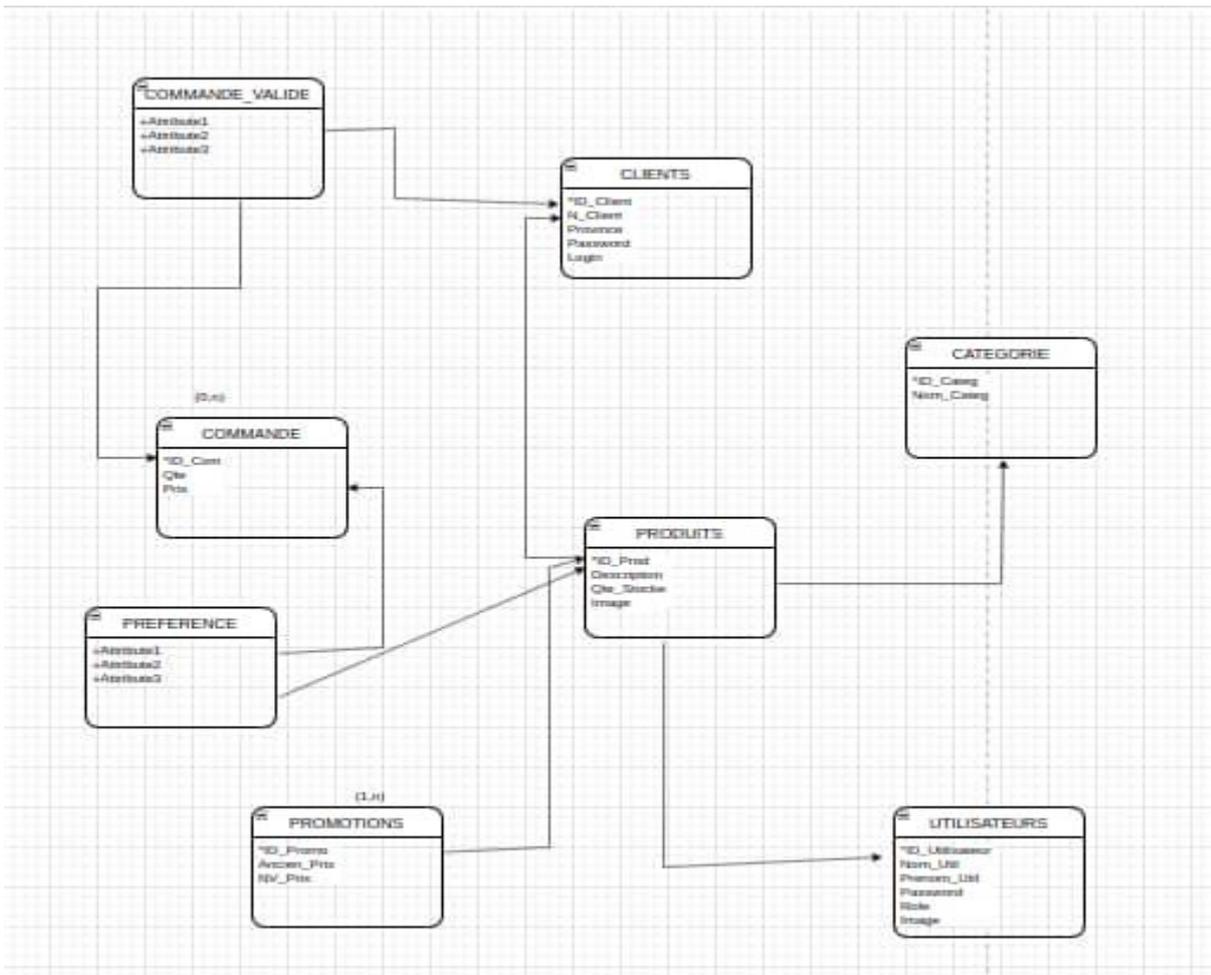
### 5.1 Schéma du modèle conceptuel de données

C'est d'une représentation graphique des concepts et des relations entre les entités dans un système d'information. Dans la vente de voiture en ligne nous avons décrit la structure logique des données et les liens entre les entités sans tenir compte des détails techniques de la mise en œuvre.



