

**Gestion d'un Hôtel**

**Projet Scolaire**

Réalisé par :

Ghayt El idrissi Dafali :

Reda Bouimakliouine :

Amine Ibnou-Cheikh :

Encadré par :

Mr. Ali IDMANSOUR :

**Table de matière :**

Table des matières

[1. Introduction : 3](#_Toc187165849)

[2. Présentation Du projet : 4](#_Toc187165850)

[2.1 Fonctionnalités principales 5](#_Toc187165851)

[2.2 Différences entre web et desktop 6](#_Toc187165852)

[3. Technologie Utilisées 8](#_Toc187165853)

[4.Implementation 10](#_Toc187165854)

[4.1 Structure du projet 10](#_Toc187165855)

[5.Base de donnee : 12](#_Toc187165856)

[5.1 Table principales 12](#_Toc187165857)

[5.2 Diagramme de la base de donnee : 16](#_Toc187165858)

[6.Resultat finale : 18](#_Toc187165859)

[6.1 Interfaces utilisateur 18](#_Toc187165860)

[6.1 Interface utiisateur : 24](#_Toc187165861)

[Conclusion 38](#_Toc187165862)

# Introduction :

L’objectif principal de ce projet est de concevoir et développer une solution logicielle complète permettant de gérer les opérations d’un hôtel. Ce système comprend une application web pour les utilisateurs externes (comme les clients) et une application desktop dédiée à l’administration interne.

Pour atteindre cet objectif, plusieurs technologies modernes ont été utilisées :

* ASP.NET Core MVC : pour la création de l’application web.
* WPF (Windows Presentation Foundation) : pour l’application desktop.
* SQL Server : pour gérer les données de manière structurée.
* Entity Framework : pour faciliter les opérations sur la base de données

Ce rapport est structuré comme suit :

* Le premier chapitre présente le projet et ses fonctionnalités principales.
* Le second chapitre décrit les technologies et outils utilisés.
* Les chapitres suivants détaillent l’implémentation, la structure de la base de données et les résultats obtenus.
* Enfin, le rapport se termine par une conclusion et une discussion sur les améliorations possibles.

# Présentation Du projet :

Ce projet a été réalisé dans le cadre de nos études pour mettre en pratique les concepts appris en programmation et en gestion de bases de données. Il nous a permis de développer nos compétences dans des technologies modernes comme ASP.NET Core MVC, WPF, SQL Server, et Entity Framework.

Le projet a pour but de simuler un système de gestion d’hôtel simple, destiné à apprendre le développement d'applications web et desktop, tout en mettant en œuvre une base de données relationnelle.

L’objectif principal est de développer une application complète combinant plusieurs technologies et de maîtriser les concepts suivants :

* Développement web avec ASP.NET Core MVC.
* Création d'applications desktop avec WPF.
* Utilisation d'une base de données relationnelle avec SQL Server.
* Manipulation des données avec Entity Framework.

## 2.1 Fonctionnalités principales

Le projet que nous avons réalisé inclue plusieurs Fonctionnalités tels que :

* 1. **Gestion des clients:** l’administrateur peut faire différentes taches , il peut ajouter des client ,Supprimer , modifier les information et recherche des clients tous via l’interface gestion des client.
  2. **Gestion des chambres:** pour l’interface de gestion des chambre il peut voir les chambre occupées dans une date bloqué l’accès a une chambre, ajouter supprimer ou bien modifier la catégorie.
  3. **Gestion Des Réservations**: l’interface de gestion des réservations permet à l’employé ou à l’admin de réserver une chambre ou bien supprimer, annuler une réservation et afficher les réservation , trier , et également afficher les réservation payer ou non payer.
  4. **Gestion Des payements :** cette interface gère les réservations non payer.

## 2.2 Différences entre web et desktop

La version web est destinée aux **clients** de l’hôtel. Son objectif principal est de leur offrir une plateforme simple et accessible pour consulter les chambres disponibles et effectuer des réservations. Voici les fonctionnalités spécifiques de cette version :

* **Version web**

La version web est destinée aux clients de l’hôtel. Son objectif principal est de leur offrir une plateforme simple et accessible pour consulter les chambres disponibles et effectuer des réservations. Voici les fonctionnalités spécifiques de cette version.

1. **Consultation des chambres disponibles** :

Les clients peuvent accéder à une liste des chambres disponibles, avec des informations détaillées telles que :

* Le type de chambre (simple, double).
* Le prix par nuit.

Les informations sont affichées de manière claire et conviviale pour faciliter la navigation

**2. Réservation en ligne** :

* Les clients peuvent choisir une chambre disponible, sélectionner leurs dates de séjour et renseigner leurs informations personnelles (nom, email, téléphone).
* Un Email est envoyer a leurs courrier indiquant toutes les informations concernant ses réservation avec un pdf généré.

**Exemple de scénario utilisateur (web)** :

* Un client ouvre le site web de l’hôtel.
* Il consulte les chambres disponibles pour ses dates.
* Il sélectionne une chambre, renseigne ses informations et clique sur "Réserver".
* Le système confirme la réservation et affiche un numéro de référence.
* **Version desktop**

La version desktop est destinée à l’administrateur de l’hôtel. Elle est conçue pour gérer toutes les opérations internes liées à la gestion des clients, des chambres et des réservations. Voici les fonctionnalités spécifiques de cette version :

1. Gestion des clients :
2. Gestion des réservations :
3. Gestion des chambres :
4. Gestion des Payements :

**Résumé**

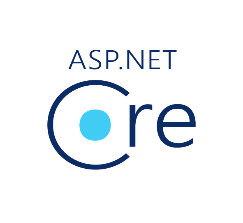
* **Version web** : Simple et conçue pour les clients. Elle permet de consulter les chambres et de réserver en ligne depuis n’importe quel appareil.
* **Version desktop** : Plus complète, elle est destinée à l’administrateur pour gérer l’ensemble des données de l’hôtel (clients, chambres, et réservations). Elle nécessite une installation sur un poste de travail.

# Technologie Utilisées

Dans le cadre de ce projet, plusieurs technologies modernes ont été utilisées pour assurer la réussite du développement. Ces outils ont été choisis en fonction des besoins du projet (développement web, application desktop, gestion de base de données, etc.). Voici une description des technologies principales :

**3.1 ASP.NET Core MVC :**

ASP.NET Core MVC est un framework puissant pour créer des applications web modernes en suivant le modèle MVC (Modèle-Vue-Contrôleur). Ce choix a été fait pour sa flexibilité, sa compatibilité avec les bases de données, et sa capacité à gérer des interfaces web dynamiques. Dans ce projet, il a été utilisé pour développer la partie web de l'application, permettant aux clients de consulter les chambres disponibles et de réserver en ligne.



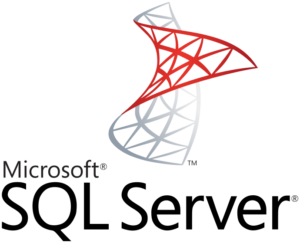
**3.2 WPF (Windows Presentation Foundation) :**

WPF est une technologie utilisée pour créer des applications desktop avec des interfaces graphiques riches et modernes. Elle a été utilisée pour développer l’application desktop destinée à l’administrateur. Ce choix a été motivé par la facilité d’intégration avec Entity Framework et la possibilité de créer des interfaces personnalisées.



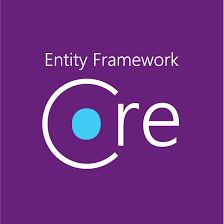
**3.3 SQL Server :**

SQL Server a été utilisé pour gérer les données structurées du projet (clients, chambres, réservations). Son intégration avec Entity Framework en fait un choix idéal pour ce type de projet, garantissant des performances élevées et une gestion efficace des données.



