

Filière Sciences Mathématiques et Informatique (SMI)

**MEMOIRE DE FIN D’ETUDES**

*Réalisé par :*

**SALIMI ACHRAF & DONGMO**

**KENGMO DUCLAIR**

****



**Intitulé**



|  |  |
| --- | --- |
| **Pr. Laassiri Jalal    Examinateur** | *Faculté des Sciences, Kénitra* |
| **Pr. Alami Chentoufi Jihane Examinatrice** | *Faculté des Sciences, Kénitra* |
| **Pr. Enneya Nourddine    Encadrant** | *Faculté des Sciences, Kénitra* |



Remerciement

Louange à ALLAH qui, par sa grâce infinie, nous a permis de conduire ce modeste travail.

Nous adressons nos remerciements principalement à nos enseignants. Nous avons l'honneur d'être parmi vos élèves et de bénéficier de votre riche enseignement. Vos qualités pédagogiques et humaines sont pour nous un modèle.

Nous tenons à exprimer nos remerciements les plus vifs et sincères à notre encadrant, Pr. ENNEYA Nourddine, pour l'expérience enrichissante qu'il nous a fait vivre durant ce semestre. Il nous a transmis son expertise dans le domaine et nous a consacré du temps tout au long de cette période, satisfaisant notre curiosité.

Nos remerciements les plus sincères vont également à toutes les personnes qui nous ont aidés de près ou de loin à réaliser ce travail et qui ont apporté leur soutien continu.

Tables des matières

Chapitre 1 : Introduction générale ………………………………………

* Introduction …………………………………………………………….

1. Objectifs ………………………………………………………….
2. La cible ……………………………………………………………
3. Les admins……………………………………………………….
4. Le fonctionnement…………………………………………….

* Problématique …………………………………………………………
* Solution proposée ……………………………………………………
* Conclusion …………………………………………………………….

Chapitre 2 : Etudes préalable ……………………………………………

* Introduction…………………………………………………………….
* Spécialisation des besoins……………………………………….

1. Besoins fonctionnels…………………………………………
2. Besoins non fonctionnels………………………………….

* Descriptions générales du système…………………………….
* Conclusions…………………………………………………………….

Chapitre 3 : Etudes Conceptuelle……………………………………….

* Introduction ……………………………………………………………
* Conception de la Base de Données ……………………………
  + 1. Tables Utilisateurs (USER) …………………………
    2. Tables Offres (OFFERS) …………………………….
    3. Tables Commandes (ORDERS) ……………………
    4. Tables Détails des Commandes (OrderOffers)…
    5. Tables Panier (CART) …………………………………
    6. Tables Détails du Panier (CartOffers)…………
    7. Tables Abonnements (SUBSCRIPTION) ………
    8. Tables Commentaire (COMMENT)……………….
* Diagrammes UML……………………………………………………
  1. Diagramme de Cas d'Utilisation ……………………….
  2. Diagramme de Séquence …………………………………….
  3. Diagramme d'Activité ………………………………………….
  4. Diagramme descriptif de SIGN UP (Diagramme de flux )…
* Conclusion …………………………………………………………

Chapitre 4 : Réalisation……………………………………………………….

* Introduction ………………………………………………………………
* Les Besoins Technique
  1. Les langues et les bibliothèques
  2. Les outillés de développement
* Présentation des Pages principales …………………………….
* Scénarios de Fonctionnalités……………………………………….
  + 1. Gestion des Abonnements
    2. Gestion des Offres
    3. Gestion des Commandes
    4. Gestion des Utilisateurs
    5. Gestion des commentaires
* Détails des Fonctions Python…………………………………….
  + 1. Gestion des Images et Fichiers
    2. Gestion de numéro de téléphone
    3. Envoie des emails
* Assurance qualité et optimisation……………………………….
* Conclusion…………………………………………………………………

Chapitre 5 : Conclusion ………………………………………………………

* Conclusion Générale ..........................
* Webographie .......................

Listes des Figures :

[Figure 1 : diagramme de cas utilisation](about:blank) 20

[Figure 2 : diagramme de sequence](about:blank) 21

[Figure 3 : diagramme de classes](about:blank) 23

[Figure 4 : DIAGRAMME DE FLUX](about:blank) 24

[Figure 5 : menu](about:blank) 32

[Figure 6 : footer](about:blank) 32

[Figure 7 : home page fig1](about:blank) 33

[Figure 8 : home page fig2](about:blank) 33

[Figure 9 : HOME PAGE FIG3](about:blank) 33

[Figure 10 : offres 34](#_heading=h.7g3045kep1r0)

[Figure 11 : description d'offre 35](#_heading=h.z25iajxqlftw)

[Figure 12 : section commentaires 35](#_heading=h.t1hqh8ng1qje)

[Figure 13 : formulaire de contact de la page](about:blank) 36

[Figure 14 : list des contacts](about:blank) 36

[Figure 15 : presentation de l'application web 36](#_heading=h.fbpiog6ys504)

[Figure 16 : page de panier 37](#_heading=h.2qe3mo5hogn5)

[Figure 17 : formulaire de modification des information personnel](about:blank) 38

[Figure 18 : partie informations personnel](about:blank) 38

[Figure 19 : statistique et operation sur les utilisateurs](about:blank) 39

[Figure 20 : liste des utilisateurs et les commands des clients](about:blank) 39

[Figure 21 : partie operation sur les offres](about:blank) 40

[Figure 22 : partie modification d'un offre](about:blank) 40

[Figure 23 : page employer](about:blank) 41

[Figure 24 : page client](about:blank) 41

[Figure 25 : page sign up 42](#_heading=h.h4pch1ur2ykv)

[Figure 26 : verification d'email](about:blank) 42

[Figure 27 : l'email de verification 43](#_heading=h.gdz0zb2cp0dx)

[Figure 28 : page log in](about:blank) 43

[Figure 29 : page reset password](about:blank) 44

[Figure 30 : page d'envoie d'email de changement de mot de passe](about:blank) 44

[Figure 31 : l'email de changement de mot de passe](about:blank) 44

[Figure 32 : abonnment](about:blank) 45

[Figure 33 : commande 46](#_heading=h.nyvrvgqfftt3)

[Figure 34 : changement de role et suppresion d'utilisateur 46](#_heading=h.8zj8nuuuh5pp)

[Figure 35 : partie commentaires 47](#_heading=h.bfho9wsex3le)

[Figure 36 : les emails de verifications 50](#_heading=h.7ruu3183c2r3)

[Figure 37 : configuration de gmail smtp 50](#_heading=h.m5ma87cyzfxy)

[Figure 38 : template d'evoie d'un email 51](#_heading=h.hvh0t2td6hsx)

*CHAPITRE 1 : Introduction General*

1.1 Introduction

C’est quoi DAMSO\_STREAM ?

DAMSO\_STREAM est une application web innovante dédiée à fournir des services digitaux à des prix extrêmement compétitifs. Conçue pour satisfaire les besoins des clients désirant de s’acquérir des services digitaux, l'application propose une vaste gamme de services tels que Netflix, Prime vidéo, Microsoft 365, Adobe créative cloud, et bien plus encore.

Grâce à une interface conviviale, les utilisateurs peuvent facilement parcourir les différentes offres qui sont disponible, les consulter et lire les avis de d’autres clients. DAMSO\_STREAM intègre des fonctionnalités comme la possibilité de suivre l’évolution de son abonnement, les devis rapides et les paiements sécurisés.

Avec une approche centrée sur la qualité, la transparence et l'accessibilité, DAMSO\_STREAM permet à chacun de bénéficier de services digitaux professionnels à des coûts réduits, facilitant ainsi l'accès à la technologie pour tous.

1. Les objectifs

DAMSO\_STREAM a pour objectifs

* Faciliter l’acquisition des services digital à ses clients
* Offrir les meilleurs services digitaux aux meilleurs des prix

1. La Cible

Cible principale : ouvert au grand public

1. Les Admins (propriétaires)

Les responsables et créateurs de l’application sont KENGMO DONGMO DUCLAIR et ACHRAF SALIMI deux étudiants en science informatique à l’université ibn tofail.

1. Le Fonctionnement

Les clients devront juste :

1. Accéder à l’application.
2. Choisir le service qu’ils veulent acheter.
3. Vérifier les différents prix et plan associer au service.
4. Choisir un mode de paiement et payer en suite les l’accès (login) aux services leurs sera envoyer soit par email, messages WhatsApp ou Instagram.
5. Les clients pourront nous joindre à tout moment sur les contacts que nous mettrons à dispositions dans l’application.

1.2 Les problématiques et solutions proposée

1. **Accessibilité Financière :**
   * **Problème :** Les petites étudiants, les travailleurs se plaignent plus souvent du coup élevé des services digitaux.
   * **Solution :** Un site proposant des services à bas prix permet à ces clients d'accéder à des services professionnels et de divertissement sans dépasser leur budget.
2. **La difficulté de s’acquérir certain service :**
   * **Problème :** Certaines personnes ne savent malheureusement pas comment faire pour crée des

comptes sur certaines plateforme ce streaming ou éducatifs

* + **Solution :** DAMSO\_STREAM intervient en leurs proposant des comptes déjà crée et près à être utiliser directement

1. **Productivité et Efficacité :**
   * **Problème :** Certaines ont souvent besoin de divers services digitaux mais manquent de temps ou de compétences en interne pour les gérer.
   * **Solution :** DAMSO\_STREAM s’occupe de la gestion des comptes à votre place et vous notifie des dattes des différents fin d’abonnements.
2. **Flexibilité et Adaptabilité :**
   * **Problème :** Les besoins peuvent changer rapidement, nécessitant des ajustements fréquents de leurs services digitaux.
   * **Solution :** une panoplie de service est disponibles sur le site et facilement accessible par les utilisateurs.

En résolvant ces problèmes, un site web offrant des services digitaux à des prix très bas peut devenir un allié précieux pour les petites entreprises, les indépendants

1.3 Conclusion

En conclusion, l'introduction d'un site web offrant des services digitaux à des prix très bas, représente une solution innovante et accessible. En répondant aux défis financiers, en comblant les lacunes de compétences, et en simplifiant le processus de commande, ce type de plateforme démocratise l'accès aux services professionnels de haute qualité.

*Chapitre 2 : Etude préalable*

1. Introduction

Dans ce chapitre on va parler sur les différentes fonctionnalités offertes par note application web

1. Spécialisation des besoins

Dans cette partie, on va poser les besoins qu’on veut de les implémenter dans notre système

* 1. Besoins fonctionnels

Dans le cadre de note système en va avoir 3 types d’utilisateurs (rôle) : utilisateur(Client), employé(Employee), administrateur(Admin), voici les fonctionnalités de chaque ces rôles :

**Tous les utilisateurs doivent pouvoir faire  :**

* **Modification d’information Personnel :** L’utilisateur doit pourvoir modifier ces informations personnelles comme, image de profile, nom, prénom, numéro de telephone, email ….
* **Operations sur le compte :** L’utilisateur doit pouvoir de se déconnecter (log out) et le droit de supprimer son propre compte personnel .

**Administrateur :**

* **Gestion des utilisateurs :**L’administrateur doit pouvoir modifier le rôle d’un autre utilisateur vers (Client, Employee, Admin) mais il ne peut pas changer le rôle d’un autre admin , et aussi il doit avoir le droit de supprimer un utilisateur mais un admin ne peut pas supprimer le compte d’un autre admin pour des raison de sécurité.
* **Gestion des offres :** L’administrateur doit pouvoir ajouter dans le panier des offres, envoyez des commandes, Ajouter, modifier et supprimer les offres.
* **Gestion des commandes :** L’administrateur doit pouvoir peux accéder aux commandes envoyez par les utilisateurs et le traiter.
* **Voir les statistiques :** L’administrateur doit pouvoir accède au donner statistique comme le nombre de totale de revenu, nombre de commandes, nombre de clients, doit avoir accède à la liste des utilisateur (chaque rôle a sa propre liste)(liste admin, liste employé, liste client ).
* **Gestion de commentaires :**L’administrateur doit pourvoir supprimer les commentaires de tous les utilisateurs .

**Employé :**

* **Gestion des commandes :** L’employé doit pouvoir accéder aux commandes envoyez par les utilisateurs et le traiter mais il ne peut pas envoyez une commande, pour des raisons de sécurité.

**Client :**

* **Voir les statistiques des abonnements de de la commande :** Le client doit pouvoir voir le statut de ces commandes, ainsi que le statut de ces abonnements.
* **Envoyez des commandes :** Le client doit pouvoir envoyez des commandes a partir de sa propre panier personnel.
* **Opérations sur les abonnements :** Le client doit pouvoir de supprimer l’abonnement inactive (dépasser la date finale), ou renouvelez son abonnement.
  1. Besoins non fonctionnels

Les besoins non fonctionnels sont Le comportement et la performance que l’application web doit avoir

**Sécurité :** Stocker les fichiers téléchargés de manier sécuriser, éviter les fichiers exécutables ;

**Performance** : la vérification doit être plus vite, c’est-à-dire ne pas recharger toute la page juste pour l’affichage d’une erreur au niveau d’insertion de donner.