## 5.2 Schéma du modèle logique de données

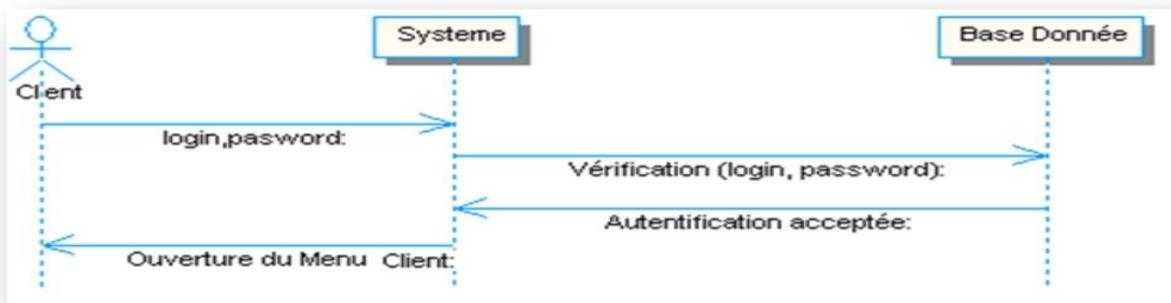
C'est une représentation plus détaillée du MCD, spécifiant les entités, les relations et les contraintes. Il se concentre sur la façon dont les données seront stockées dans la base de données spécifique, en prenant en compte les types de données, les clés primaires, les clés étrangères, etc.

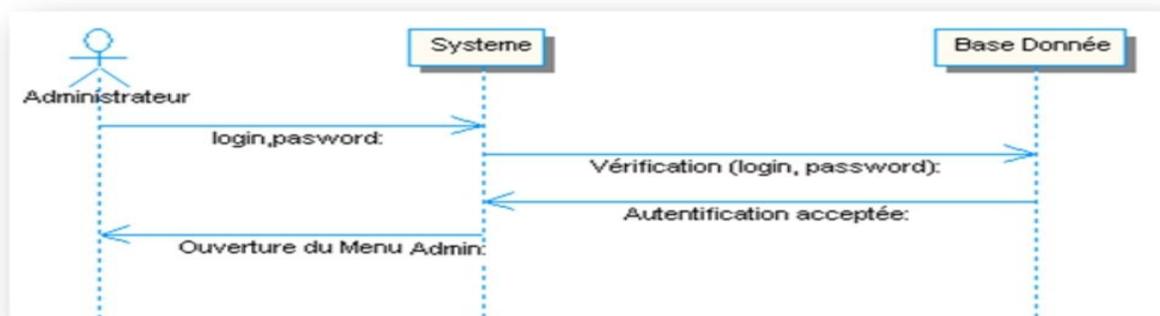


## 6. LES DIAGRAMME DE SEQUENCE

Les diagrammes de séquence documentent les interactions à mettre en œuvre entre les classes pour réaliser un résultat, tel qu'un cas d'utilisation. UML étant conçu pour la programmation orientée objet, ces communications entre les classes sont reconnues comme des messages. Le diagramme de séquence énumère des objets horizontalement, et le temps verticalement. Il modélise l'exécution des différents messages en fonction du temps.

**Représentation :** dans un diagramme de séquence, les classes et les acteurs sont énumérés en colonne avec leurs lignes de vie verticales indiquant la durée de vie de l'objet.





## 7. MISE EN ŒUVRE

La phase de mise en œuvre renvoie au dossier de réalisation de la solution. Il permet de décrire de manière physique l'application c'est l'aboutissement de tous les dossiers précédant. Notre travail dans ce dossier consistera à présenter les technologies utilisées ainsi que le diagramme de déploiement.

### 7.1 CHOIX DES OUTILS

Le choix des outils nécessite une attention particulière car nous permettra de visualiser notre application sur le plan structurel et aussi environnemental. Nous avons recueilli l'ensemble d'outils nécessaires à la réalisation de notre application dans le tableau ci-après.