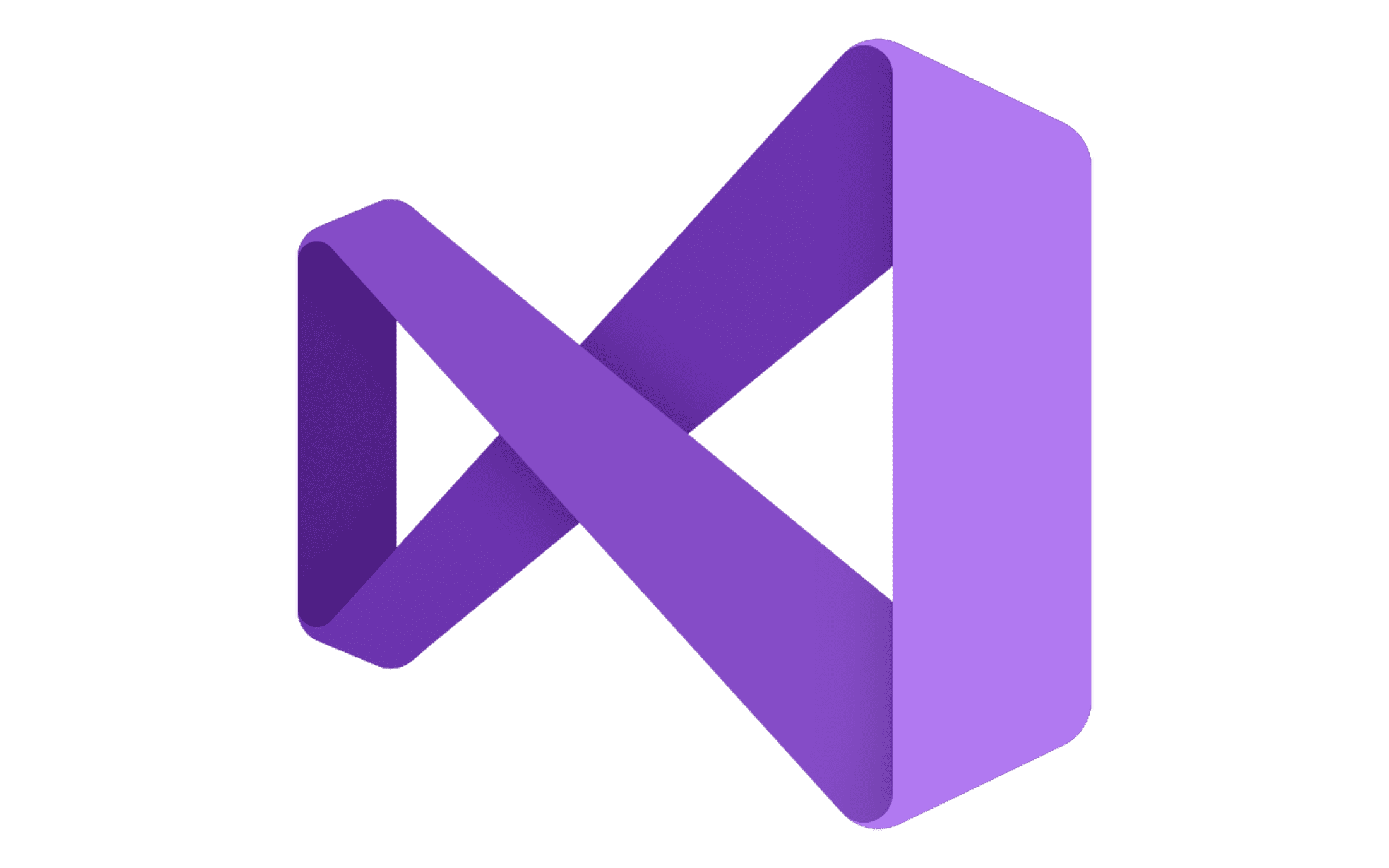
**3.4 Entity Framework :**

Entity Framework est un ORM (Object Relational Mapping) utilisé pour interagir avec la base de données sans écrire de requêtes SQL complexes. Grâce à Entity Framework, toutes les opérations CRUD ont été réalisées de manière simple et efficace.



**3.5 Outils de développement :**

* Visual Studio : Pour le développement, le débogage et la gestion des projets ASP.NET Core et WPF.



* SQL Server Management Studio (SSMS) : Pour gérer la base de données et tester les requêtes SQL.



# 4.Implementation

Ce chapitre décrit en détail les étapes de développement du projet, en mettant en avant l’architecture utilisée, les fonctionnalités principales, et les interactions avec la base de données. Les extraits de code présentés permettent de mieux comprendre les mécanismes sous-jacents de l’application.

## 4.1 Structure du projet

Le projet est divisé en deux parties principales : une application web et une application desktop, qui partagent une base de données commune. Voici un aperçu de leur structure et fonctionnement.

* + 1. Application web (ASP.NET Core MVC)

L’application web utilise l’architecture Modèle-Vue-Contrôleur (MVC), qui facilite la séparation des responsabilités et la maintenance du code.

**Modèle** :

* Contient les classes représentant les entités Client, Chambre, Reservation,Payement , Type chambres ,Notification, Login .
* Ces classes sont directement mappées à la base de données grâce à Entity Framework.

**Vue** :

* Génère les pages HTML affichées à l’utilisateur à l’aide de Razor.

**Contrôleur** :

* Gère les requêtes HTTP, traite les données, et renvoie les réponses.
* Exemple

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

4.1.2 Application desktop (WPF)

L’application desktop est conçue pour permettre à l’administrateur de gérer les clients, les chambres et les réservations. Elle utilise une structure combinant le modèle **MVVM** (Modèle-Vue-Modèle de Vue) et des **services** pour séparer davantage les responsabilités.

1. **Modèle (Model)** :
   * Contient les classes représentant les entités principales : Client, Chambre, Reservation et Payement .
   * Ces classes interagissent avec la base de données via Entity Framework.
2. **Vue (View)** :
   * Définie avec **XAML**, elle offre une interface graphique intuitive.
   * La vue est liée au Modèle de Vue grâce à la liaison des données (**Data Binding**).
3. **Modèle de Vue (ViewModel)** :
   * Contient la logique métier liée à la vue (ex. : validation des champs, gestion des commandes).
   * Communique avec des **services** pour effectuer des actions sur les données.
4. **Services** :
   * Les services encapsulent la logique métier et les interactions avec la base de données.
   * Ils permettent de réutiliser le code et d’éviter de mélanger la logique métier avec la logique de présentation.
5. **Contrôleur (Controller)** :
   * Les contrôleurs sont utilisés pour gérer des actions complexes et coordonner les interactions entre les services et le Modèle de Vue.

# 5.Base de donnee :

## 5.1 Table principales

Dans le cadre du développement du site web de notre hôtel, nous avons utilisé **SQL Server Management Studio (SSMS)** pour gérer la base de données. Cet outil nous a permis de créer, configurer et maintenir efficacement les données nécessaires au bon fonctionnement du site.

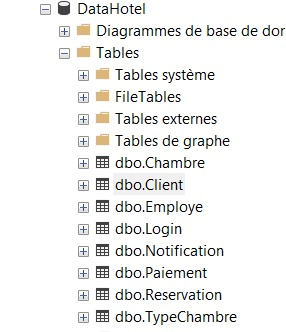
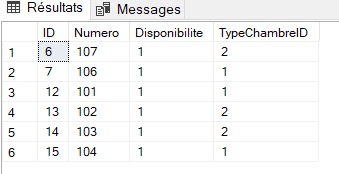


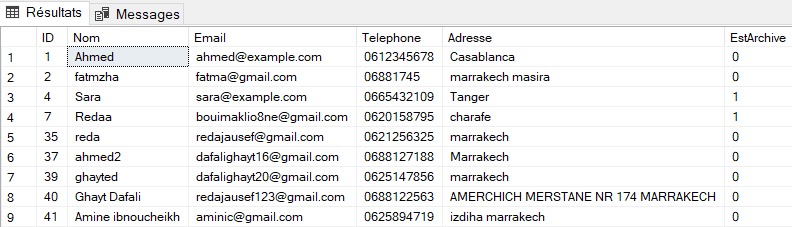
Figure1

EXPLICATION : La base de données est composée de plusieurs tables, identifiables par le préfixe **dbo**, qui contiennent les informations nécessaires pour gérer les différentes fonctionnalités du site. Voici une description des tables principales :

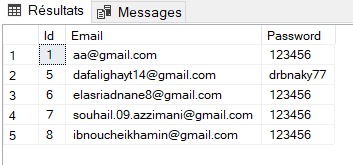
1. **dbo.Chambre**  
   Cette table stocke les informations sur les chambres de l’hôtel, comme :
   * Le numéro de la chambre.
   * La dispobinilite.
   * Le type de chambre



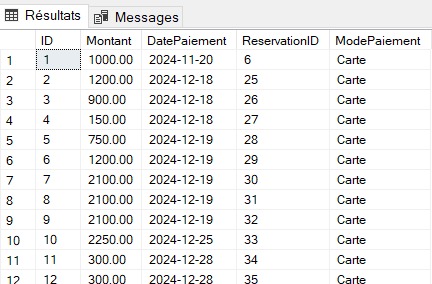
1. **dbo.Client**  
   Contient les données des clients, y compris :
   * Le nom et prénom.
   * Les coordonnées (adresse, numéro de téléphone, e-mail,archive).