**Intégrité des données :** Les donnes avant de les stocker dans la base de donner doivent être bien vérifier, au niveau de format, type, ou l’existence dans la base de donner (pour éviter les duplication non nécessaire) par exemple duplication des email ( doivent être interdit).

On a aussi la Vérification des contraints donner avant l’insertion est aussi un **aspect d’Intégrité** **des données** par exemple le mot de passe doivent vérifier plusieurs contraint avant de l’accepter pour avoir un mot de passe sécuriser.

1. Description générale du système

Le système en bref est un système qui permet a le client d’avoir acheté des services digitaux online, en utilisant cette plateforme en facilite le contacte entre le client et l’employé ou l’admin, et aussi faciliter l’achat, c’est juste un envoie d’une commande, et l’admin ou l’employé qui va vous contacter, il facilite aussi la présentation des offres, pour l’admin et le client, il facilite l’ajout la suppression et la modification des offres pour l’admin, et pour le client il peut justement chercher d’une manier plus vite et confortable de l’offre qu’il aime et l’envoyez sous forme d’une commande à travers son panier personnel.

1. Conclusion

Dans le chapitre 2 on a bien détailler la partie des besoin fonctionnel et non fonctionnel et bien expliquer en bref l’aider général de notre projet

*CHAPITRE 3 : Etudes Conceptuel*

1. Introduction

Dans la partie conception de base de données on a pensé de crée un système qui géré la partie des utilisateurs en utilisant la table USER et COMMENT, et une autre partie de gestion tous ce qui concerne les offres, en utilisant les tables OFFERS, ORDERS, ORDEROFFERS, CART, CARTOFFERS, SUBSCRIPTION.

Et dans la partie de conception UML on va vous présenter comment notre system fonctionne et comment les opérations suivantes fonctionne aussi :

1. SIGNUP, LOGIN, LOGOUT, DELETE (le compte),
2. Modifier les informations d’utilisateur,
3. Récupération de mot de passe oublier,
4. Envoyer une commande,
5. Option de renouveler les abonnements, ou les supprimer après la fin de dure d’abonnement,
6. Ajouter dans le panier des offres avec l’option de supprimer les choses qui existe dans le panier,
7. (Ajouter, modifier, supprimer) les offres,
8. Supprimer, changer le rôle d’un Usr,
9. Recevoir les commandes pour le traiter,
10. Voire les stats de l’application web (revenue, les commandes, nbr de clients),
11. Ecrire et supprimer les commentaires sur un offres,
12. Envoie des emails (partie contact Us).
13. Conception de la Base de Données
    1. Table Utilisateurs (USER)

La table USER contient toutes les informations concernant l’utilisateur ces informations venant de SIGN UP page, cette table est de la forme suivante :

create table USER(

idCli integer NOT NULL AUTO\_INCREMENT,

Email char(60),

Password char(60) ,

FirstName char(32) ,

LastName char(32) ,

status char(30) DEFAULT 'unverified' ,

role char(30) DEFAULT 'client',

Date\_Joined DATETIME ,

whatsapp char(30),

pfpName char(255) DEFAULT 'user583abc\_1649114257.png',

primary key (idCli)

);

1. IdCli c'est la clé primaire et aussi contient la contrainte AUTO\_INCREMENT pour que la clé se crée automatiquement .
2. Status est de la valeur par défaut : 'unverified' prend aussi la valeur ('verified').
3. Role est de valeur par défaut 'client' prend aussi les deux valeurs ('employee', 'admin').
4. PfpName c'est le nom l'image de profil, est de valeur par défaut 'user583abc\_1649114257.png' qui est le nom de l'image par défaut si l'user n'a pas changé l'image de profil à partir de ça page personnel (User\_Page).
   1. Table Offres (OFFERS)

La table OFFERS contient les offres avec tous les informations nécessaires donnez par les utilisateurs qui ont le rôle 'admin’, cette table est de la forme suivante :

CREATE TABLE OFFERS(

idOffer INT AUTO\_INCREMENT,

description TEXT,

image\_Name VARCHAR(255) DEFAULT 'default-product-image.png',

Offer\_price DECIMAL(10, 2),

name VARCHAR(255),

duration INT,

PRIMARY KEY (idOffer)

);

1. IdOffer le primaire et aussi contient la contrainte AUTO\_INCREMENT pour que la clé se crée automatiquement.
2. Description : la description de l'offre.
3. Image\_Name : pour stocker le nom de l'image de l'offre, et aussi a de valeur par défaut 'default-product-image.png' si l'admin ne saisir pas la partie qui concerne l'image de l'offre.
4. Duration : La durée de l'offre (en jours).
   1. Table Commandes (ORDERS)

La table ORDERS contient les informations sur les offres choisies par les clients et envoyées à la partie de la page (Cart), cette table est de la forme suivante :

CREATE TABLE Orders (

idOrder INT AUTO\_INCREMENT,

idCli INT,

StatOfTreatment VARCHAR(50) DEFAULT 'Pending Treatment',

PaymentStat VARCHAR(50) DEFAULT 'Pending Payment',

TotalPrice DECIMAL(10, 2),

PRIMARY KEY (idOrder),

FOREIGN KEY (idCli) REFERENCES USER(idCli)

);

1. IdOrder est une clé primaire et aussi contient la contrainte auto\_increment.
2. IdCli est une clé étrangère qui référence idCli de la table USER.
3. StatOfTreatment qui décrit le statut de traitement de la commande (ordre) avec une valeur par défaut 'Pending Treatment' et aussi prend la valeur 'treated'.
4. PaymentStat qui décrit le statut de payement de la commande avec une valeur par défaut 'Pending Payment' et aussi prend la valeur 'Payed'.
5. TotalPrice c'est la somme des prix des offres de la commande, qui a les id de ces offres stocker dans la table orderOffers.
   1. Table Détails des Commandes (OrderOffers)

La table OrderOffers c'est pour stocker les offres de la commande envoyez par les clients, cette table est de la forme suivante :

CREATE TABLE OrderOffers (

OrderOffers\_Id INT AUTO\_INCREMENT,

idOrder INT,

idOffer INT,

PRIMARY KEY (OrderOffers\_Id),

FOREIGN KEY (idOrder) REFERENCES Orders(idOrder),

FOREIGN KEY (idOffer) REFERENCES OfferS(idOffer)

);

1. OrdeOffer\_Id : est une clé primaire et aussi contient la contrainte auto\_increment.
2. IdOrder est une clé étrangère qui référence idOrder de la table Orders.
3. IdOffer est une clé étrangère qui référence idOffer de la table Offers.
   1. Table Panier (CART)

La table CART c’est pour stocker la codépendance de (client – cart) pour que tous les clients ayant leur propre (cart), cette table est de la forme :

CREATE TABLE CART (

idCart INT AUTO\_INCREMENT PRIMARY KEY,

idCli INT,

fullPrice DECIMAL(10, 2),

FOREIGN KEY (idCli) REFERENCES USER(idCli)

);

1. idCart est une clé primaire et aussi contient la contrainte auto\_increment.
2. idCli est une clé étrangère qui référence idCli de la table USER.
3. fullPrice contient la somme des prix de toutes les offres dans la table cartOffers.
   1. Table Détails du Panier (CartOffers)

La table CartOffers créée pour stocker temporairement les offres choisies par les clients, car à chaque envoie de commande (ORDER) cette table est vidée toutes les infos dans la table cart qui correspond à ce client qui a envoyé cette commande, cette table est de la forme :

CREATE TABLE CartOffers (

CartOffer\_id INT AUTO\_INCREMENT ,

idCart INT,

idOffer INT,

PRIMARY KEY (cartOffer\_id),

FOREIGN KEY (idCart) REFERENCES CART(idCart),

FOREIGN KEY (idOffer) REFERENCES OFFERS(idOffer)

);

1. CartOffer\_id est une clé primaire et aussi contient la contrainte auto\_increment.
2. idCart est une clé étrangère qui référence idCart de la table Cart, c'est pour faire la correspondance de la (CART) avec les offres choisies dans (CART)
3. idOffer est une clé étrangère qui référence idOffer de la table Offers, pour avoir âpres les informations des offres choisies en utilisant la jointure entre cette table et la table Offers.
   1. Table Abonnements (SUBSCRIPTION)

La table SUBSCRIPTION c’est pour stocker les informations d'abonnement des clients, cette table est de la forme suivante :

CREATE TABLE Subscription (

idSubscription INT AUTO\_INCREMENT,

idCli INT,

idOffer INT,

subscriptionStatus VARCHAR(50) DEFAULT 'Active',

startDate DATE,

endDate DATE,

PRIMARY KEY (idSubscription),

FOREIGN KEY (idOffer) REFERENCES OFFERS(idOffer),

FOREIGN KEY (idCli) REFERENCES USER(idCli)

);

1. idSubscription est une clé primaire et aussi contient la contrainte auto\_increment.
2. idCli est une clé étrangère qui référence idCli de la table USER.
3. idOffer est une clé étrangère qui référence idOffer de la table Offers .
4. subscriptionStatus : c'est l'état de l'abonnement est de valeur par défaut 'Active' et aussi peux prendre la valeur 'Inactive' quand la durée de l’abonnement est finie.
5. startDate : c'est la date quand l'abonnement commence.
6. endDate : c'est la date quand l'abonnement finit.
   1. Table Commentaire (COMMENT)

La table COMMENT c’est pour stocker à chaque offre ces propre commentaires , chaque utilisateur peux écrire plusieurs commentaires et chaque offre a une seule section de commentaires, cette table est de la forme :

CREATE TABLE COMMENT (

idComment integer AUTO\_INCREMENT,

idOffer integer NOT NULL,

idCli integer NOT NULL,

comment\_TEXT VARCHAR(10000) NOT NULL,

Date\_written DATE,

PRIMARY KEY(idComment),

FOREIGN KEY (idOffer) REFERENCES offers(idOffer),

FOREIGN KEY (idCli) REFERENCES user(idCli)

);

1. idComment est une clé primaire et aussi contient la contrainte auto\_increment.
2. idCli est une clé étrangère qui référence idCli de la table USER.
3. idOffer est une clé étrangère qui référence idOffer de la table Offers .
4. comment\_TEXT c’est le commentaire saisi par l’utilisateur, ce commentaire est taille max 10 000.
5. Date\_written c’est quand l’utilisateur a écrit ce commentaire.
6. Diagrammes UML :

Pour concevoir un modèle UML, nous commencerons par les éléments suivants :

1. **Diagramme de Cas d'Utilisation (Use Case Diagram)** : Cela montrera les interactions de différents types d'utilisateurs (clients, employé, administrateurs) avec le système.
2. **Diagramme de Séquence (Séquence Diagram)** : Cela expliquera comment les processus se déroulent dans le temps pour un scénario spécifique, comme l'achat d'un abonnement par un client.
3. **Diagramme de Classes (Class Diagram)** : Cela décrira la structure des classes et leurs relations qui forment notre système backend.

**Diagramme de Cas d'Utilisation**

**Acteurs :**

* **Client** : Peut créer un compte, se connecter, consulter les offres, ajouter des abonnements au panier, et effectuer des paiements.
* **Employé**: Choisi par l’administrateur, reçoit des droits sur le site lui permettant d’effectuer certaines taches privilégier sur le site .
* **Administrateur** : A le contrôle total sur le site, peut ajouter ou modifier des offres, gérer les comptes utilisateurs, et visualiser des rapports de vente.

**Cas d'utilisation principaux :**

* **Sign up** (Client)
* **Log In** (Client)
* **Consulter les offres** (Client)
* **Ajouter au panier** (Client)
* **Envoyer une commande**
* **Payer** (Client)
* **Ajout d’un abonnement (**Employé / administrateur**)**
* **Traitement de commande (**Employé/ administrateur**)**
* **Control des clients et employés** (Administrateur)
* **Operations sur les offres** (Administrateur)
* **Visualiser les rapports** (Administrateur)

Je vais créer un diagramme de cas d'utilisation simple pour illustrer ces interactions.



### Diagramme de Séquence

Nous pouvons créer un exemple de diagramme de séquence pour le scénario où un client sélectionne une offre et procède au paiement. Ce diagramme montrera les étapes suivies par le client de la connexion à la finalisation de l'achat, ainsi que les interactions avec le système backend.