#### 7.1.1 Environnement matériel

matériel	quantité
Ordinateur HP Disque dur 500 giga Ram 4 giga Processeur 1.6GHZ	05
Modem	01
Imprimante	01

#### 7.1.2. Environnement logiciel

Types	Noms
Systèmes d'exploitation	Windows10
Système de gestion de base de données	MySQL
Logiciel de modélisation	Windesign, StarUML, Power AMC
Environnement de développement	Template
Logiciel de traitement de texte	Microsoft office Word 2010, 2013
Logiciel de planification	Xampp

### 7.1.3. Langage de programmation

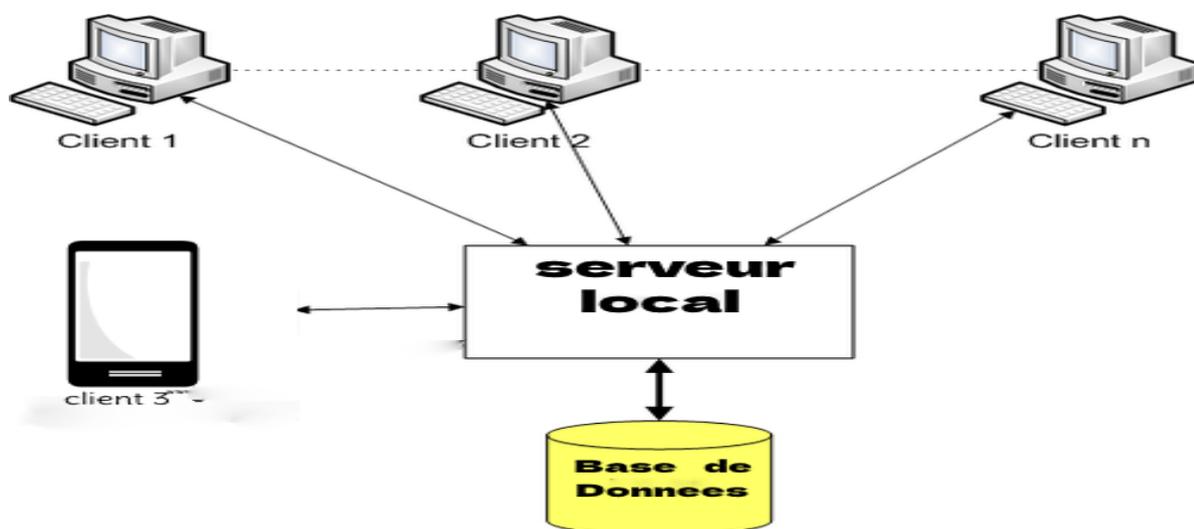
Nous avons choisi de développer notre application en avec un Template contenant Bootstrap pour les vues et PHP, JavaScript, HTML comme langage de programmation.

## 7.2 DIAGRAMME DE DEPLOIEMENT

Le diagramme de déploiement modélise les composants matériels utilisés pour implémenter un système et l'association entre ces composants. Des diagrammes de déploiement peuvent être mise en œuvre dès la phase de conception pour documenter l'architecture physique et logique du système.

### 7.2.1 L'architecture physique (ou matérielle)

Elle décrit les composants concrets: les ordinateurs (postes clients), les téléphones mobiles, le stockage, le serveur local qui communique avec la base de données de l'application, etc.