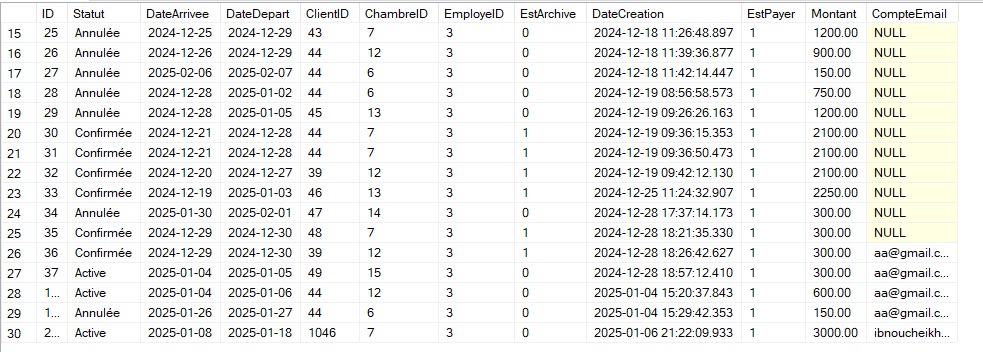
1. **dbo.Login**  
   Gère l’authentification des utilisateurs, incluant :
   * L’email d’utilisateur.
   * Les mots de passe .



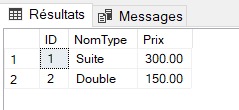
1. **dbo.Paiement**  
   Contient les détails des transactions financières :
   * Les méthodes de paiement utilisées (carte bancaire, espèce, etc.).
   * Les numéros de transaction et montants.



1. **dbo.Reservation**  
   Cette table est au cœur du système de réservation :
   * Elle enregistre les dates d’arrivée et de départ.
   * Elle associe chaque réservation à une chambre spécifique et à un client.
   * Date creation de la reservation
   * Clinet,chambre,employe IDs
   * Compte email associe



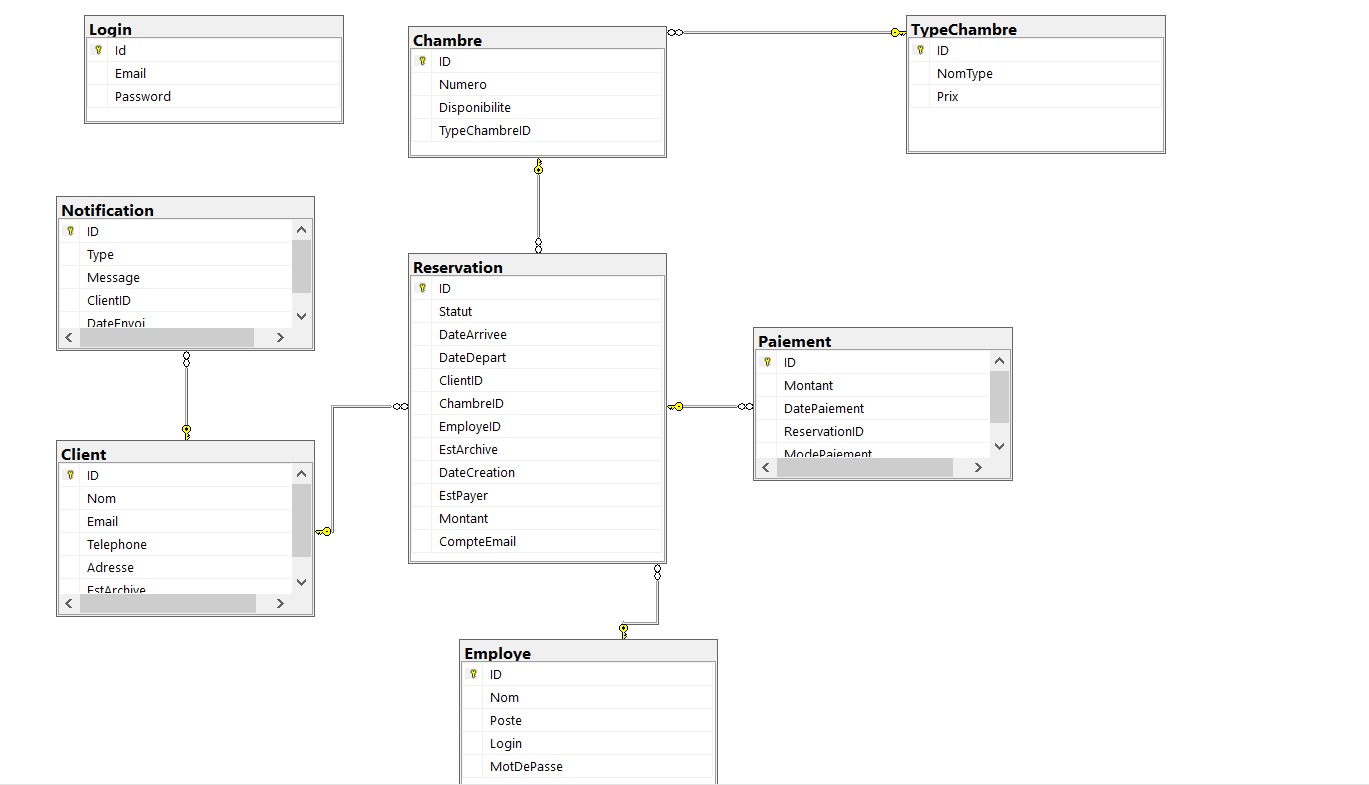
1. **dbo.TypeChambre**  
   Définit les catégories de chambres disponibles :
   * Simple, double, suite, etc.
   * Les tarifs associés à chaque type de chambre.



Ces tables sont interconnectées grâce à des clés primaires et étrangères, permettant de structurer et d’organiser les données efficacement. Par exemple, la table dbo.Reservation est liée à dbo.Client pour identifier le client ayant fait la réservation, et à dbo.Chambre pour attribuer une chambre spécifique.

Dans les sections suivantes, des captures d’écran illustreront ces tables et expliqueront leur contenu en détail, ainsi que leur utilisation dans le fonctionnement global du site.

## 5.2 Diagramme de la base de donnee :



L’image ci-dessus illustre le modèle relationnel de la base de données utilisée pour gérer les opérations de l’hôtel. Ce modèle repose sur une structure bien définie qui connecte les différentes entités grâce à des relations logiques. Voici une explication des tables principales et de leurs relations :

1. Table Client

* Contient les informations personnelles des clients, telles que :
  + Nom, e-mail, téléphone et adresse.
  + Statut d’archivage pour indiquer si le client est actif ou non.
* Relation : La clé primaire ID est utilisée comme clé étrangère dans les tables Notification et Reservation pour associer les clients aux réservations et notifications.

2. Table Chambre

* Stocke les détails des chambres, incluant :
  + Numéro de chambre, disponibilité, et type de chambre (référence à TypeChambre).
* Relation : La clé primaire ID est utilisée comme clé étrangère dans Reservation pour indiquer la chambre réservée.

3. Table TypeChambre

* Définit les types de chambres et leurs tarifs :
  + Nom du type (e.g., simple, double, suite).
  + Prix correspondant.
* Relation : Connectée à la table Chambre via la clé étrangère TypeChambreID.

4. Table Reservation

* Au cœur du système, cette table enregistre :
  + Les détails de réservation (dates d’arrivée et de départ, statut, montant).
  + Les relations avec les clients, chambres, et employés via des clés étrangères.
* Relation : Associée à Client, Chambre, et Employe.

5. Table Paiement

* Gère les paiements des réservations :
  + Montant, date de paiement, et mode de paiement.
* Relation : Connectée à Reservation via la clé étrangère ReservationID.

6. Table Employe

* Contient les informations du personnel :
  + Nom, poste, identifiants de connexion (login et mot de passe).
* Relation : Associée à Reservation pour identifier l’employé responsable.

7. Table Login

* Permet de gérer l’accès utilisateur :
  + Stocke les identifiants (e-mail et mot de passe).
* Utilisée pour authentifier les utilisateurs du système.