Dans ce scénario, un client visite le site et décide de créer un compte utilisateur. Lors de la création de son compte, le client remplit un formulaire de registration. Le système entame alors une série de vérifications pour s'assurer de la validité des informations fournies. Il vérifie d'abord si le formulaire est entièrement complété. Ensuite, il contrôle le format du mot de passe, du courriel, et du numéro de téléphone, en s'assurant que le code pays est inclus si un numéro de téléphone est spécifié. Le système vérifie également si le courriel utilisé existe déjà dans la base de données. Si toutes ces vérifications sont satisfaites, le compte est créé, et le client est invité à vérifier son courriel en cliquant sur un lien envoyé par le système. Si l'email a déjà été vérifié, le système redirige vers une page d’erreur ; sinon, il dirige le client vers une page confirmant la vérification du compte.

**Diagram de Classes**

**Classes Principales**

1. **User**
   * **Attributs** : userID, username, password, email, phoneNumber, isEmailVerified
   * **Méthodes** : login(), verifyEmail(), updateProfile()
2. **RegistrationForm**
   * **Attributs** : username, password, email, phoneNumber, countryCode
   * **Méthodes** : validateForm(), validateEmail(), validatePassword(), validatePhoneNumber()
3. **System**
   * **Méthodes** : sendVerificationEmail(), redirectToErrorPage(), redirectToSuccessPage()
4. **Data base**
   * **Méthodes** : checkUserExists(), addUser(), getUserInfo()

**Relations**

* **User <> RegistrationForm** : Une interaction où l'objet User est créé et initialisé basé sur les données validées fournies par RegistrationForm.
* **System > Registration] (Form)** : Le système utilise RegistrationForm pour traiter les données saisies par l'utilisateur et initier les vérifications.
* **System <> Database** : Le système interagit avec la base de données pour vérifier si un utilisateur existe déjà et pour ajouter un nouvel utilisateur après la validation.

**Description des Interactions**

1. **Processus de Création de Compte** : L'utilisateur soumet le formulaire de création de compte via RegistrationForm, qui est ensuite traité par le système.
2. **Vérification et Validation** : Le système utilise RegistrationForm pour effectuer des validations spécifiques (champs non vides, formats valides). En cas de données invalides, le formulaire est renvoyé à l'utilisateur pour correction.
3. **Interaction avec la Base de Données** : Après validation, le système interroge la Database pour s'assurer que l'email n'existe pas déjà. Si la vérification est positive, les informations de l'utilisateur sont enregistrées dans la base de données.
4. **Finalisation** : Une fois le compte créé, le système envoie un email de vérification. Suivant la réponse de l'utilisateur (lien cliqué dans l'email), il est soit redirigé vers la page de succès, soit vers une page d'erreur si l'email était déjà vérifié.

**Diagramme descriptif de SIGN UP :**

1. Conclusion :

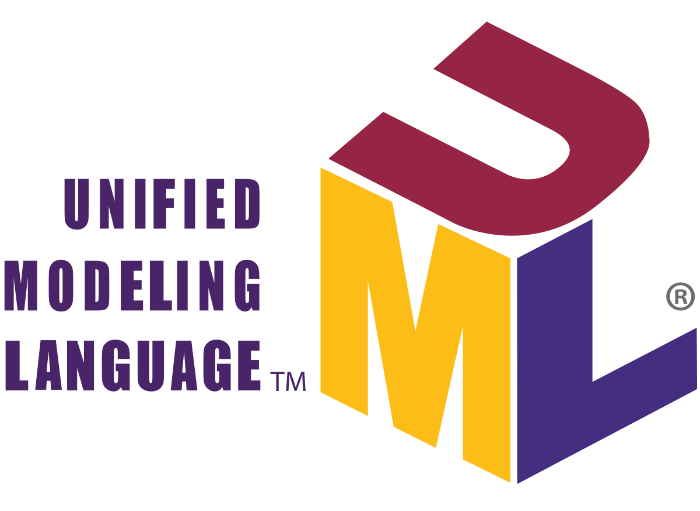
Dans ce chapitre on a parlé de partie conceptionnel de base de donner et conception uml, de notre Projet en expliquer à structure générale et toutes les fonctionnalités qui offre notre application web

*CHAPITRE 4 : Réalisation*

1. Introduction

Dans la partie Réalisation c’est la partie ou on va expliquer le détaille de réalisation de notre Project et comment ces fonctionnalités fonctionne ; On va parler d’abord des pages principales de l’application web et leur fonctionnement ,et après on va un peu détailler d’explication de fonctionnement de ces pages et comment on a les réalisé et après on va trouver un peu d’explication sur l’optimisation d’une partie de notre code.

1. Les Besoins Technique
   1. Les langues et les bibliothèques

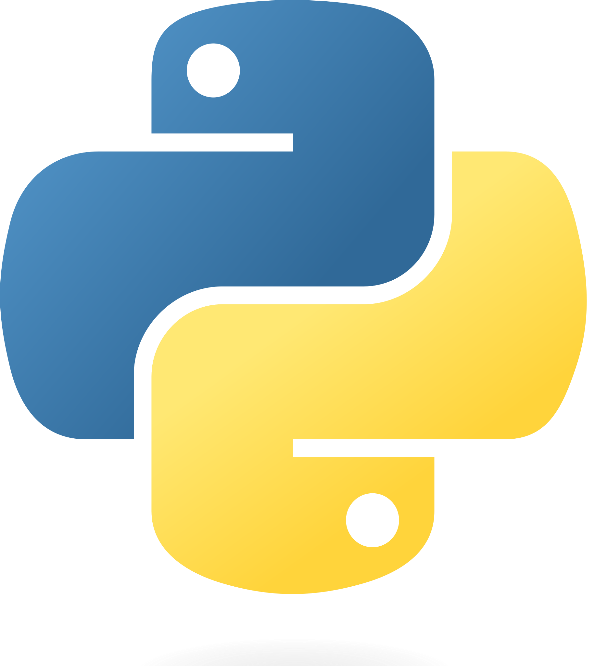


Le langage **UML** (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et

La mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement. L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie.

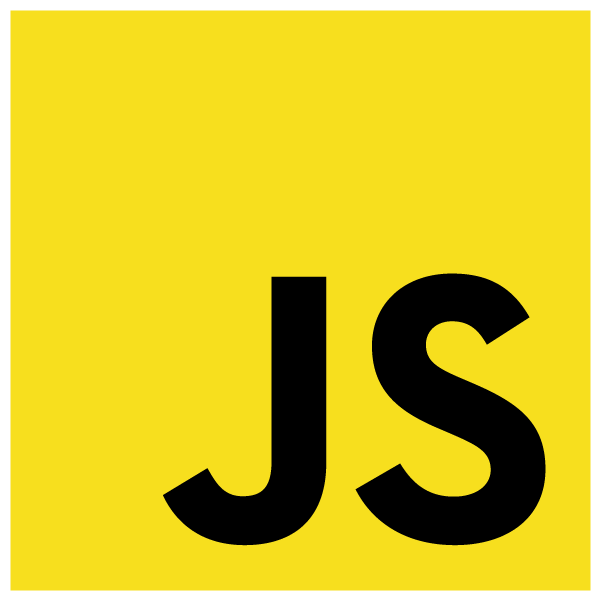
Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet.



**Python** est un langage de programmation largement utilisé dans les applications Web, le développement de logiciels, la science des données et le machine Learning (ML). Les développeurs utilisent Python parce que c'est un langage efficace et facile à apprendre, et qu'il peut s'exécuter sur de nombreuses plateformes différentes. Le logiciel Python peut

être téléchargé gratuitement, il s'intègre bien avec tous les types de systèmes et accélère le développement.



**JavaScript** (« JS » en abrégé) est un langage de programmation dynamique complet qui, appliqué à un document HTML, peut fournir une interactivité dynamique sur les sites Web. Il a été inventé par Brendan Eich, co-fondateur du projet Mozilla, de la Mozilla Foundation et de la Mozilla Corporation. JavaScript est d'une incroyable flexibilité. Vous pouvez commencer petit, avec des carrousels, des galeries d'images, des variations de mises en page et des réponses aux clics de boutons. Avec plus d'expérience, vous serez en mesure de créer des jeux, des graphiques 2D et 3D animés, des applications complètes fondées sur des bases de données et bien plus encore !

L'HyperText Markup Language, **HTML**, désigne un type de langage informatique descriptif. Il s'agit plus précisément d'un format de données utilisé dans l'univers d'Internet pour la mise en forme des pages Web. Il permet, entre autres, d'écrire de l'hypertexte, mais aussi d'introduire des ressources multimédias dans un contenu.

Développé par le [W3C](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203239-w3c-world-wide-web-consortium-definition-traduction/) (World Wide Web Consortium) et le WHATWG (Web Hypertext Application Technology Working Group), le format ou langage **HTML** est apparu dans les années 1990. Il a progressivement subi des modifications et propose depuis 2014 une [version **HTML5** plus aboutie](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203257-html5-hypertext-markup-langage5-definition-traduction/).

L'**HTML** est ce qui permet à un créateur de sites Web de gérer la manière dont le contenu de ses pages Web va s'afficher sur un écran, via le navigateur. Il repose sur un [système de balises](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203619-balise-definition-traduction/) permettant de titrer, sous-titrer, mettre en gras, etc., du texte et d'introduire des éléments interactifs comme des images, des liens, des vidéos... L'HTML est plus facilement compris des robots de crawl des moteurs de recherche que le language [JavaScript](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203585-javascript/), aussi utilisé pour rendre les pages plus interactives.



Le **CSS** pour Cascading Style Sheets, est un langage informatique utilisé sur Internet pour la mise en forme de fichiers et de pages [HTML](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/). On le traduit en français par feuilles de style en cascade.

Apparu dans les années 1990, le **CSS** se présente comme une alternative à la mise en forme via des balises, notamment HTML. Un peu plus complexe à maîtriser, il permet un gain de temps considérable dans la mise en forme d'une page web par

rapport à ces balises. Grâce au **CSS**, vous pouvez en effet appliquer des règles de mise en forme (titrage, alignement, polices, couleurs, bordures, etc.) à plusieurs documents simultanément.

Sur le plan de la conception d'une [page Web](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203265-page-web-definition/), le **CSS** permet par ailleurs de séparer la présentation d'une page HTML et sa structure. Ses standards sont définis par le [World Wide Web Consortium (W3C)](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203239-w3c-world-wide-web-consortium-definition-traduction/).

Les Bibliothèque :

Flask : **Flask** est un micro [framework](https://fr.wikipedia.org/wiki/Framework) [open-source](https://fr.wikipedia.org/wiki/Open-source) de développement web en [Python](https://fr.wikipedia.org/wiki/Python_(langage)). Il est classé comme [microframework](https://fr.wikipedia.org/wiki/Microframework) car il est très léger. Flask a pour objectif de garder un noyau simple mais extensible. Il n'intègre pas de système d'authentification, pas de couche d'abstraction de base de données, ni d'outil de validation de formulaires. Cependant, de nombreuses extensions permettent d'ajouter facilement des fonctionnalités. Il est distribué sous [licence BSD](https://fr.wikipedia.org/wiki/Licence_BSD).

Flask\_Mail : **Flask-Mail** est une extension pour Flask qui facilite l'envoi d'e-mails depuis votre application. Elle simplifie le processus d'intégration des fonctionnalités de messagerie, vous permettant ainsi de vous concentrer sur la création de fonctionnalités exceptionnelles pour votre application.

PhoneNumber : Le module **phonenumbers** offre de nombreuses fonctionnalités, telles que la fourniture d'informations de base sur un numéro de téléphone, la validation d'un numéro de téléphone, etc.

MySql Connector : Le module **mysql.connector** fournit la méthode connect qui permet de retourner un objet qui représente la connexion vers la base de données. Vous devez fournir les paramètres host, user et password pour donner l’adresse du SGBDR, le login et le mot de passe de connexion.

DateTime : Le module [**datetime**](https://docs.python.org/fr/3.6/library/datetime.html#module-datetime) fournit des classes pour manipuler de façon simple ou plus complexe des dates et des heures. Bien que les calculs de date et d’heure sont gérés, l’implémentation est essentiellement tournée vers l’efficacité pour extraire des attributs pour les manipuler et les formater pour l’affichage. Pour d’autres fonctionnalités associées, voir aussi les modules [time](https://docs.python.org/fr/3.6/library/time.html#module-time) et [calendar](https://docs.python.org/fr/3.6/library/calendar.html#module-calendar).

WerkZeug: On a utilisé seulement la fonction **secure\_filename** pour récupérer le nom de fichier sans penser a la partie sécurité, cette bibliothèque nous aide dans cette partie, d’éviter les nom exécutable.

AXIOS : **Axios** est une bibliothèque JavaScript open source qui permet de gérer les requêtes HTTP basées sur le protocole XMLHttpRequest. Il peut également être utilisé pour envoyer des données à partir d'un serveur à un autre.