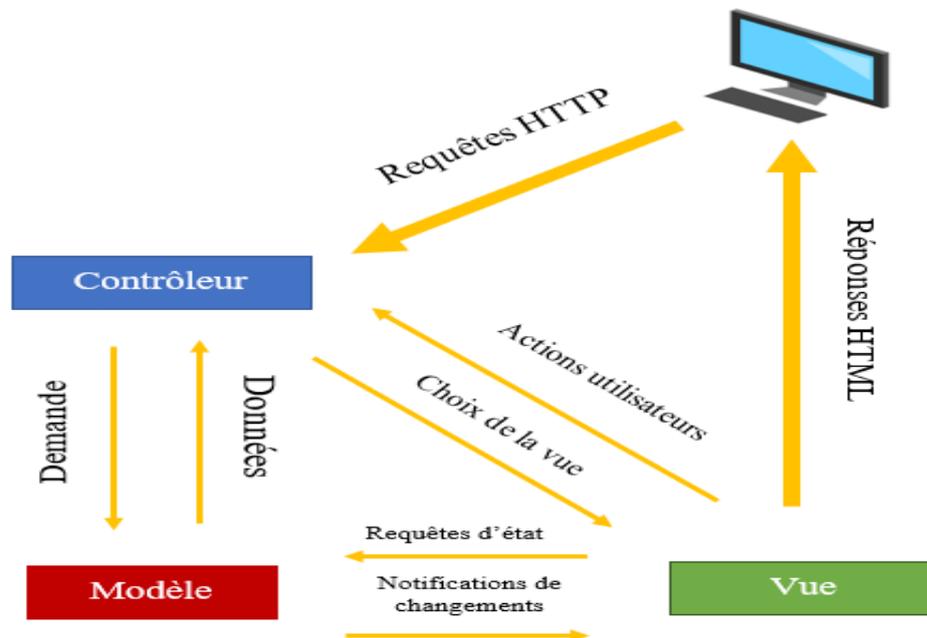


### 7.2.2 L'ARCHITECTURE LOGIQUE

Pour le cas d'espèce nous allons opter pour une architecture MVC. Le pattern MVC permet de bien organiser son code source. Il va nous aider à savoir quels fichiers créent, mais surtout à définir leur rôle logique. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts.

- **Modèle** : cette partie gère les données de votre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc entre autres les requêtes SQL.
- **Vue** : cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple une liste d'enseignants pour le cas d'espèce.

- **Contrôleur** : cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès).



## 8. IMPLEMENTATION

### 8.1 GUIDE D'INSTALLATION

Le procédé d'installation de notre application est très simple, il suffit d'installer sur son poste WAMP Server ou XAMPP Server qui contient MySQL pour la gestion de la base de données. Vous devez donc suivre les instructions suivantes :

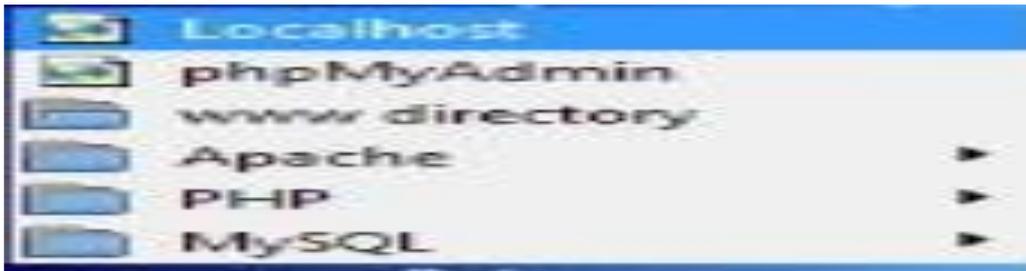
- Avoir un ordinateur ayant au moins les caractéristiques suivantes : Disque dur (5G0), processeur (1.5GHZ), RAM (1GO), système d'exploitation (Windows 10)
- Télécharger XAMPP CONTROL PANEL en allant sur le site officiel de XAMPP Server et l'installer en suivant tout simplement les instructions. Une fois le XAMPP téléchargé, double-cliquez pour l'installer sur votre appareil. Une fenêtre s'ouvre alors, vous demandant l'autorisation d'effectuer des modifications sur votre appareil. Cliquez sur **OUI**. Dans l'étape suivante, sélectionnez la langue de configuration du serveur XAMPP. Vous pouvez choisir entre le français et l'anglais. Cliquez ensuite sur **ok**. Un contrat de licence apparaîtra ensuite. Lisez l'accord si vous le souhaitez, puis cliquez sur **J'accepte**. Dans l'étape suivante, cliquez simplement sur **Suivant**.

Maintenant, XAMPP Server vous demandera l'emplacement de votre installation. Vous pouvez sélectionner l'emplacement en cliquant sur l'option Parcourir ou simplement sur Suivant.

Cliquez enfin sur Installer après avoir choisi l'emplacement.

Après l'installation, le serveur XAMPP vous expliquera son fonctionnement. Vous pouvez le lire, puis cliquer sur **Suivant**. Enfin, cliquez sur l'option **Terminer** pour terminer le processus.

Ainsi, XAMPP CONTROL PANEL est installé avec succès sur votre système Windows 10.



Lancer le logiciel qui a été fourni dans une clé USB ou tout support de stockage en faisant une copie du dossier dans le chemin : **C:/xampp/htdocs/ecommerce.**

➤ **Connexion comme client sur l'application :**

Saisir dans un navigateur de votre choix dans la barre URL : localhost/ecommerce/ afin de vous connecter comme client avec le login : nawal et passwd nawalnawal.