8. Table Notification

* Permet l’envoi de messages aux clients :
  + Type et contenu du message, date d’envoi.
* Relation : Connectée à Client via la clé étrangère ClientID

Diagramme de cas d’utilisation ADMIN ET CLIENT :

**A screenshot of a diagram

Description automatically generated**

Le diagramme présente l'ensemble des actions que l'administrateur peut effectuer dans un système. Ces cas d'utilisation sont reliés directement au rôle d'Admin, qui agit comme principal acteur. Les différentes fonctions disponibles sont les suivantes :

1. **Gérer les employés**  
   L'administrateur a la possibilité de gérer les informations relatives aux employés, comme l'ajout, la modification ou la suppression de leurs profils.
2. **Générer des reçus au format PDF (pour Admin)**  
   Une fonctionnalité permet à l'administrateur de créer et d'imprimer des reçus en format PDF, probablement pour documenter des transactions ou réservations.
3. **Gérer les types de chambres**  
   L'administrateur peut définir, modifier et organiser les types de chambres disponibles dans le système, telles que des suites ou des chambres standard.
4. **Envoyer des notifications par email**  
   Cette fonction permet d'envoyer des emails aux clients ou employés pour communiquer des informations importantes, comme des confirmations de réservation ou des mises à jour.
5. **Gérer les clients**  
   L'administrateur peut accéder et modifier les informations des clients, y compris leurs coordonnées et historique de réservation.
6. **Vérifier la disponibilité des chambres**  
   Une fonction essentielle permettant de vérifier quelles chambres sont libres ou occupées en temps réel.
7. **Gérer les réservations**  
   L'administrateur peut effectuer, modifier ou annuler des réservations, et consulter les détails associés.
8. **Exporter et importer des données (Excel, CSV)**  
   Cette fonction permet d'importer des données externes ou d'exporter des informations sous des formats standards (Excel, CSV), facilitant la gestion et le partage des données.
9. **Gérer les paiements**  
   L'administrateur a le contrôle des transactions financières, comme la validation des paiements effectués par les clients.

A diagram of a client

Description automatically generated

Ce diagramme illustre les fonctionnalités accessibles au client (utilisateur final) dans le système. Les cas d'utilisation reflètent les actions que le client peut effectuer pour interagir avec la plateforme. Voici une description des principales fonctionnalités :

1. **Générer des reçus au format PDF (pour Client)**  
   Le client a la possibilité de générer des reçus de ses transactions ou réservations sous format PDF, ce qui peut être utile pour des preuves d'achat ou de réservation.
2. **Créer un compte**  
   Cette fonctionnalité permet au client de s'inscrire sur la plateforme en créant un compte personnel. Cela peut inclure la saisie de ses informations personnelles pour accéder à des services personnalisés.
3. **Réserver des chambres (avec confirmation par email)**  
   Le client peut effectuer des réservations de chambres via le système. Une fois la réservation confirmée, il reçoit un email de confirmation, assurant un suivi efficace.

**Résumer :**

Les diagrammes de cas d'utilisation présentent les principales interactions possibles avec le système, selon deux rôles principaux : l'Administrateur et le Client.

1. Rôles et Fonctions de l'Administrateur :

L'administrateur dispose de nombreuses fonctionnalités lui permettant de gérer le système efficacement. Parmi celles-ci :

* Gestion des employés, clients, types de chambres et réservations.
* Vérification de la disponibilité des chambres.
* Génération de reçus PDF et gestion des paiements.
* Import/export de données (Excel, CSV).
* Envoi de notifications par email.

Ces fonctionnalités visent à assurer un contrôle global du système et une gestion fluide des opérations.

2. Rôles et Fonctions du Client :

Le client, en tant qu'utilisateur final, interagit avec le système via des fonctionnalités simplifiées et adaptées :

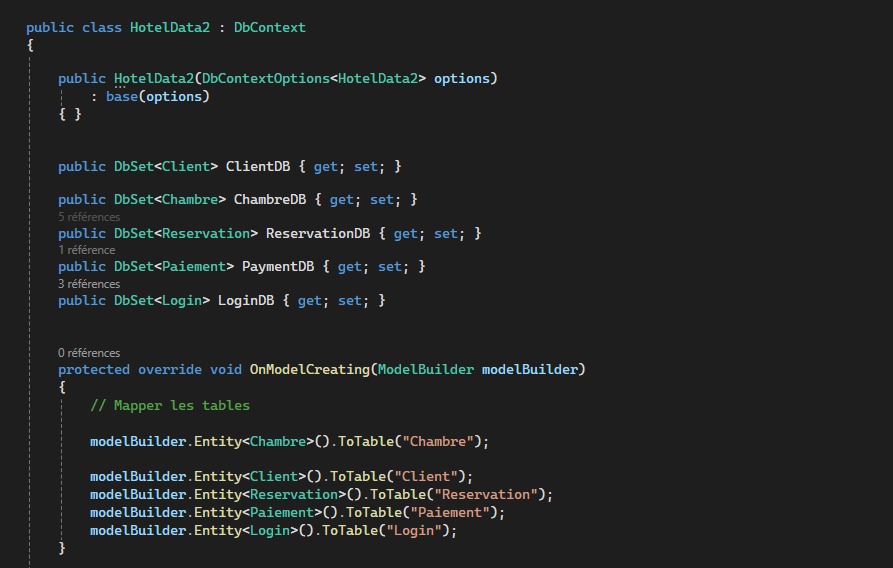
* Création d’un compte pour accéder aux services.
* Réservation de chambres, avec confirmation par email.
* Génération de reçus PDF pour documenter ses transactions.

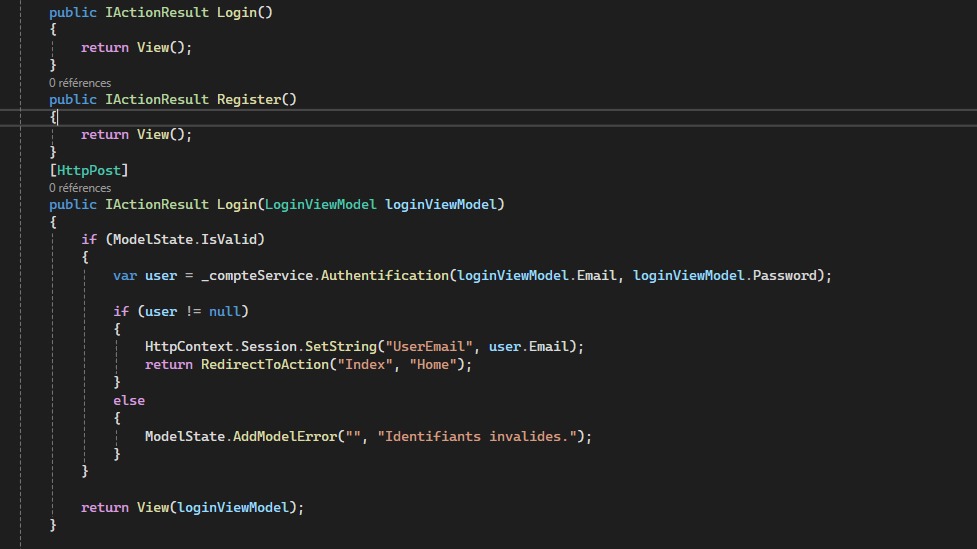
Ces fonctionnalités garantissent une expérience utilisateur intuitive et professionnelle.

# 6.Resultat finale :

## 6.1 Interfaces utilisateur

Dans cette section, nous expliquons comment la base de données a été connectée avec le code pour permettre une interaction fluide entre le backend et les données de l'application.





Implémentation de l'Architecture MVC :

L'architecture **MVC (Model-View-Controller)** a été utilisée pour organiser le projet, garantissant une séparation claire entre la logique métier, l'interface utilisateur et la gestion des données. Cela facilite la maintenance, l'évolutivité et la compréhension du code. Voici comment chaque composant a été mis en œuvre :

**1. Models**

Les modèles représentent les données et la logique métier de l'application. Les classes suivantes ont été créées pour refléter la structure de la base de données :

* **Chambre.cs** : Contient les propriétés liées aux chambres, telles que le numéro, la disponibilité, etc.
* **Client.cs** : Gère les informations sur les clients, y compris leur nom, email, téléphone, etc.
* **Login.cs** : Modèle pour la gestion des informations d'identification des utilisateurs.
* **Paiement.cs** : Décrit les détails de paiement, comme le montant et la méthode utilisée.
* **Reservation.cs** : Contient les propriétés pour gérer les réservations.

**2. Views**

Les vues se trouvent dans le dossier Views et sont responsables de l'affichage des données aux utilisateurs. Par exemple :

* **Compte** : Pour les pages liées à la gestion du compte utilisateur(Login et registation).
* **Home** : Pour l'affichage des pages d'accueil ou générales.

**3. Controllers**

Les contrôleurs gèrent la logique de l'application et les interactions entre les modèles et les vues. Les principaux contrôleurs incluent :

* **CompteController.cs** : Gère les actions liées au compte utilisateur, comme l'inscription et la connexion.
* **PaymentController.cs** : Responsable des opérations liées au paiement.
* **ReservationController.cs** : Gère la logique des réservations.