Moteur de Template :

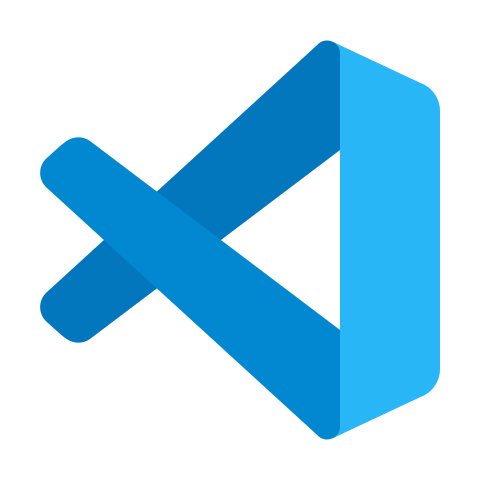
**Jinja** est un moteur de template utilisé par le langage [Python](https://fr.wikipedia.org/wiki/Python_(langage)). Créé par [Armin Ronacher](https://fr.wikipedia.org/w/index.php?title=Armin_Ronacher&action=edit&redlink=1) et distribué sous licence [BSD](https://fr.wikipedia.org/wiki/Licence_BSD), il est très similaire au moteur de template [Django](https://fr.wikipedia.org/wiki/Django_(framework)) mais fournit des expressions Python et évalue les templates dans une [sandbox](https://fr.wikipedia.org/wiki/Sandbox_(s%C3%A9curit%C3%A9_informatique)). C'est un langage orienté texte qui peut ainsi être utilisé pour générer n'importe quel type de fichier pouvant être balisé.

Ce moteur permet certaines personnalisations de tags[2](https://fr.wikipedia.org/wiki/Jinja_(moteur_de_template)#cite_note-Jinja2_Documentation_(2.8-dev)-2), de filtres, de tests, et d'expressions[3](https://fr.wikipedia.org/wiki/Jinja_(moteur_de_template)#cite_note-Jinja2_Documentation_(2.8-dev)_2-3). Cependant contrairement à Django, Jinja permet au développeur d'appeler des fonctions avec des arguments sur des objets. Jinja, comme [Smarty](https://fr.wikipedia.org/wiki/Smarty), utilise également un système de filtre qui se rapproche du [Unix pipeline](https://fr.wikipedia.org/wiki/Pipeline_(Unix)).

Jinja est basé sur le framework [Flask](https://fr.wikipedia.org/wiki/Flask_(framework))[4](https://fr.wikipedia.org/wiki/Jinja_(moteur_de_template)#cite_note-DuPlain_2013_p._30-4) et utilisé par [Ansible](https://fr.wikipedia.org/wiki/Ansible_(logiciel))[5](https://fr.wikipedia.org/wiki/Jinja_(moteur_de_template)#cite_note-redhat-5),[6](https://fr.wikipedia.org/wiki/Jinja_(moteur_de_template)#cite_note-6), Trac, et [Salt](https://fr.wikipedia.org/wiki/Salt_(logiciel)). Il aurait inspiré [Twig](https://fr.wikipedia.org/wiki/Twig), le moteur de template de [PHP](https://fr.wikipedia.org/wiki/PHP).

* 1. Les outillés de développement

**MySQL** (prononcé [[maj.ɛs.ky.ɛl](https://fr.wikipedia.org/wiki/Alphabet_phon%C3%A9tique_international)]) est un [système de gestion de bases de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es) relationnelles (SGBDR). Il est distribué sous une double licence [GPL](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU) et [propriétaire](https://fr.wikipedia.org/wiki/Logiciel_propri%C3%A9taire). Il fait partie des logiciels de gestion de [base de données](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es) les plus utilisés au monde[2](https://fr.wikipedia.org/wiki/MySQL#cite_note-2), autant par le grand public ([applications web](https://fr.wikipedia.org/wiki/Application_web) principalement) que par des professionnels, en concurrence avec [Oracle](https://fr.wikipedia.org/wiki/Oracle_Database), [PostgreSQL](https://fr.wikipedia.org/wiki/PostgreSQL) et [Microsoft SQL Server](https://fr.wikipedia.org/wiki/Microsoft_SQL_Server).



**Visual Studio Code** est un [éditeur de code](https://fr.wikipedia.org/wiki/%C3%89diteur_de_texte) extensible développé par [Microsoft](https://fr.wikipedia.org/wiki/Microsoft) pour [Windows](https://fr.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://fr.wikipedia.org/wiki/Linux) et [macOS](https://fr.wikipedia.org/wiki/MacOS)[3](https://fr.wikipedia.org/wiki/Visual_Studio_Code#cite_note-TechCrunch-3).

Les fonctionnalités incluent la prise en charge du [débogage](https://fr.wikipedia.org/wiki/D%C3%A9bogage), la [mise en évidence de la syntaxe](https://fr.wikipedia.org/wiki/Coloration_syntaxique), la complétion intelligente du code (IntelliSense[4](https://fr.wikipedia.org/wiki/Visual_Studio_Code#cite_note-4).), les [snippets](https://fr.wikipedia.org/wiki/Snippet), la [refactorisation](https://fr.wikipedia.org/wiki/Refactorisation) du code et [Git](https://fr.wikipedia.org/wiki/Git) intégré. Les utilisateurs peuvent modifier le [thème](https://fr.wikipedia.org/wiki/Th%C3%A8me_(informatique)), les [raccourcis clavier](https://fr.wikipedia.org/wiki/Raccourci_clavier), les préférences et installer des [extensions](https://fr.wikipedia.org/wiki/Plugin) qui ajoutent des fonctionnalités supplémentaires.

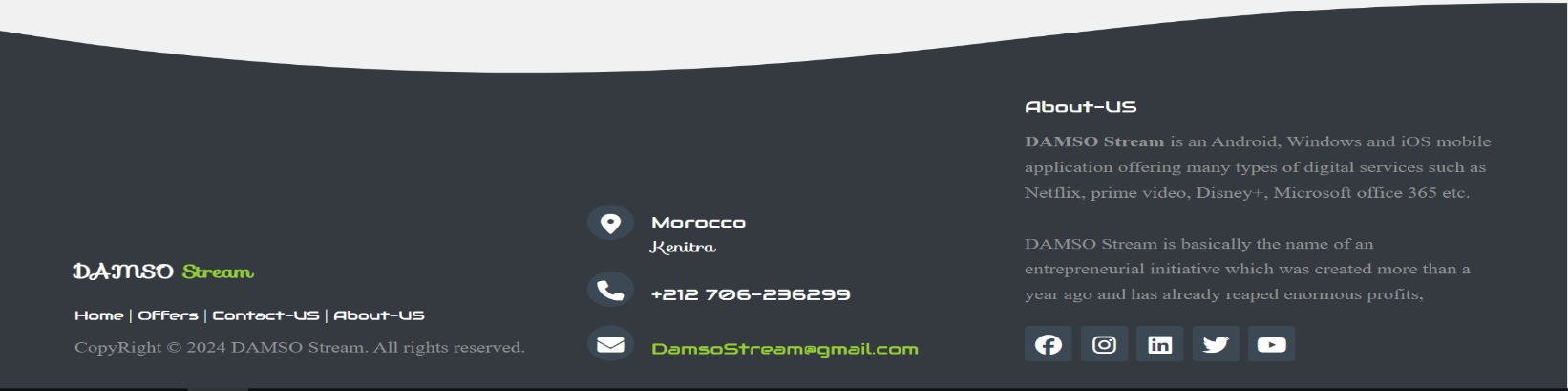
Astah est un outil de modélisation graphique primé permettant de créer des diagrammes UML rapidement et facilement pour les individus et les équipes de toutes tailles. Cela fonctionne sur Windows, Mac, Linux et iOS

1. Présentations des Pages principales

Tous Les Pages Principles sont entendus (Extended) d’une page html qui s’appelle Menu qui contient le menu et le (Footer) en utilisant le Moteur de Template Jinja pour éviter la répétions de ces deux parties.

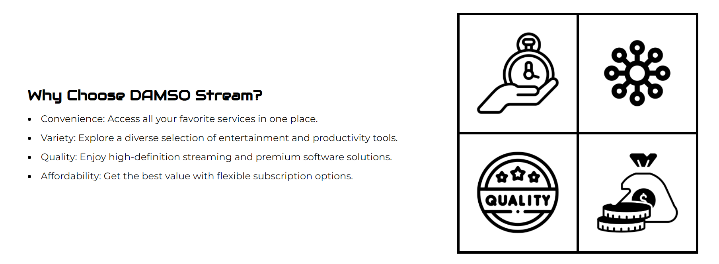






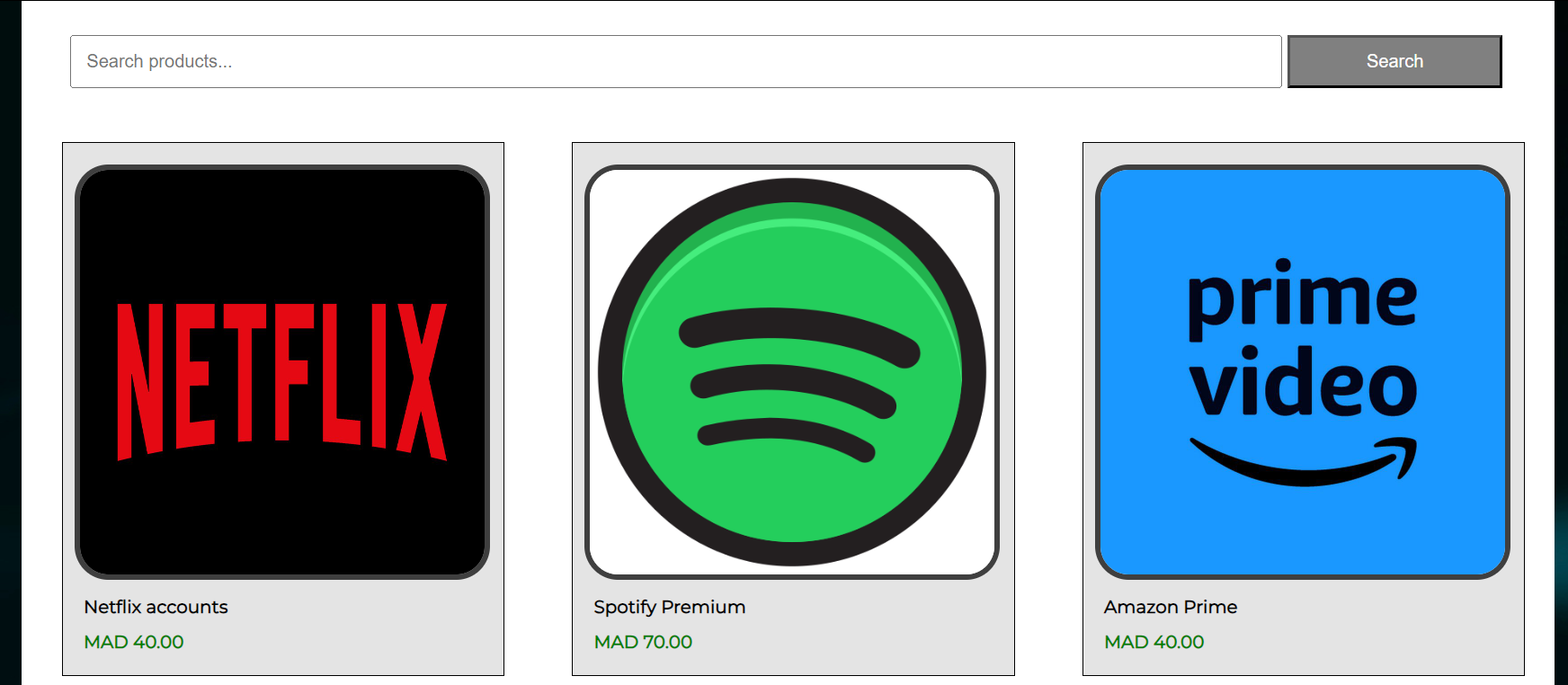
* 1. Page Home :

C'est l'interface de Site Web, c'est la première page l'utilisateur voir donc il faut que cette partie aille être ou ont présenté l'application web et aussi rependre aux questions, quoi offre notre service et pour quoi le choisir ?