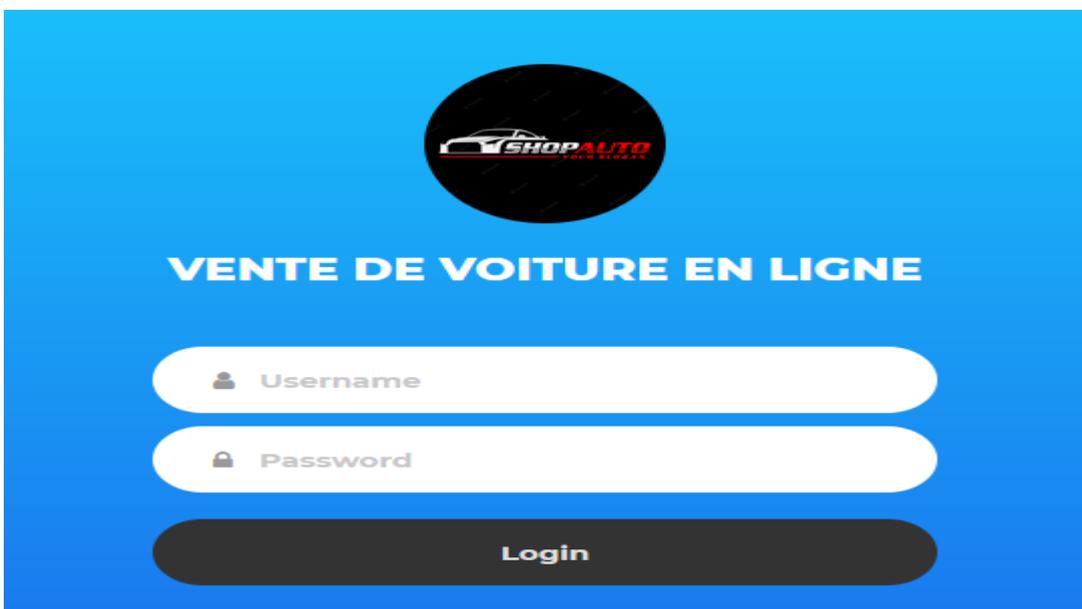
➤ **Connexion comme administrateur :**

Saisir dans un navigateur de votre choix dans la barre URL : localhost/ecommerce/ admin afin de vous connecter comme Administrateur avec le login : janobe et passwd : admin.

## 8.2 GUIDE D'UTILISATEUR

Le guide d'utilisateur explique la démarche d'utilisation de notre application. En effet, notre application comprend plusieurs interfaces. Ces interfaces sont :

### Interface de connexion Admin



### Interfaces Liste des Produits :

localhost / 127.0.0.1 / db\_ecom x Produits x +

localhost/ecommerce/admin/products/

UYDEI\_INF3036 E-Commerce + Nouveau Ydel, Janobe Palacios

Produits /

## Liste des Produits [Ajouter](#)

Show 10 entries Search:

#	Image	Produit	Description	Prix	%Remise	Prix réduit	Quantité
<input type="checkbox"/>		TOYOTA	Voiture de luxe consomme moins	Fcfa 6,000,000.00	Fcfa 0	Fcfa 6,000,000.00	40
<input type="checkbox"/>		TOYOTA	voiture stable et rapide	Fcfa 4,000,000.00	Fcfa 0	Fcfa 4,000,000.00	50
<input type="checkbox"/>		TOYOTA	TOYOTA Futuriste 2035	Fcfa 7,500,000.00	Fcfa 0	Fcfa 7,500,000.00	50
<input type="checkbox"/>		MERCEDES	moins rapide et consomme plus	Fcfa 4,000,000.00	Fcfa 0	Fcfa 4,000,000.00	50
<input type="checkbox"/>		MERCEDES	Rapide sur terre battue	Fcfa 2,000,000.00	Fcfa 0	Fcfa 2,000,000.00	30
<input type="checkbox"/>		MERCEDES	Voiture rapide et Automatique	Fcfa 4,500,000.00	Fcfa 0	Fcfa 4,500,000.00	50

## Interfaces de Produits –Voiture enregistré

localhost / 127.0.0.1 / db\_ecom x Settings x +

localhost/ecommerce/admin/settings/index.php

UYDEI\_INF3036 E-Commerce + Nouveau Ydel, Janobe Palacios

Settings /

Paramètre

Produit

## Liste de Produits

Show 10 entries Search:

Produits	Categorie	Prix	Action
Voiture de luxe consomme moins	TOYOTA	Fcfa 6,000,000.00	<a href="#">Available</a> <a href="#">Set Discount</a>
voiture stable et rapide	TOYOTA	Fcfa 4,000,000.00	<a href="#">Available</a> <a href="#">Set Discount</a>
TOYOTA Futuriste 2035	TOYOTA	Fcfa 7,500,000.00	<a href="#">Available</a> <a href="#">Set Discount</a>
moins rapide et consomme plus	MERCEDES	Fcfa 4,000,000.00	<a href="#">Available</a> <a href="#">Set Discount</a>
Rapide sur terre battue	MERCEDES	Fcfa 2,000,000.00	<a href="#">Available</a> <a href="#">Set Discount</a>
Voiture rapide et Automatique	MERCEDES	Fcfa 4,500,000.00	<a href="#">Available</a> <a href="#">Set Discount</a>

## Interfaces Rapport de Produits Commandé :

Report

From: 06/21/2023 To: 06/23/2023 [Rechercher](#)

### Liste des Produits Commandés

Inclusive Dates: From : 06/21/2023 - To : 06/23/2023

Date Commande	Produits	Prix Original	Prix	Quantité	Sub-total
Jun/22/2023 05:44:47	LAMBORGHINI Futuriste	13000000	25000000	1	25000000
Jun/22/2023 05:44:47	TOYOTA Futuriste 2035	7000000	7500000	1	7500000
Jun/22/2023 05:44:47	Voiture de luxe consomme moins	5000000	6000000	1	6000000
Total		25000000	38500000	3	38500000

[Rapport d'impression](#)

Copyright © Bachelor's degree in mathematics and computer science

## Interfaces Commandes statut validés :

UYDEI\_INF3036 E-Commerce

Orders /

### Liste de Commande

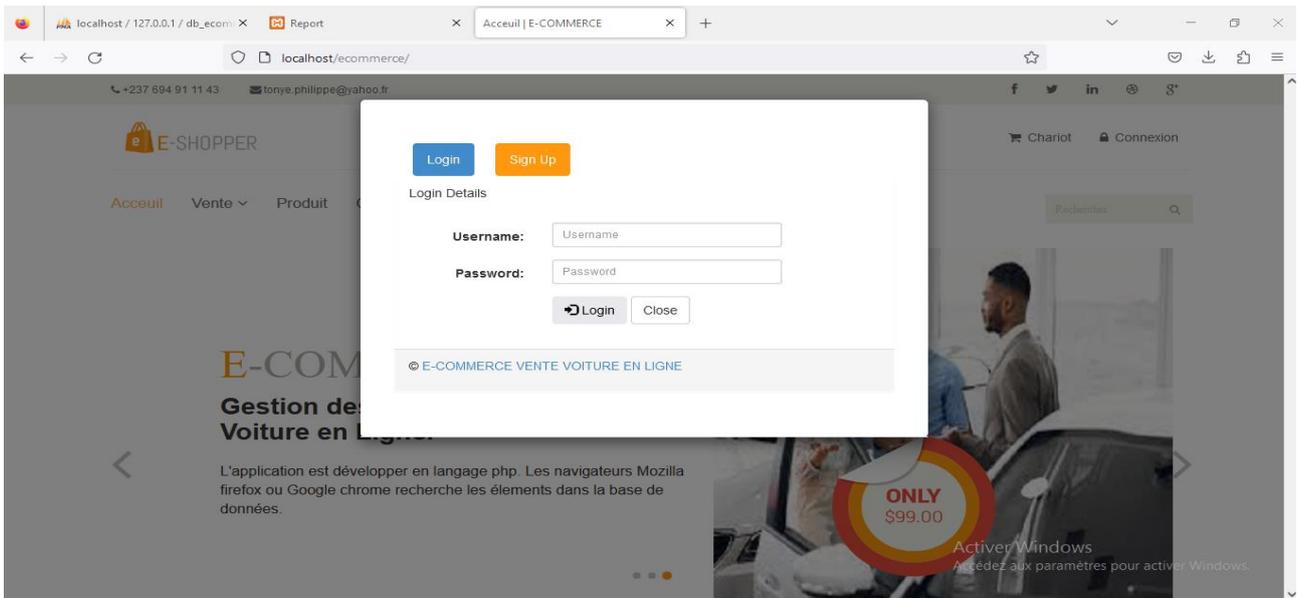
Show 10 entries Search:

#	Commande#	Client	Date de la Commande	Prix	Mode de paiement	Statut	Action
1	98	nawal nawal	Jun/22/2023 05:44:47	Fcfa 38,500,000.00	Cash on Delivery	Pending	<a href="#">Cancel</a> <a href="#">Confirm</a>
2	96	tata toto	Jun/22/2023 12:27:11	Fcfa 300,000,000.00	Cash on Delivery	Confirmed	<a href="#">Confirmed</a>
3	95	tata toto	Jun/21/2023 08:54:09	Fcfa 1,548.00	Cash on Delivery	Confirmed	<a href="#">Confirmed</a>
4	94	Annie Paredes	Aug/21/2019 06:27:09	Fcfa 269.00	Cash on Delivery	Confirmed	<a href="#">Confirmed</a>
5	93	Annie Paredes	Aug/21/2019 06:24:24	Fcfa 0.00	Cash on Delivery	Confirmed	<a href="#">Confirmed</a>

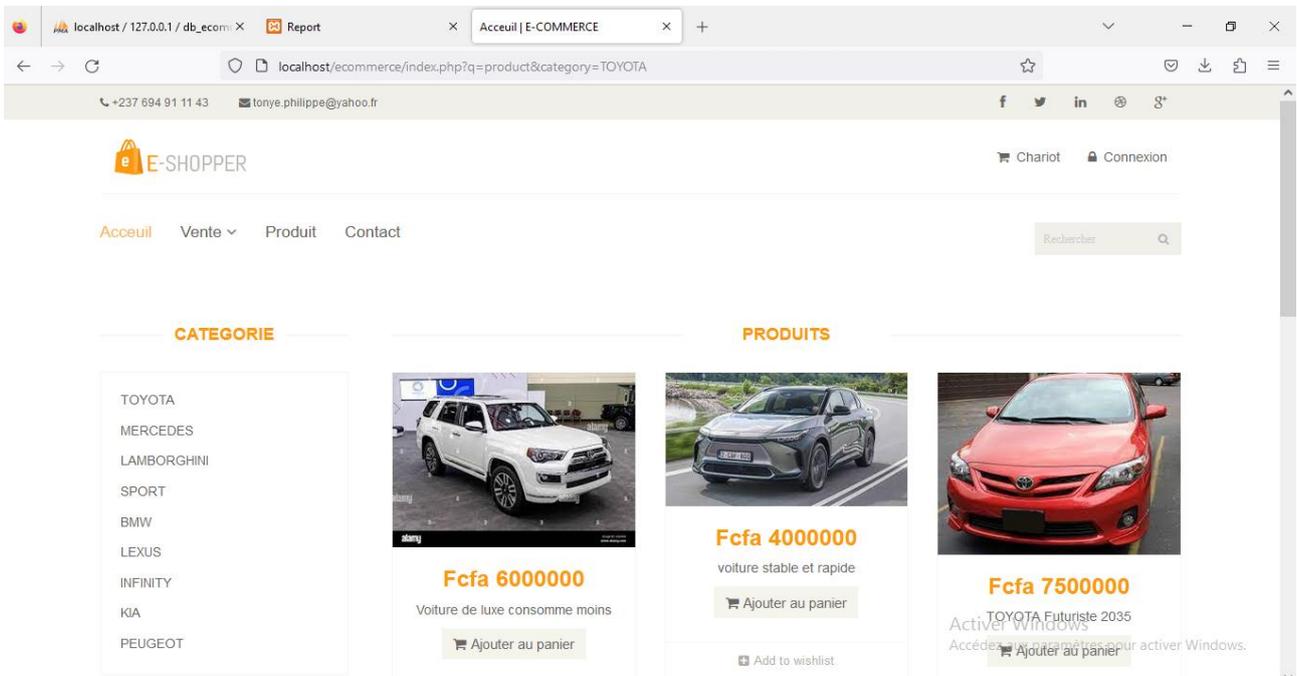
Showing 1 to 5 of 5 entries

Previous 1 Next

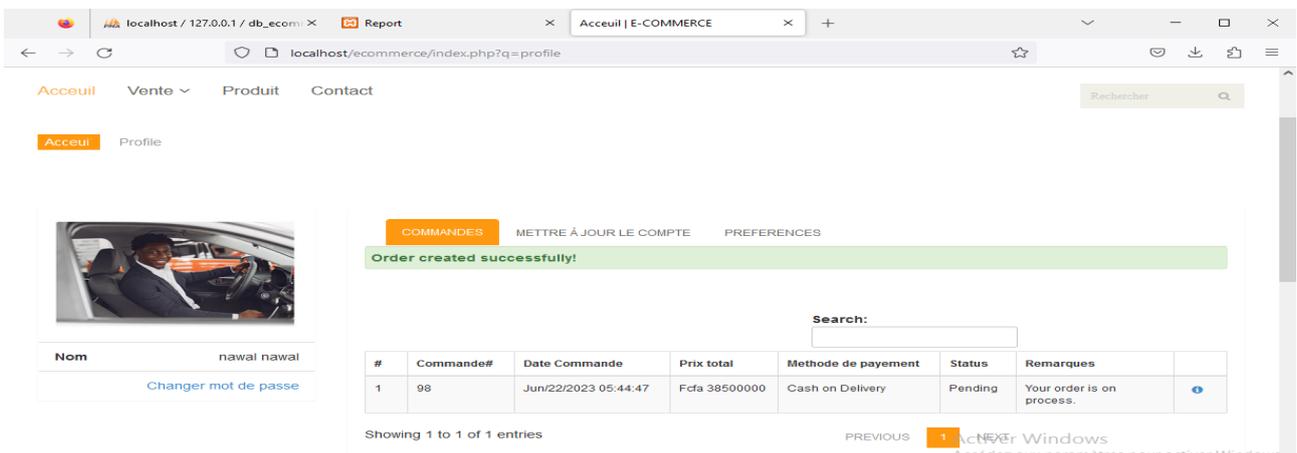
## Interfaces connexion client :



## Interfaces choix voiture par catégories



## Interfaces Préférences (Réservation commande)



## Interfaces reçu bon commande :



Your order is on process. Please check your profile for notification of confirmation.

### Order Information

PRODUCT	PRICE	QUANTITY	TOTAL PRICE
TOYOTA Futuriste 2035	Fcfa 7,500,000.00	1	Fcfa 7,500,000.00
Voiture de luxe consomme moins	Fcfa 6,000,000.00	1	Fcfa 6,000,000.00
LAMBORGHINI Futuriste	Fcfa 25,000,000.00	1	Fcfa 25,000,000.00

Ordered Date : Jun/22/2023 05:44:47  
Payment Method : Cash on Delivery

Total Price : Fcfa 38,500,000.00  
Delivery Fee : Fcfa 0.00  
Overall Price : Fcfa 38,500,000.00

Close

Print

Activer v  
Accédez au