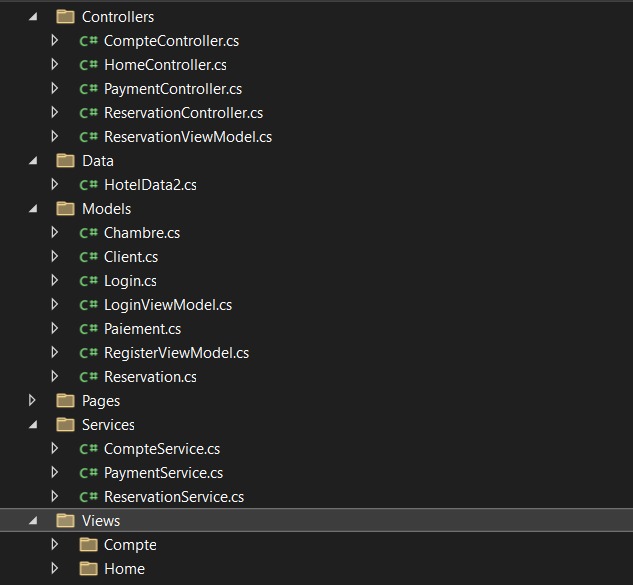
**4. Services**

Les services encapsulent la logique métier complexe pour une meilleure réutilisabilité. Par exemple :

* **CompteService.cs** : Fournit des fonctions pour gérer les comptes.
* **PaymentService.cs** : Centralise la gestion des paiements.
* **ReservationService.cs** : Regroupe la logique associée aux réservations.

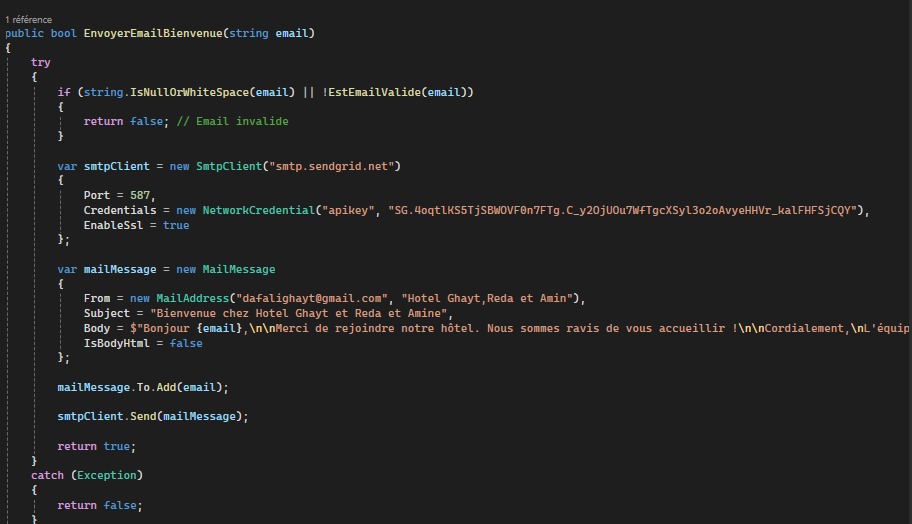
**5. Data**

Le fichier HotelData2.cs contient la configuration pour interagir avec la base de données.



Envoi d’email :

Pour l'envoi des emails, on a utilisé le service SMTP de **SendGrid** (<http://smtp.sendgrid.net/>) afin de garantir que les messages arrivent directement dans la boîte de réception de l'utilisateur et non dans le dossier spam. SendGrid est une plateforme fiable qui permet d'envoyer des emails en toute sécurité et efficacement.

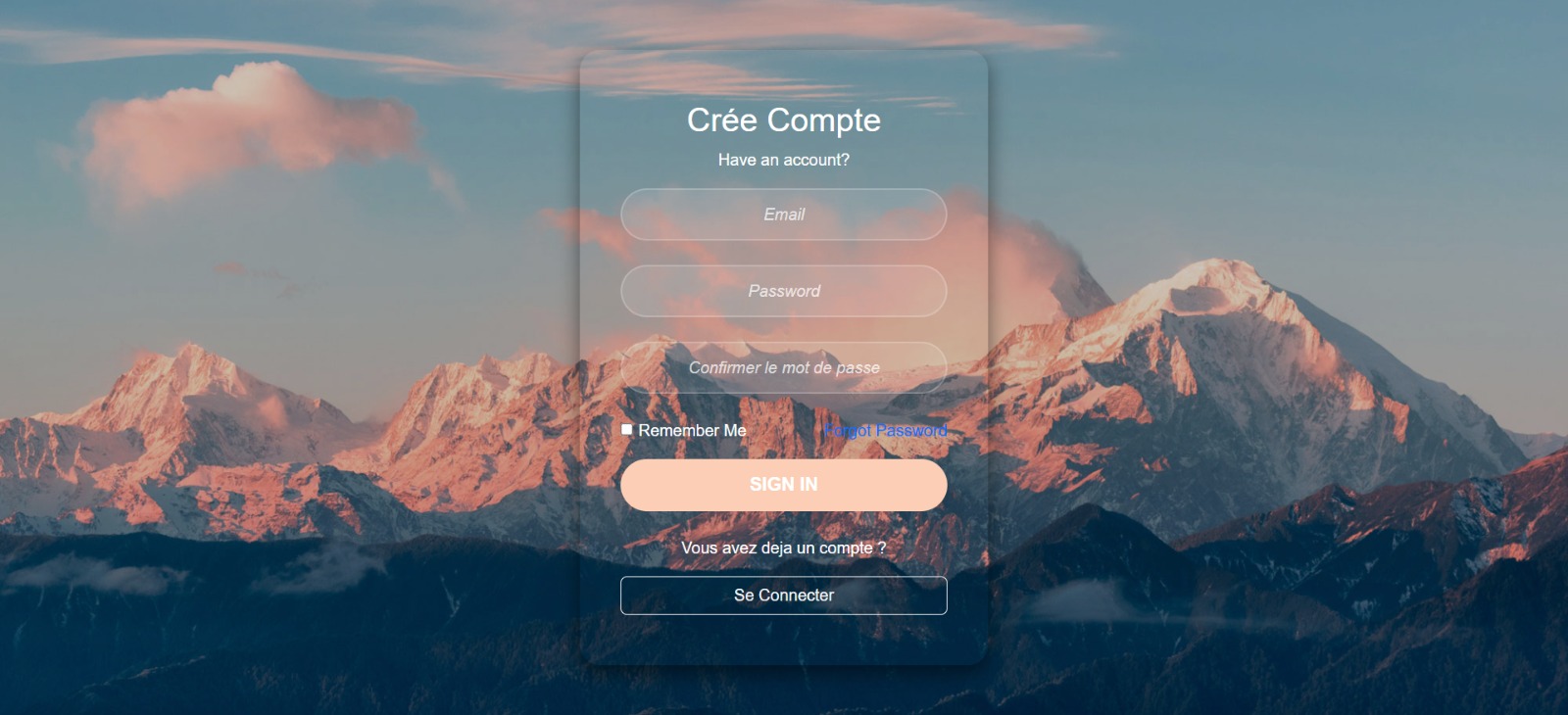


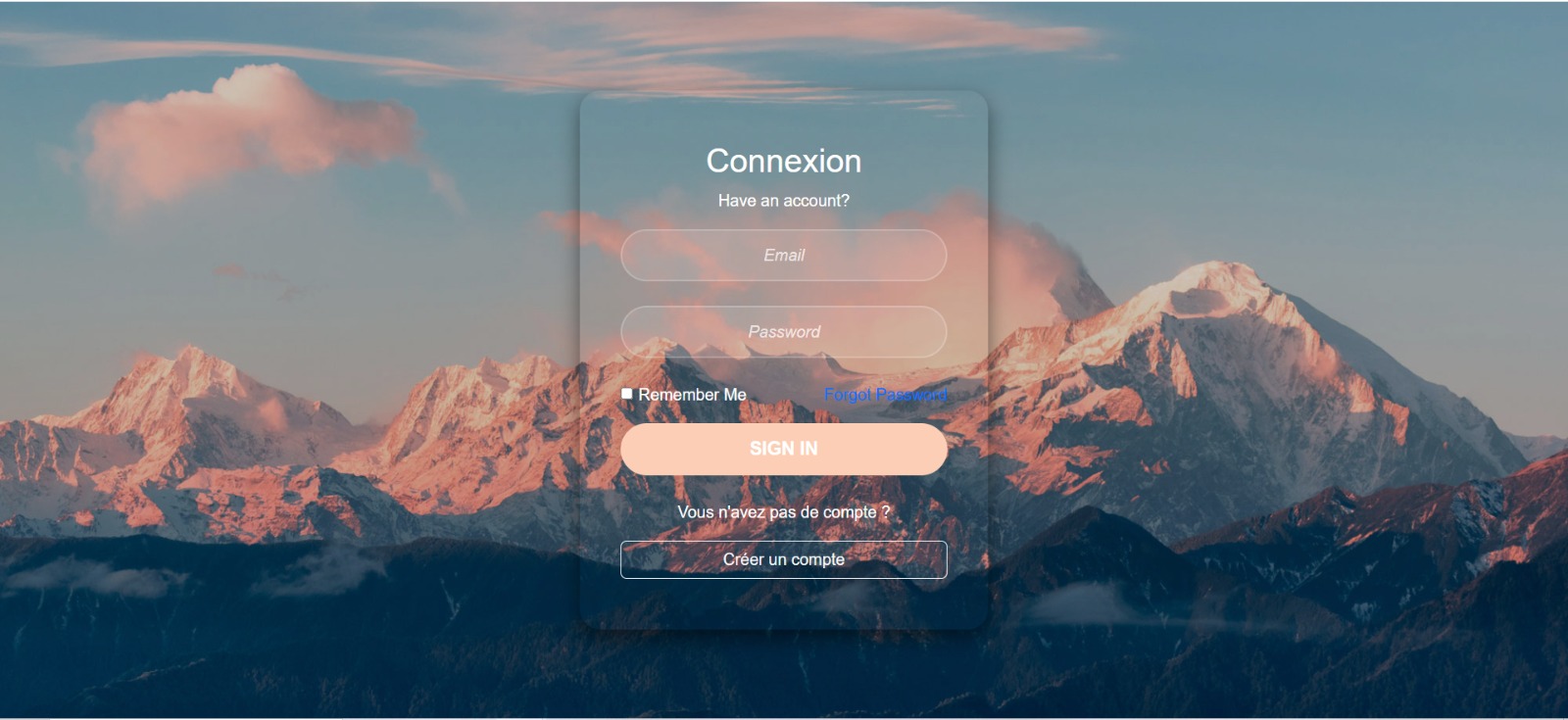
En configurant ces paramètres, j'ai pu envoyer des emails directement à l'adresse du destinataire en utilisant le service de SendGrid. Ce processus garantit que les emails sont envoyés de manière sécurisée et réduisent considérablement la probabilité que les messages finissent dans le dossier spam.