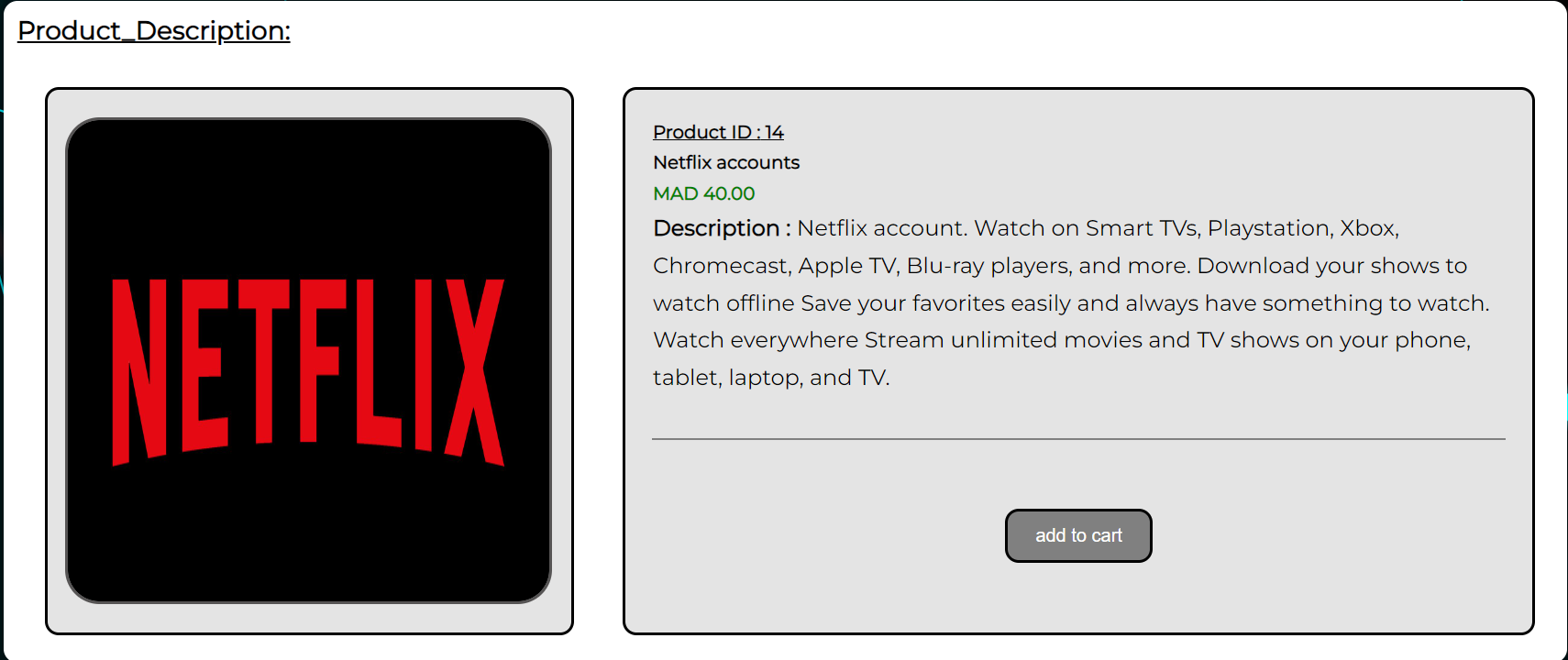
* 1. Page Offers

C’est la partie ou le client trouve les offres qu’il veut, cette partie est dynamique c’est-à-dire quand on fait un changement dans la base de donner, le contenue de la page se change automatiquement aussi, par exemple quand on fait (comme ajout, modification, suppression d’une offre), On a aussi une partie pour la recherche des offres écrit dans cette case.

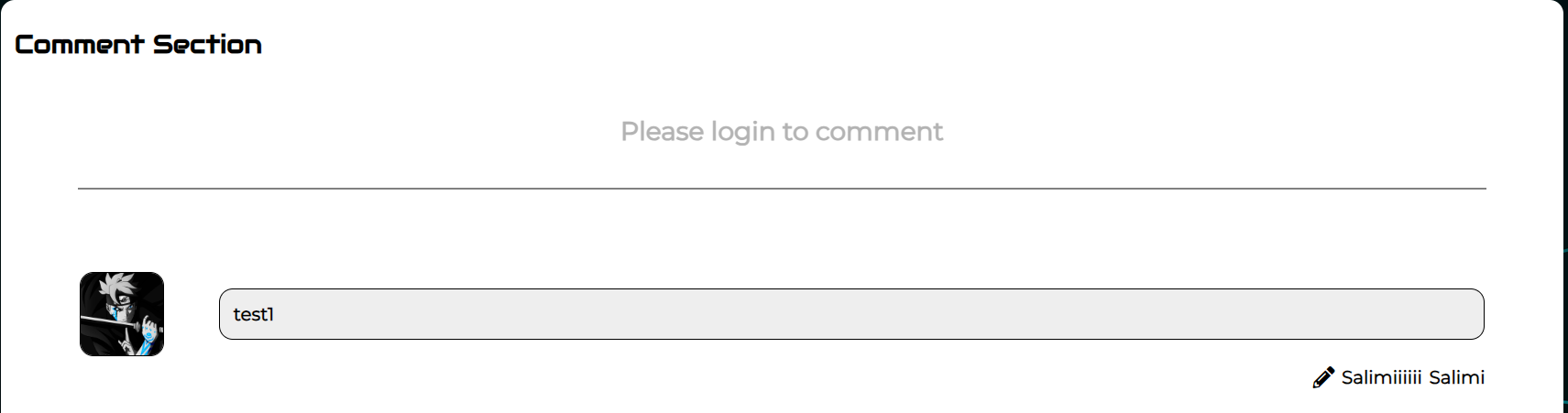


**Figure 10 : offres**

Chaque Offre a son propre page de description ou le client peux commenter là-bas ou ajouter ce produit dans le panier pour l’envoyer a les employés ou les admins pour traiter ça commande.



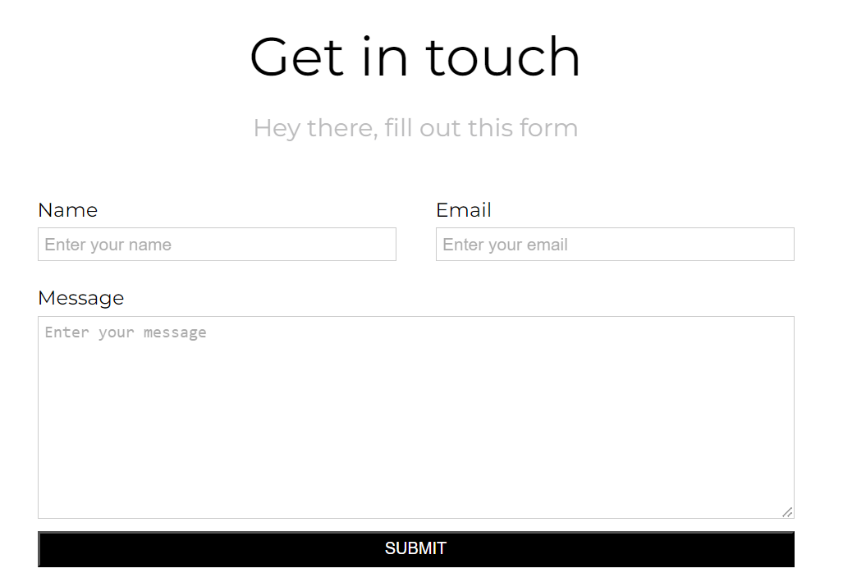
**Figure 11 : description d'offre**

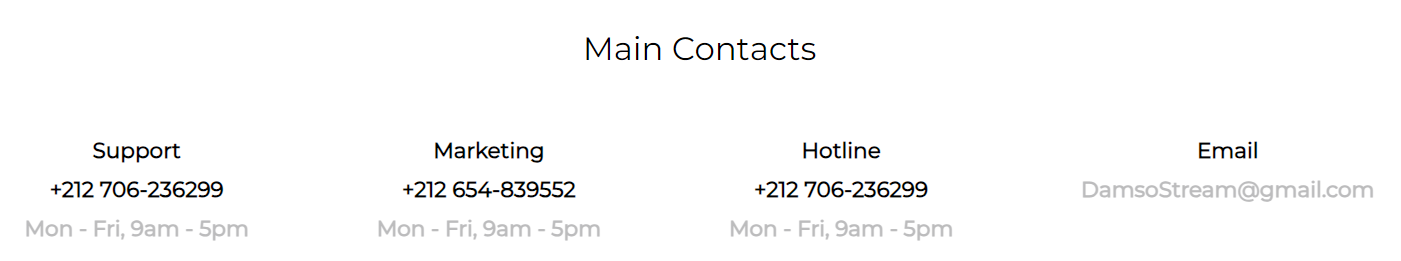


**Figure 12 : section commentaires**

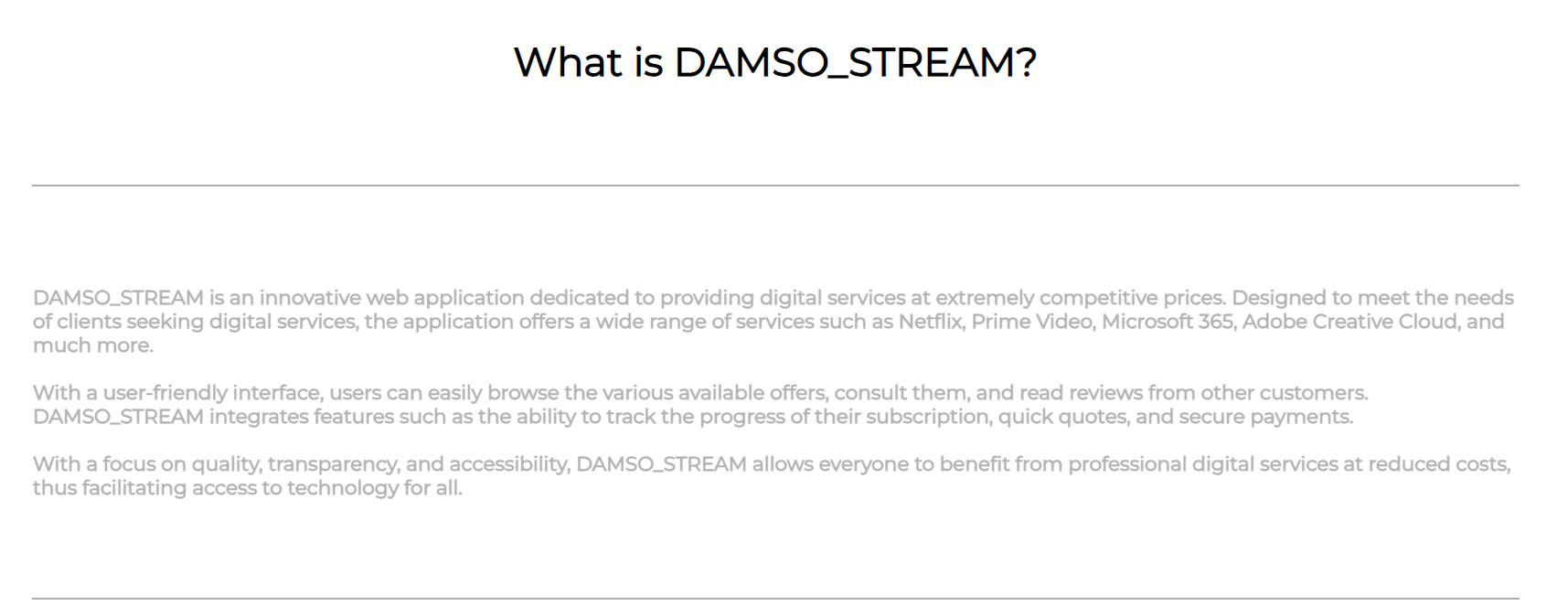
* 1. Page Contact US

C’est ou le client peux trouver toutes les informations de contact la localisation s’il existe et peux aussi envoyez un mail personnel a la page, si l’utilisateur est connecté sur leur compte personnel, les champs Name et Email son remplit automatiquement il reste que l’écriture de msg personnaliser.