## 9. CONCLUSION GENERALE

Rendu au terme de notre projet académique INF3036, effectué au sein de la structure université Yaoundé I du 18 juin au 22 juin 2023. Nous pouvons dire que ce projet a été pour nous non seulement un moyen de nous familiariser avec le milieu professionnel, mais aussi un gain en termes de connaissances pratiques sur la gestion d'un projet logiciel. Il y en ressort que la vente de voiture nécessitait d'être informatisé de par des limites liées à une gestion manuelle. Nous avons pu ainsi ressortir les besoins fonctionnels et non fonctionnels du futur système et avons mené une analyse basée sur le langage PHP. Par la suite nous avons choisi les outils de réalisation de notre application qui sont MySQL comme serveur de base de données et avons abordé la phase d'implémentation de notre projet. Ceci ayant pour objectif de répondre aux besoins énoncés dans le rapport.

Dès lors, nous ne pourrions avoir la prétention de dire que nous avons abordé tous les aspects d'une gestion de vente en ligne.

## 10. PERSPECTIVES DE DEVELOPPEMENT

En guise de perspective, ce travail reste ouvert à toutes améliorations envisageables :

- ❖ Terminer la phase de réalisation et mettre en place notre logiciel.
- ❖ Ouvrir le champ d'étude sur d'autres structures afin que cette application soit exploitable par ceux-ci et puisse répondre à d'autres besoins que nous n'avons pas spécifiés.
- ❖ Terminer la phase de correction et visualisation des résultats.