J'ai aussi veillé à utiliser une adresse email professionnelle pour l'expéditeur, ce qui renforce la crédibilité de l'email. Pour éviter toute mise en spam, j'ai configuré les enregistrements SPF (Sender Policy Framework) et DKIM (DomainKeys Identified Mail) pour mon domaine, ce qui permet d'authentifier les messages envoyés et d'assurer qu'ils ne sont pas bloqués par les filtres anti-spam.

## 6.1 Interface utiisateur :

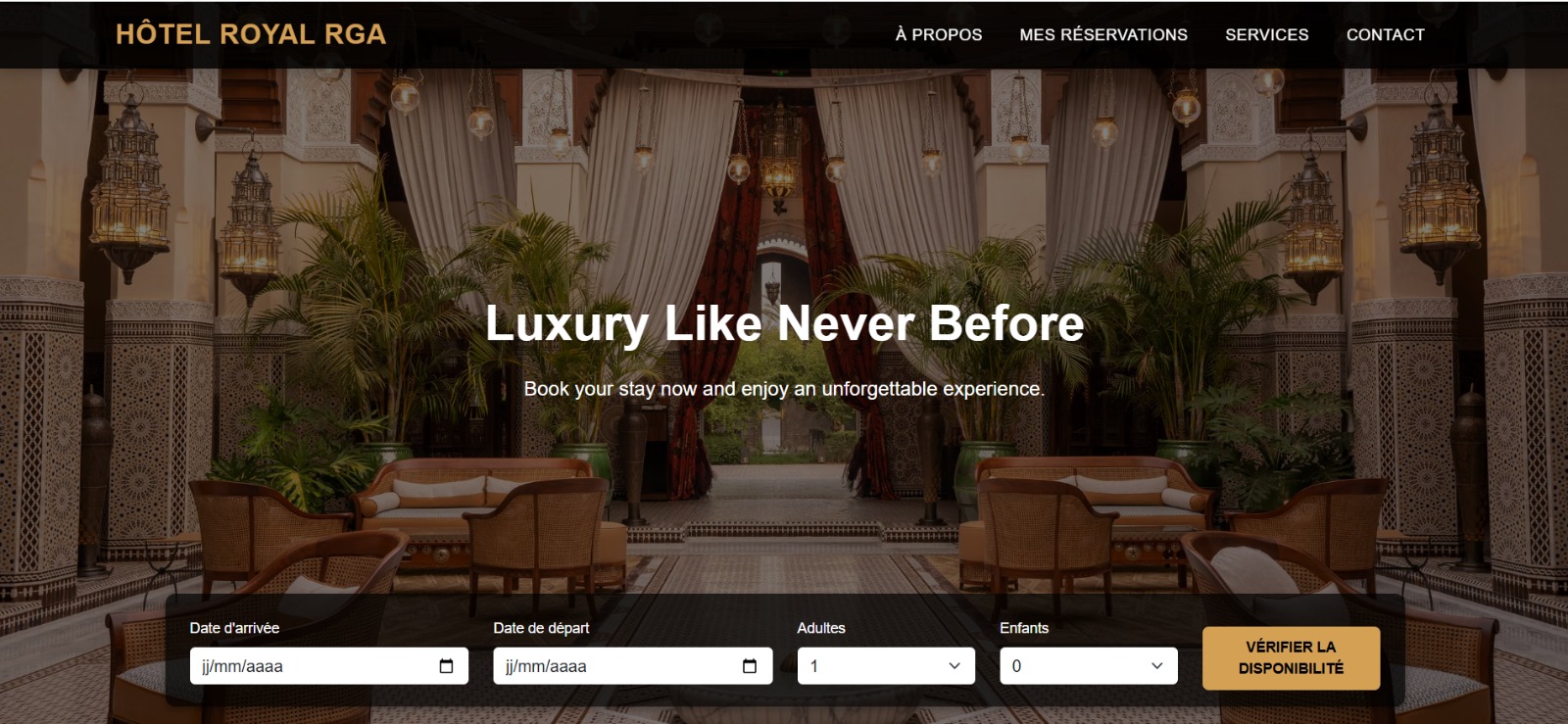
Partie WEB :



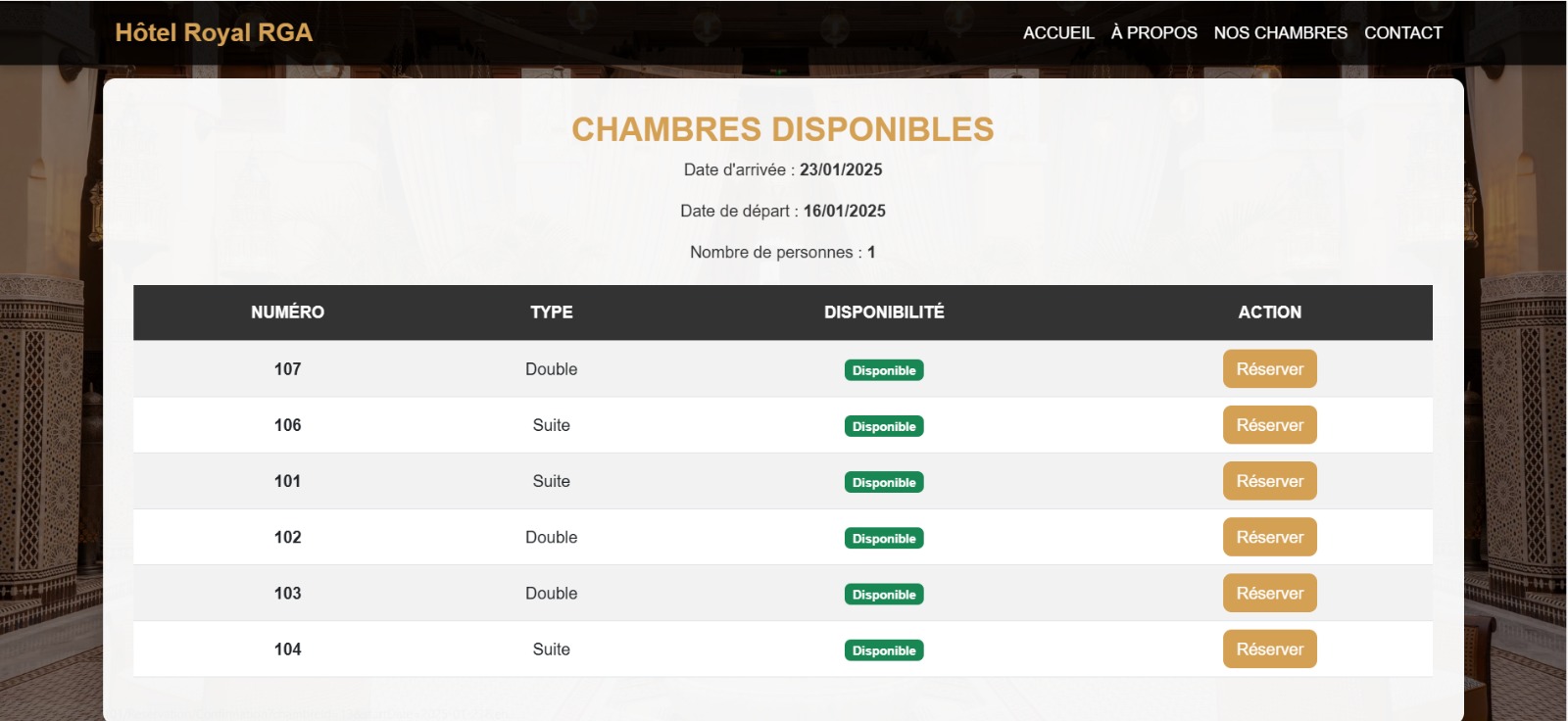


La première capture d'écran montre la page d'inscription, où l'utilisateur peut créer un compte en entrant ses informations personnelles telles que le nom, l'adresse email et le mot de passe. Ce formulaire permet de collecter les données nécessaires pour créer un compte et enregistrer l'utilisateur dans la base de données.