* 1. Page About US

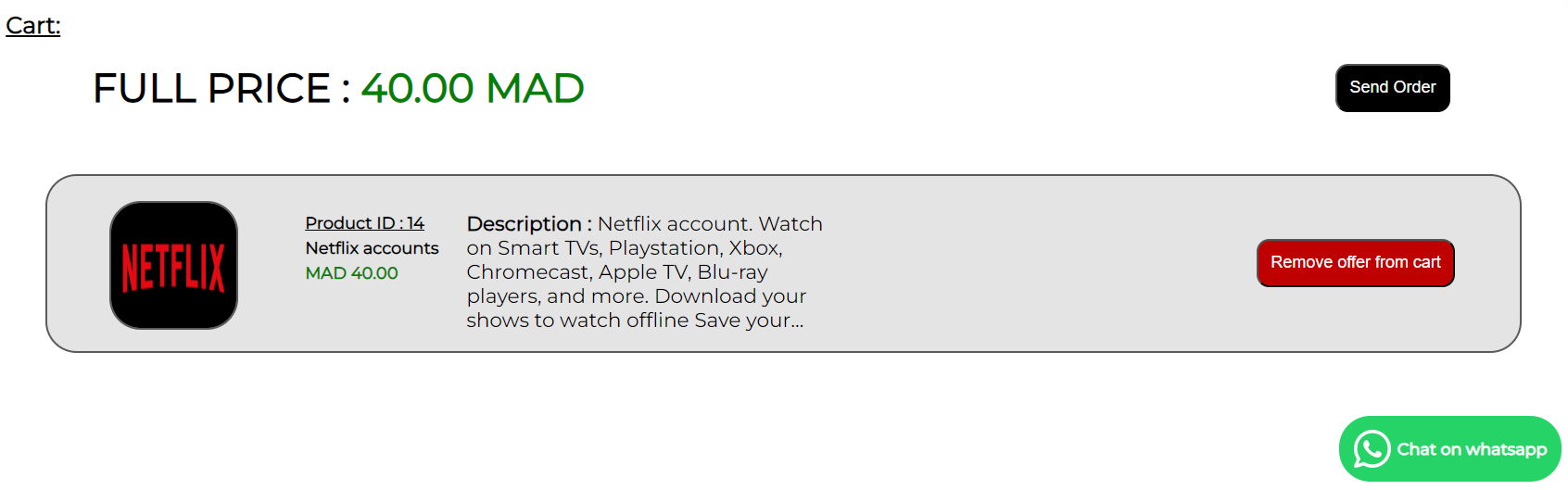
C’est la page ou l’utilisateur peux trouveur plus d’information concernant l’idée de la page et l’objective de cette application web.

**Figure 15 : presentation de l'application web**

.

* 1. Page Cart

Le client doit être connecté pour avoir accès à cette page, afin d'envoyer des commandes aux employés ou aux administrateurs pour qu'elles puissent être traitées par eux, cette page et vider après chaque envoie de commande (c’est-à-dire que la table cart et vider), et chaque utilisateur a son propre panier personnel au moment de création de compte.



**Figure 16 : page de panier**

* 1. Page d’utilisateur

C’est ou l’utilisateur peux voir ces informations personnelles, faire un changement des informations, Log out, changement de l’image de profile, suppression de compte, et aussi accéder à sa propre page personnelle ou chaque rôle a son propre page personnel, le client a une page ou il peut voir les informations sur l’abonnement, et autre information sur les commandes, et on a aussi les employé que peux

faire des opérations sur les commandes, et finalement les admins qui peux faires des opérations sur les utilisateurs, et les offres. 



* 1. Les Pages Personnels

Chaque utilisateur a son propre page personnel ou il peut voir des information et faires des opérations cette page se dépend de le rôle d’utilisateur (client,employee,admin), chacun de ces rôle a sa propre page de rôle, si l’utilisateur changer l’email il va recevoir un mail pour vérifier son nouveau mail.

Note : (Chaque utilisateur une image par défaut au moment de création de compte avec cette page l’utilisateur peut changer cette image par défaut)

* + 1. Page admin

C’est ou l’admin peux voir les informations de l’application web comme le nombre de clients, les commandes, et le revenu, la listes des clients, employé et admins, et aussi changer le rôle d’un USER (Note : un admin ne peut pas changer le rôle d’un autre admin aussi la même chose pour la suppression d’USR ), C’est aussi la place ou l’admin peux ajouter, modifier, supprimer des offres, Et finalement il peut aussi faire le traitement des commandes comme les employé





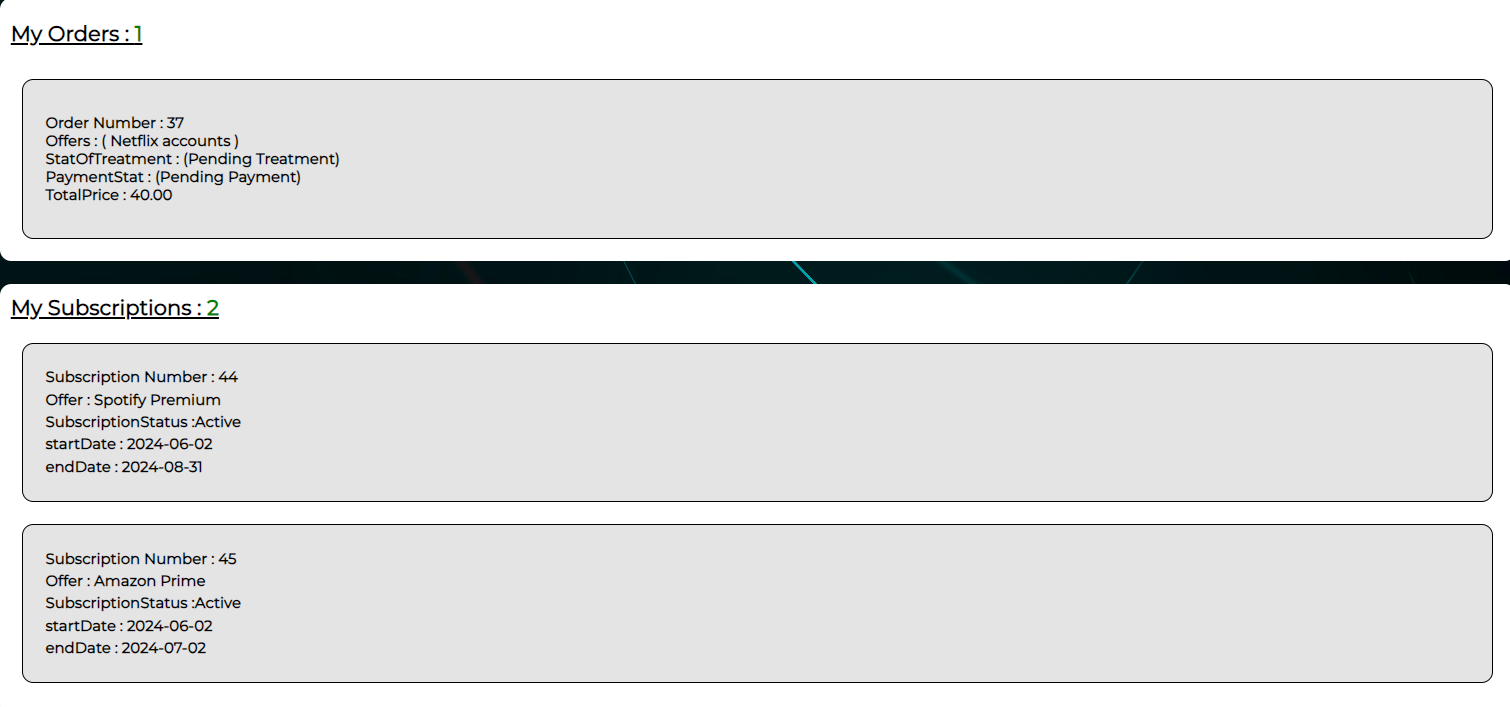
* + 1. Page Employé

C’est la page ou l’employé peux faire le traitement des commandes il a un seul privilège, c’est de traiter la commande il peut aussi contacter le client automatiquement en cliquent sur l’email ou le WhatsApp,

Pour l’employé la page cart et masquer aussi que le Button add cart.



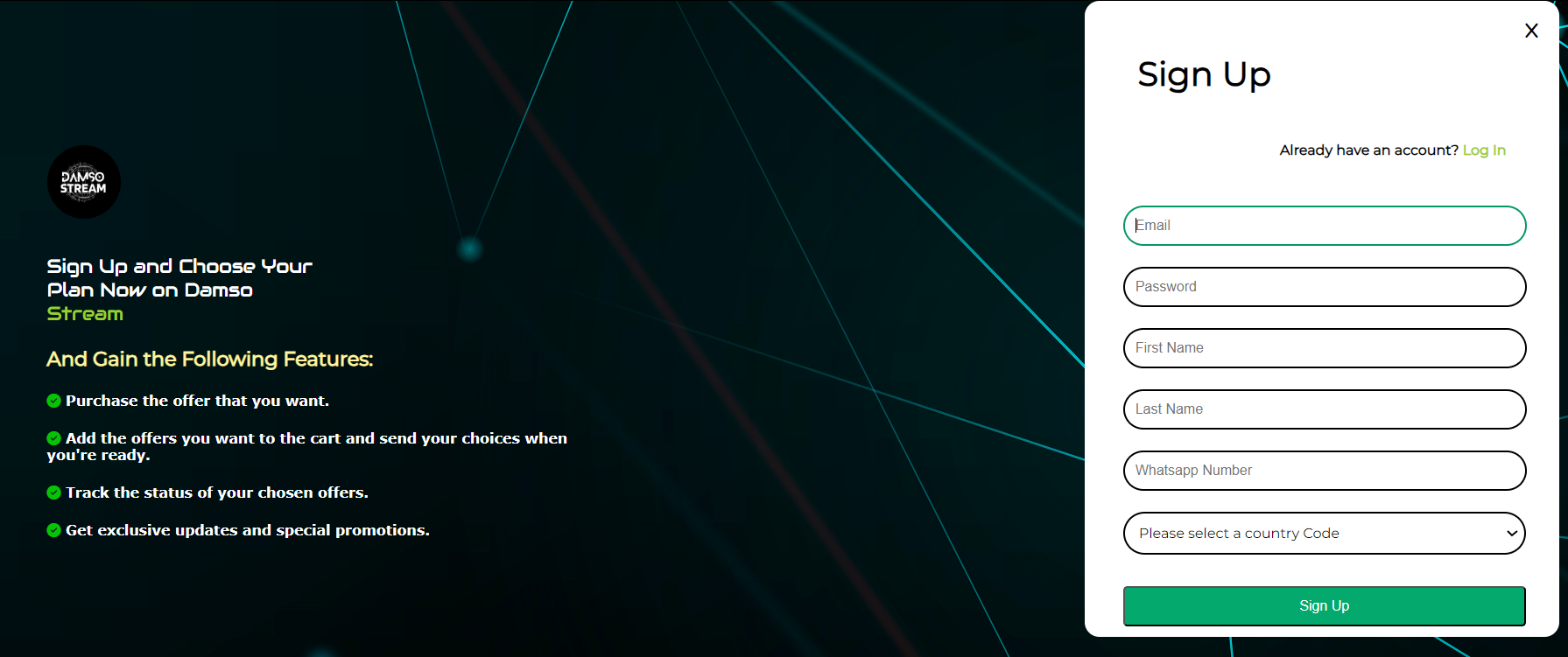
* + 1. Page Client

C’est ou le client peut voir les informations de la commande envoyez par eux et le statut de l’abonnement, il peut aussi renouveler l’abonnement si possible ou supprimer les abonnements expirer

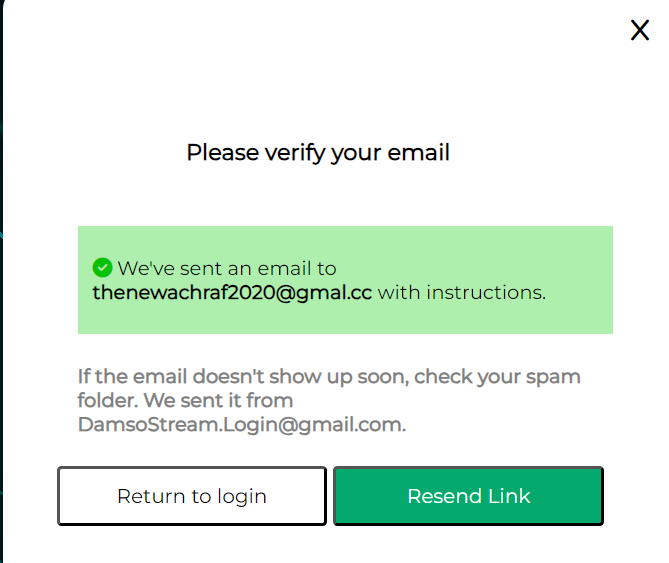
* 1. Sign Up

C’est la partie ou l’utilisateur peut créer un compte, la saisie de numéro de téléphone ce n’est pas obligatoire, mais quand l’utilisateur entre le numéro la saisie du code de pays devient obligatoire, le formulaire est bien vérifié toutes les possibilités des erreurs, le format d’email, cases vide, mot de passe qui contient des contraint à suivre, un email qui existe, ….

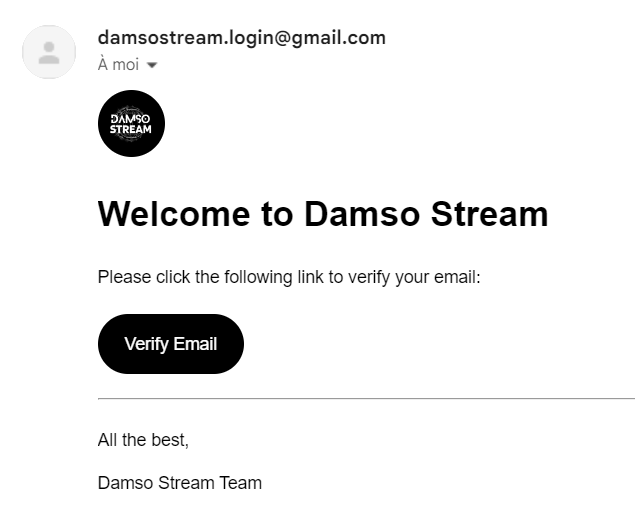
Apres ça on a une vérification d’email envoyez vers l’email saisie pour changer le statut de compte (non vérifier vers vérifier).



**Figure 25 : page sign up**



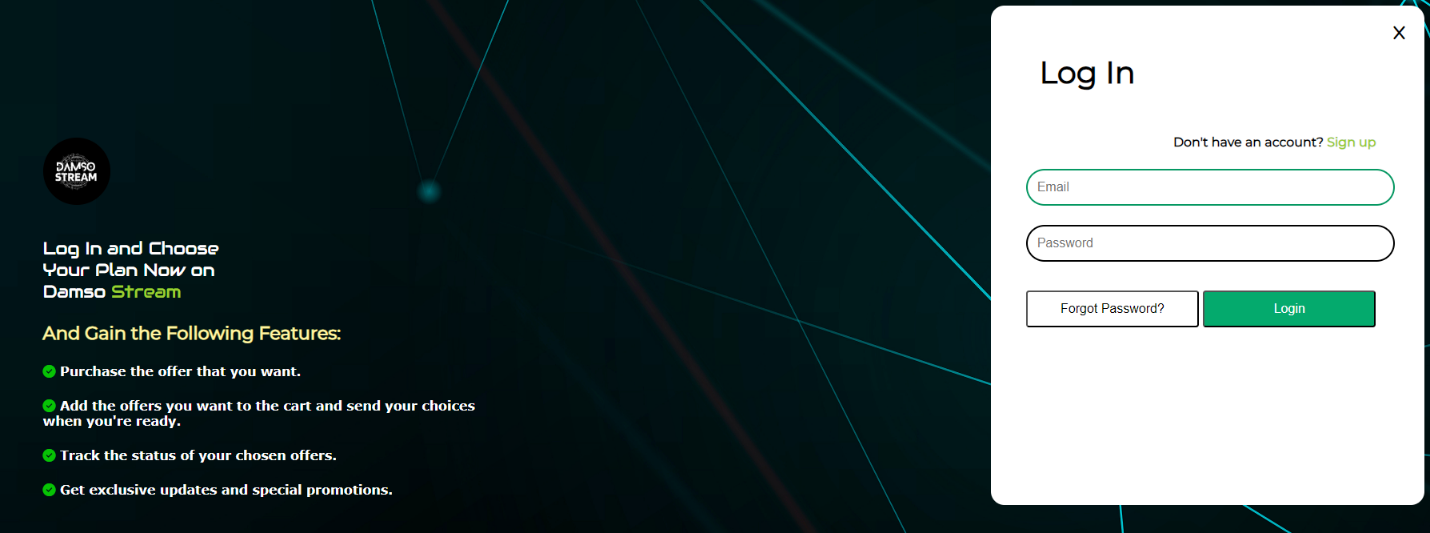
Dans cette partie le client peut juste vérifier sa boite de mail et cliquer sur le lien de vérification et le compte va être vérifier automatiquement



**Figure 27 : l'email de verification**

* 1. Login + ForgotPassword

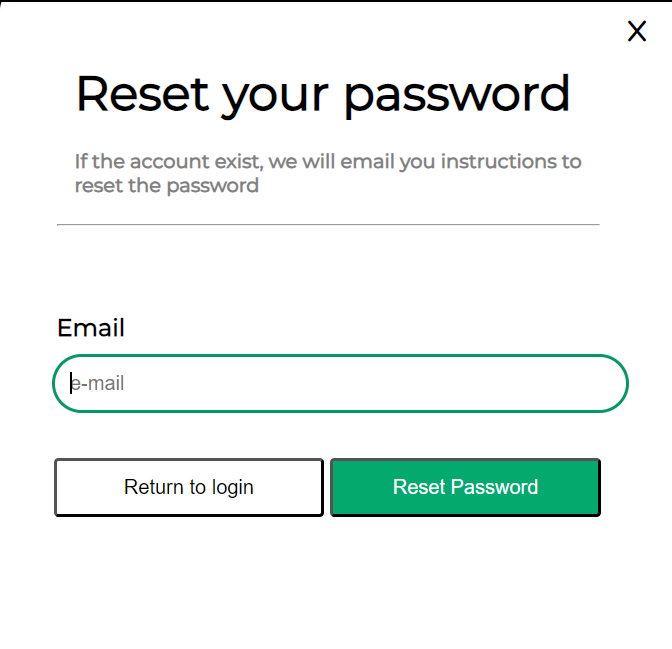
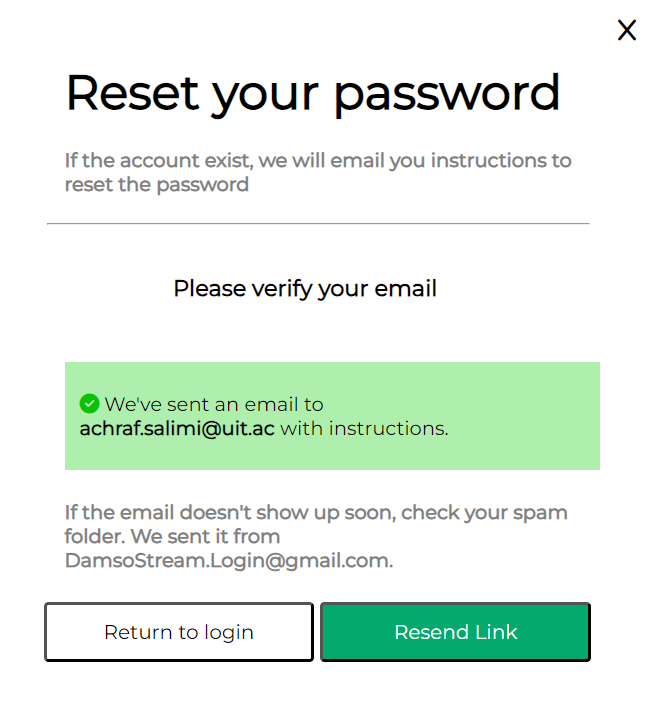
C’est la partie ou l’utilisateur peut finalement connecter sur son compte personnel, le formulaire et aussi bien vérifier les erreurs, il a aussi la partie (ForgotPassword)

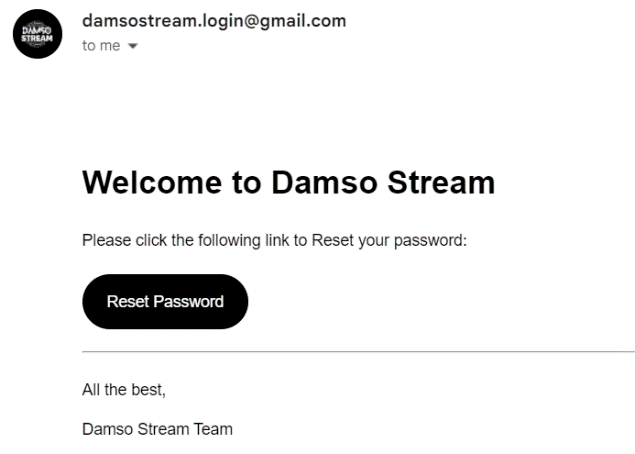
Si l’utilisateur oublier son mot de passe il peut le récupérer Just enter son email il va recevoir un email qui le donne un lien en cliquent sur ce lien il peut changer le mot de passe son compte.

Note : Pour rester connecter on a utilisé les sessions avec une durée d’expiration de 24h.

**app.permanent\_session\_lifetime = timedelta(hours=24)**

**timedelta** c’est une fonction de la bibliothèque **datetime.**





1. Scénarios de Fonctionnalités.
   1. Gestion des Abonnements

Pour la partie d’abonnement, l’abonnements s’ajouter automatiquement quand l’employé ou l’admin click sur « Change to Payed Status » , tous les offres dans la commande vont être insérer dans la table abonnement avec la dure de start date (durée courante) et la dure end Date (durée courante + la durée de l’offre ).

Et quand l’offre dépasse la date de fin, le statut de l’abonnement change de Active vers Inactive, et l’utilisateur a 2 options renouveler l’offre inactive ou le supprimer l’offre dans la partie « My Subscriptions ».



* 1. Gestion des Offres

Chaque offre caractériser par le nom de l’offre, prix, dure(en jour),description, si l’image n’est pas saisie une image par défaut va être affecter à cette offre, on peut changer l’image ailleurs avec une autre (seulement l’admin peux faire ça )

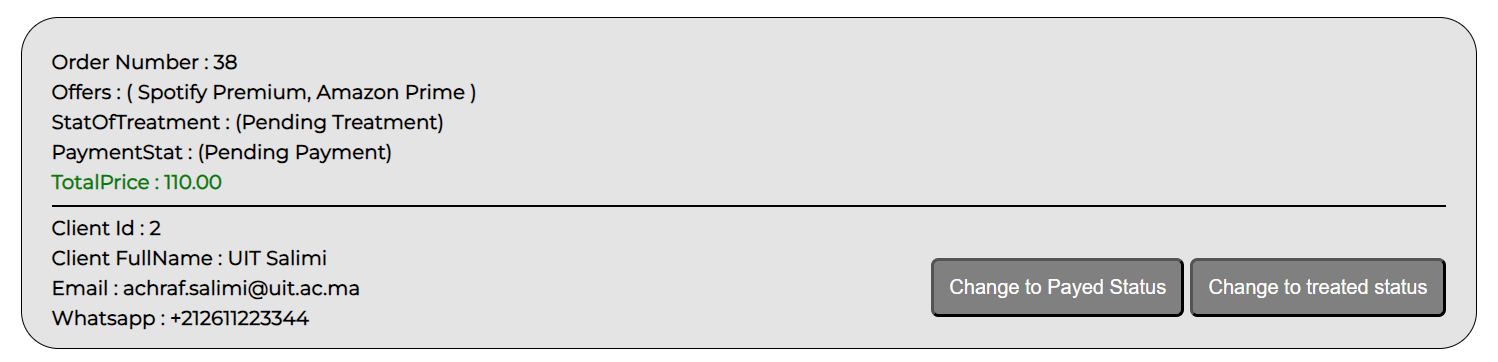
L’admin seulement qui peux ajouter, modifier, supprimer une offre

* 1. Gestion des Commandes

Les commandes arrivent de la page panier personnel d’utilisateurs, les commandes sont affichées a les admins et les employé pour le traiter, ils contact le client et changer le statuts de la commande si il est changé par payed, les offres de commande va être ajouter dans la table abonnement.

L’admin et l’employé peux juste cliquer sur le WhatsApp s’il existe ou l’email, ils sont allés être rediriger vers la page approprier pour les contacter.

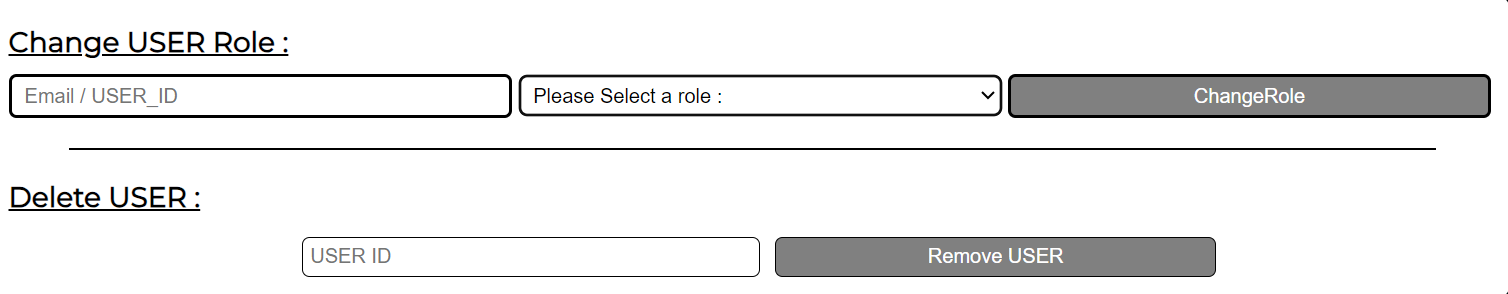
Ils ont deux opérations, « Change to Payed Status » une commande Payed sera automatiquement treated ( traiter)  « Change to treated status  » cette option c’est pour annuler la commande si le client n’a pas payé le prix demander après le contact avec eux.