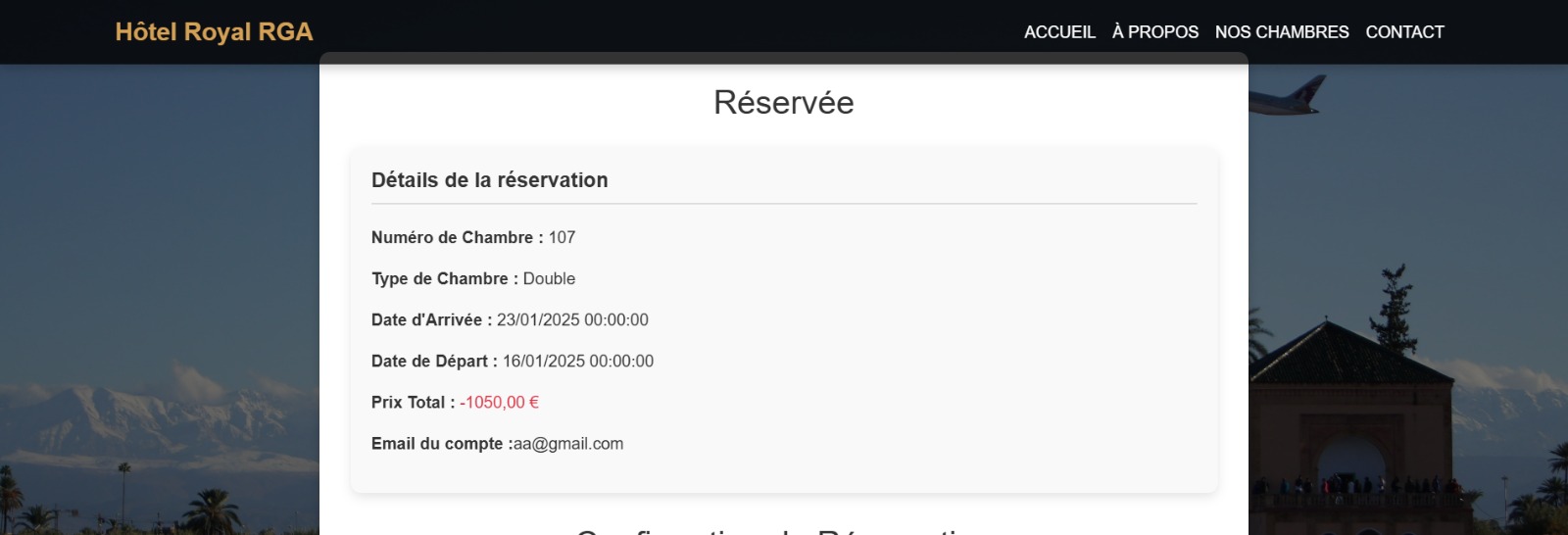
La deuxième capture d'écran illustre la page de connexion, où l'utilisateur peut se connecter en utilisant son adresse email et son mot de passe. Après validation des informations par rapport à la base de données, l'utilisateur est authentifié et accède à son compte.



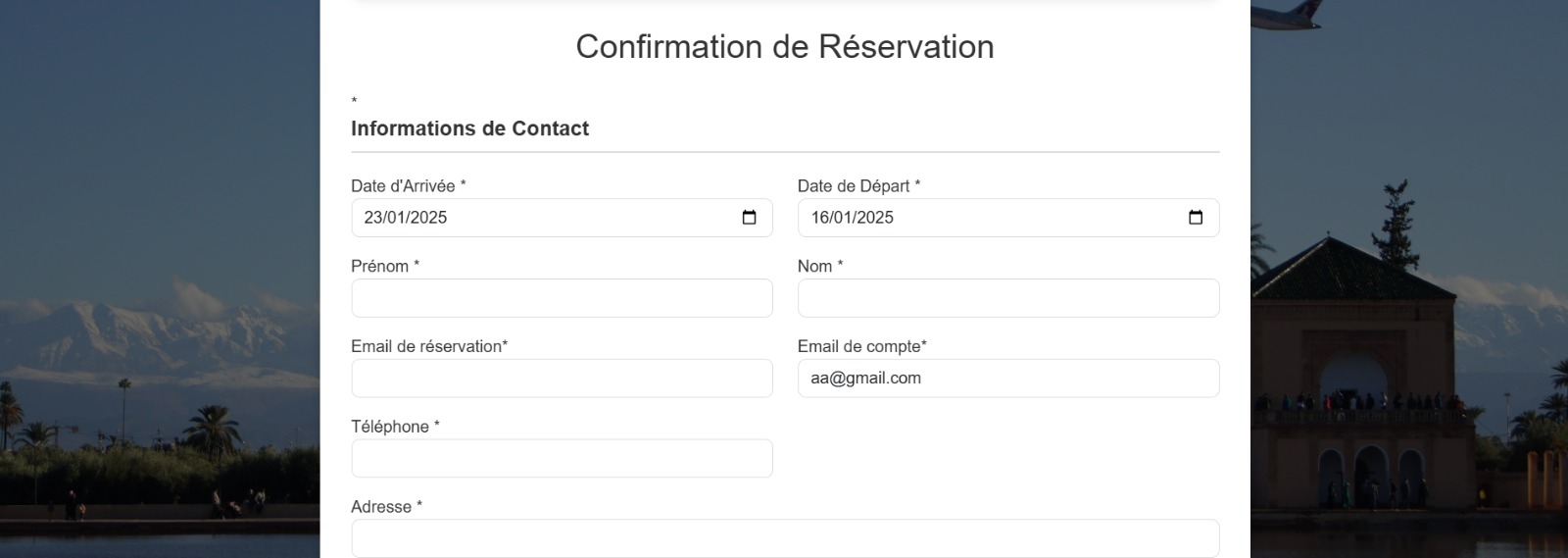
Cette page permet aux client de filtrer les chambre qui sont disponible a une date choisie .



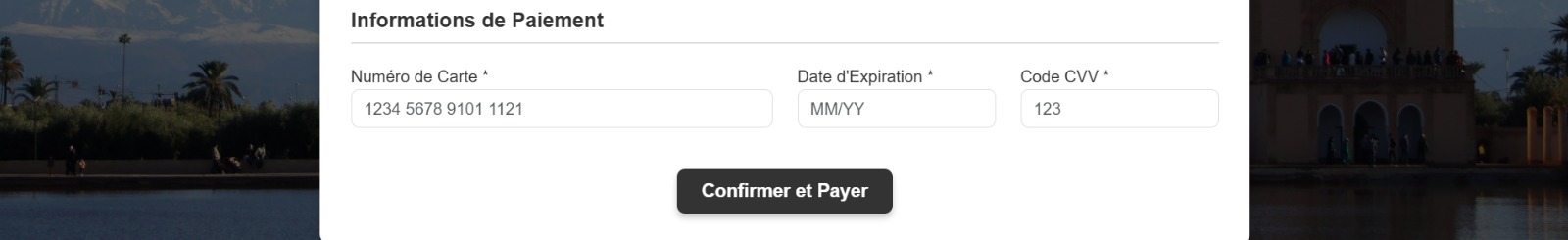
Après quand le client remplis le formulaire des date il navigue vers une autre page qui affiche toutes les chambres disponible et le choix reviens aux client de choisir une chambre .



Quand le client choisit sa chambre, il passe vers une autre page, celle du paiement.  
Cette page se divise en trois parties.  
Concernant la première partie, c'est celle des détails de la réservation qui affiche le numéro de la chambre, le type, la date d'arrivée et la date de départ. Ainsi, il calcule le prix à payer en fonction des jours comptés et l'email du compte associé à la réservation.  
On va voir les autres parties par la suite.