**Figure 33 : commande**

* 1. Gestion des Utilisateurs

Pour les opérations sur les utilisateurs sont un changement de role vers (client, employe , admin), ou supprimer le compte d’un utilisateur, les formulaires de ces options sont bien vérifiés au niveau de base de donner et au niveau de JavaScript , telle que l’admin ne peux pas supprimer ou changer le role d’un autre admin pour des raison de sécurité.



**Figure 34 : changement de role et suppresion d'utilisateur**

* 1. Gestion des commentaires

Pour la gestion des commentaires chaque offre a sa propre section de commentaire, c’est-à-dire que chaque partie a son propre commentaire, un utilisateur peux commenter s’il est connecté et peux supprimer son propre commentaire.

L’admin peux aussi supprimer les commentaires mais de tous les utilisateurs.



**Figure 35 : partie commentaires**

1. Détails des Fonctions Python
   1. Gestion des Images et Fichiers

Pour les extensions ont permis seulement

ALLOWED\_EXTENSIONS = **{**'txt'**,** 'pdf'**,** 'png'**,** 'jpg'**,** 'jpeg'**,** 'gif'**}**

UPLOAD\_FOLDER = 'static\\Images\\product\_pics'

C’est ou on stock les images de produit

UPLOAD\_FOLDER = 'static\\Images\\Users\_pfp'

C’est ou on stock les images des utilisateurs

En ajoutant enctype=multipart/form-dat pour les formulaires qui contient l’input de type file , en récupère le fichier en utilisant

**file = request.files["file"]**

En prend l’extension de fichier et en change le nom de fichier, en prend d’abord le full Name de fichier avec **secure\_filename(file.filename)**

Pour éviter les noms non sécuriser, et pour que l’utilisateur ne peux pas modifier la partie des fichiers en utilisant le nom de fichier.

Apres en configure le path de fichier **os.path.join(*path , nomFichier*)**

Si cette image (fichier) existe déjà alors on supprime l’image précèdent s’il est différent de l’image par défaut avec **os.remove(le path final générer d’après os.path.join )**

Et en utilise **file.save(le path final générer d’après os.path.join )**

pour stocker ce fichier dans ce path.

* 1. Gestion de numéro de téléphone

En Vérifier d’abord si le numéro commence avec ‘**+’** en utilisant **number.startswith('+')**

Si c’est le cas en Transfer le numéro vers un type utiliser par la bibliothèque **phonenumbers** en utilisant **phonenumbers.parse(number)**

et en utilisant la méthode suivant en prendre le numéro de téléphone sans le code de pays **phonenumbers.format\_number(parsed\_number, phonenumbers.PhoneNumberFormat.NATIONAL)**

**''.join(filter(str.isdigit, national\_number)).lstrip('0')**

**Filter(…)** crée un Loop qui retourne seulement les caractères qui ont de type nombre, et en utilisant ‘ **’.join(filter(…))** en concaténer ces caractères dans une seule chaine de caractère, et aussi en utilisant **lstrip(‘0’)** en enlevé les 0 qui on au départ de numéro c’est-à-dire a gauche de cette chaine par exemple 0611223344 il enlevé le 0 de début .

Si ce n’est pas le cas ( commence pas par + ) donc ce numéro ne contient pas un code de pays donc ce n’est pas l’appel les supprimer, ont enlevé seulement les 0 avec **number.lstrip('0')**

Enfin on combine le numéro après le traitement précèdent et le country\_code donner par l’utilisateur qui nous donne finalement

**full\_number = f"+{country\_code}{national\_number}"**

et en retourne finalement le numéro après tous ce traitement avec

**return phonenumbers.format\_number(parsed\_number, phonenumbers.PhoneNumberFormat.E164)**

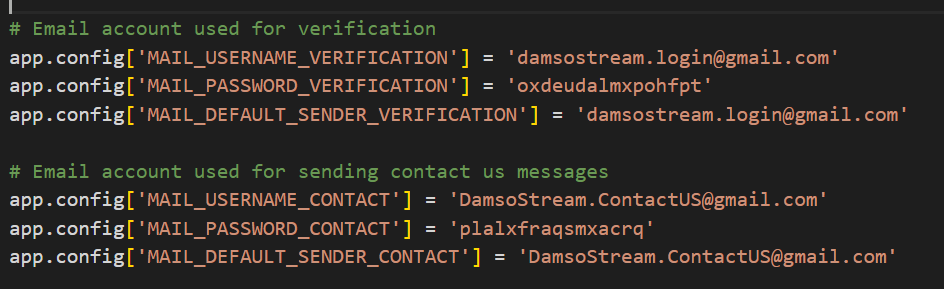
**En format E164 qui est le format international d’un numéro de téléphone a des caractéristique (**

* maximum de 15 chiffres.
* Le code pays (de 1 à 3 chiffres).
* Un signe plus (+) en tête pour indiquer la numérotation internationale.
* Le numéro national significatif (le reste des chiffres, qui comprend le code régional et le numéro local).
  1. Envoie des emails

Pour l’envoie des emails on a utilisé la bibliothèque FLASK\_MAIL

Avec Gmail SMTP serveur car il nous donne jusqu’à 2000 email gratuit par jours

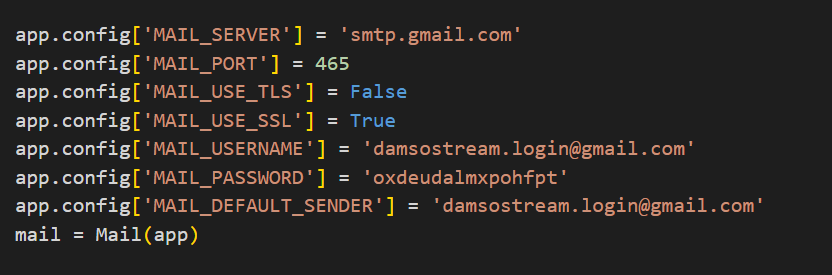
Avec la configuration suivante :



**Figure 36 : les emails de verifications**

Deux emails un pour envoyer un mail, le premier c’est utiliser pour envoyer un mail pour la vérification d’email, et pour envoyer le lien de récupération de mot de passe

Le deuxième est pour envoyer des emails, à partir de cet email le client peux envoyez à « [DamsoStream@gmail.com»](mailto:DamsoStream@gmail.com) un mail.



**Figure 37 : configuration de gmail smtp**

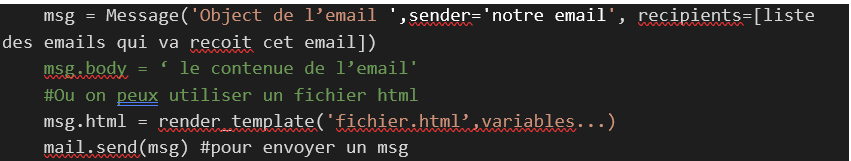
Ça c’est la configuration par défaut quand en envoie un mail, et quand en veux envoyez un mail en utilisant la Page Contact Us c’est-à-dire le deuxième email en change la configuration par défaut avec la config de deuxième email donnée précédemment.

Pour SSL : 465 ; Pour TLS : 587

Ces deux sont des protocoles de communication cryptographique,

TLS c’est plus sécurisé que SSL mais il’ avait un délai très lent que SSL ;

Est pour envoyer un email en utilise cette méthode :



**Figure 38 : template d'evoie d'un email**

1. Assurance qualité et optimisation

Pour l’assurance de qualité en a utiliser la bibliothèque AXIOS pour ne pas recharger les donner de la page avec des nouvelles donner pour afficher une erreur après le traitement dans la base de donner donc on a utilisé cette bibliothèque pour envoyer des requête de vérification vers python, pour traiter ces donner et les vérifier au niveau de la bases de donner sans recharger la page ce qui est mieux de recharger la page juste pour vérifier, par exemple vérifier si l’email de signup déjà existe dans la base de donner ou non.

1. Conclusion

Tous les pages sont dynamique c’est-à-dire que il dépend de la partie base de donner si on change quelque chose dans la base de donner la page aussi change, l’affichage, la recherche, ajout, modification, suppression sont fait à partir des requête SQL qui nécessite parfois des jointure et parfois non, les formulaire des information sont bien vérifier, donc la donner est vérifier avant de l’insérer, il est vérifié à partir de partie JavaScript en utilisant des Regular expressions, et aussi une vérification de l’existence dans la partie de base donner sans de recharger la page avec l’utilisation de la bibliothèque AXIOS et la vérifications se fait au niveau de python.

*CHAPITRE 5 : Conclusion*

1. Conclusion General

Dans le cadre de notre projet de développement de l’**application web qui offre des services digitaux**, nous avant réaliser une solution idéale et simplifier pour mettre l’achat des offres online plus simple et plus confortables pour les utilisateurs, cette application web est dédiée à fournir des services digitaux à des prix extrêmement compétitifs. Conçue pour satisfaire les besoins des clients désirant de s’acquérir des services digitaux.

Au cours de la réalisation de se projet on a améliorer nos compétences dans la partie de **programmation**, **conception** aussi, on a appris le language de programmation **python** et on a connus le fonctionnement de quelque bibliothèque qui nous ont beaucoup aides dans la réalisation de notre projet.

On a connu **comment réaliser une application web Dynamique** qui dépendre de la base de donner, appris comment **connecter la base de donner avec un language de programmation comme python**, maitriser les bases de **HTML,CSS** surtouts le **positionnement** avec **ccs**, aussi métriser la partie de **JavaScript .**

On a rencontré beaucoup des difficultés au départ mais au moment de réalisation et avec le temps on a trouvé des solutions et appris des ces erreurs beaucoup de choses , on a pris beaucoup de temps dans la partie de conception de base de donner, on a aussi le problème de temps et la gestion de temps entre les modules et le projet , et si on a le temps on aime beaucoup d'ajouter l'option de **responsivité** a notre site en utilisant les **Query Media** pour être responsive pour tous les écrans (téléphone ,ordinateur, tablette, et beaucoup de autres taille d’encrant ) mais malheureusement on a pas le temps.

1. Webographie
2. Formulaire qui contient les codes de payes : « https://gist.github.com/andyj/7108917 »
3. Pour l’upload des fichiers : « https://flask.palletsprojects.com/en/2.3.x/patterns/fileuploads/ »
4. Inspiration de design : « https://www.w3schools.com/ »
5. Définition d’UML : « <https://www.lucidchart.com/pages/fr/langage-uml#:~:text=Le%20langage%20UML%20(Unified%20Modeling,et%20riche%20s%C3%A9mantiquement%20et%20syntaxiquement>. »
6. Définition de Python : « https://aws.amazon.com/fr/what-is/python/#:~:text=Python%20est%20un%20langage%20de,le%20machine%20learning%20(ML). »
7. Définition de SQL :  « <https://datascientest.com/sql-tout-savoir>»
8. Définition de JavaScript : « <https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/JavaScript_basics> »
9. Définition de HTML : « <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/> »
10. Définition de CSS : «  <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203277-css-cascading-style-sheets-definition-traduction/> »
11. Définition de Flask : « <https://fr.wikipedia.org/wiki/Flask_(framework)> »
12. Définition de Flask\_mail :

« [https://pypi.org/project/Flask-Mail/](https://pypi.org/project/Flask-Mail/%20)  »

1. Définition de Moteur de template Jinja : «  https://fr.wikipedia.org/wiki/Jinja\_(moteur\_de\_template) »
2. Définition de PhoneNumber library :

« https://www.geeksforgeeks.org/phonenumbers-module-in-python/ »

1. Définition de MySQL connecter :

« https://gayerie.dev/docs/python/python3/mysql.html#:~:text=Le%20module%20mysql.,mot%20de%20passe%20de%20connexion. »

1. Définition de DateTime :

«  <https://docs.python.org/fr/3.6/library/datetime.html> »

1. Définition de AXIOS :

« https://pixelalliance.io/content/comment-marche-axios-la-librairie-javascript-et-comment-appeler-un-strapi-avec- »

1. Définition de MySQL :

« https://fr.wikipedia.org/wiki/MySQL#:~:text=MySQL%20(prononc%C3%A9%20%5Bmaj.,double%20licence%20GPL%20et%20propri%C3%A9taire.  »

1. Définition de Visual Studio Code :

« https://fr.wikipedia.org/wiki/Visual\_Studio\_Code#:~:text=Visual%20Studio%20Code%20est%20un%20%C3%A9diteur%20de%20code%20source%20qui,Node.js%20et%20C%2B%2B. »

1. Définition de Astah (UML) :

« https://www.guru99.com/fr/best-uml-tools.html#:~:text=Astah%20est%20un%20outil%20de,%2C%20Mac%2C%20Linux%20et%20iOS. »