Deuxieme partie : On voit un formulaire où le client doit remplir son prénom, son nom, ainsi que l’email de la réservation, puis son numéro de téléphone et son adresse.  
Sur la partie de code, nous avons mis en place certaines conditions : si l’email de la réservation existe déjà dans notre base de données, on vérifie si le client est archivé ou non. Si c’est le cas, on le désarchive. Mais si l’email n’existe pas, on ajoute un nouveau client avec cet email.  
Pour l'email de compte, cela signifie que ce compte peut passer plusieurs réservations avec des emails différents, et chaque email de réservation est considéré comme un client.



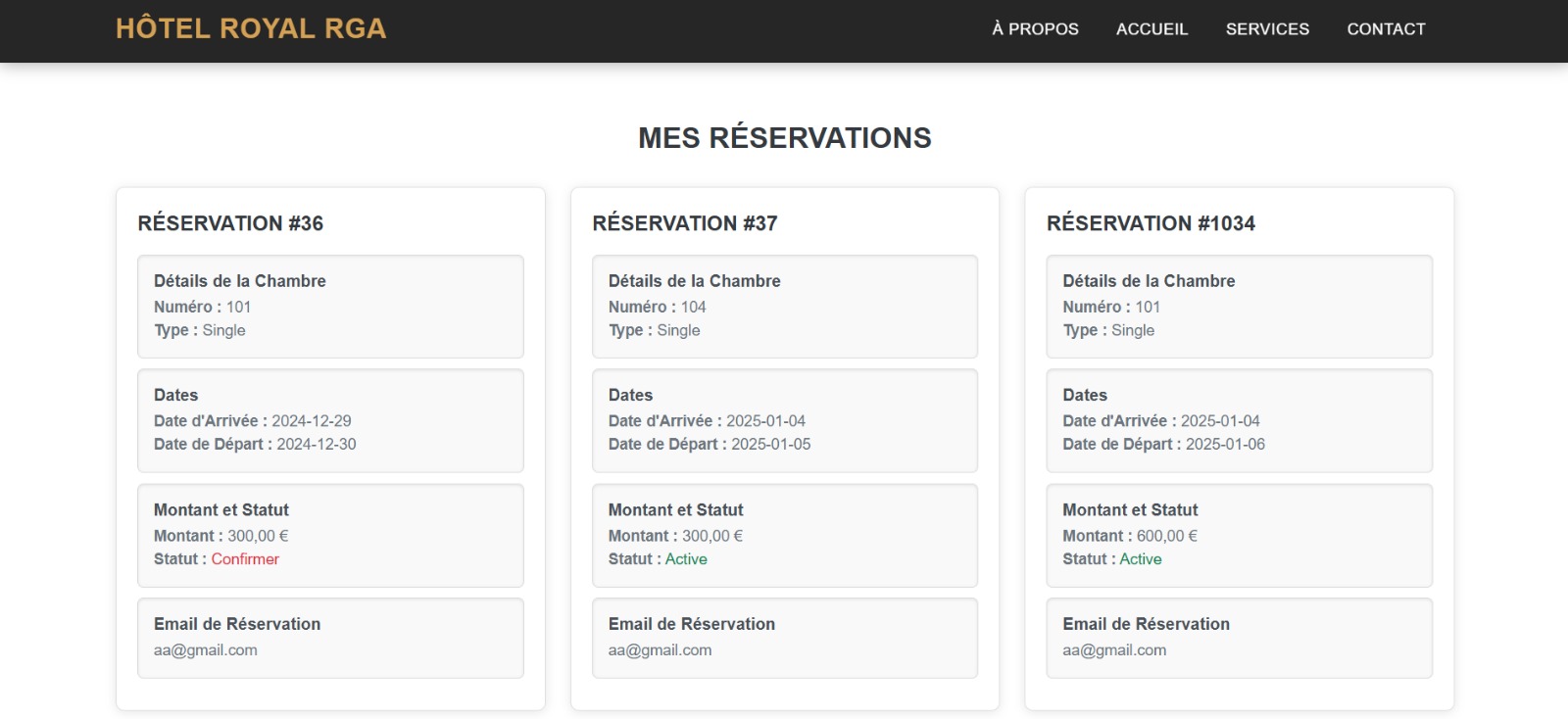
Dans cette section, nous passons au paiement. Nous n'avons pas vérifié si la carte est valide ou non, nous avons juste mis en place le processus de paiement pour montrer comment il se déroule.  
Enfin, après avoir cliqué sur le bouton de confirmation, une réservation est créée et un email est envoyé. Cet email contient les informations de la réservation, ainsi qu'un PDF généré avec les détails.

Une image contenant texte, capture d’écran

Description générée automatiquement

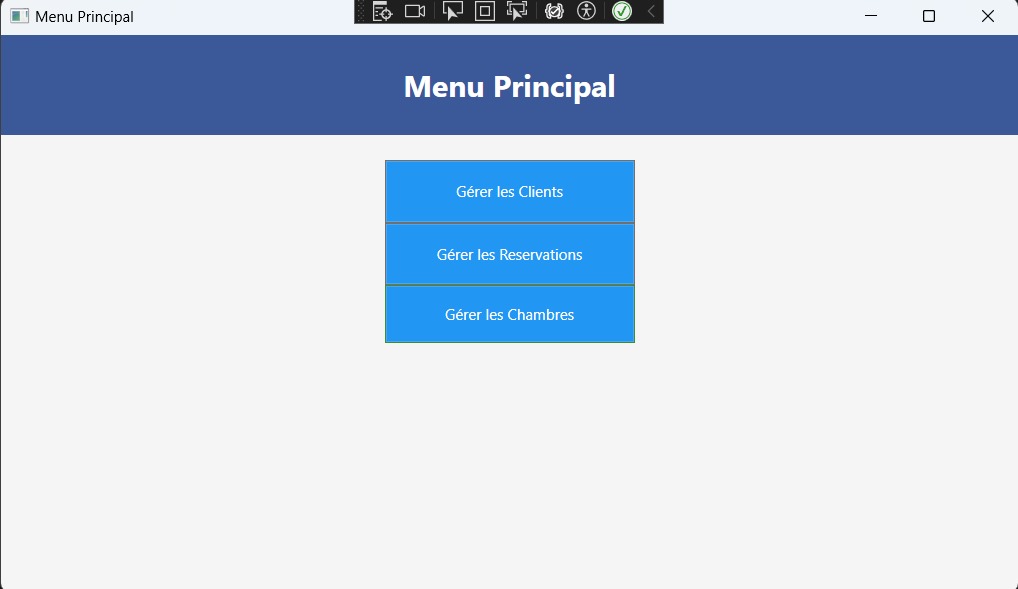
La capture d’écran illustre l’email de confirmation reçu par le client après avoir effectué une réservation. Cet email contient tous les détails importants, tels que le code de réservation, les dates d’arrivée et de départ, le montant total, ainsi que le mode de paiement. Un fichier PDF reprenant ces informations est également joint à l’email, offrant une version téléchargeable et imprimable au client.

L’envoi de cet email est réalisé à l’aide du protocole SMTP (Simple Mail Transfer Protocol) en passant par le service **SendGrid**, qui fournit une API pour une transmission sécurisée des emails. Le service SMTP utilise le port 587 avec une connexion chiffrée et une authentification par clé API, garantissant ainsi la fiabilité et la sécurité de l’envoi. Cette approche assure une communication claire et professionnelle entre l’hôtel et ses clients.



Cette page affiche toutes les réservations passées par ce compte du client, en affichant les détails de chaque réservation avec un statut.  
Le statut "actif" peut signifier "en attente" ou "en cours", tandis que "confirmé" indique que la réservation est confirmée (terminée). Enfin, il y a le statut "annulé".  
Pour la gestion des statuts, ces derniers sont modifiés manuellement depuis la partie desktop et le client n’as pas le rôle de modifier sa réservation.

**Partie desktop :**

****

L'interface Admin Manager permet à l'administrateur de gérer efficacement les différentes fonctionnalités liées aux réservations, aux clients et aux chambres. Voici les principales fonctionnalités qu'il peut modifier ou gérer :

1. Gérer les Réservations

L'administrateur peut voir toutes les réservations effectuées, les modifier si nécessaire ou les supprimer. Cela permet de garantir que les informations de réservation sont toujours à jour et bien organisées.

* Modifier une réservation : L'administrateur peut changer les dates, le type de chambre ou ajouter des informations supplémentaires.
* Supprimer une réservation : Si nécessaire, l'administrateur peut annuler une réservation.

2. Gérer les Clients

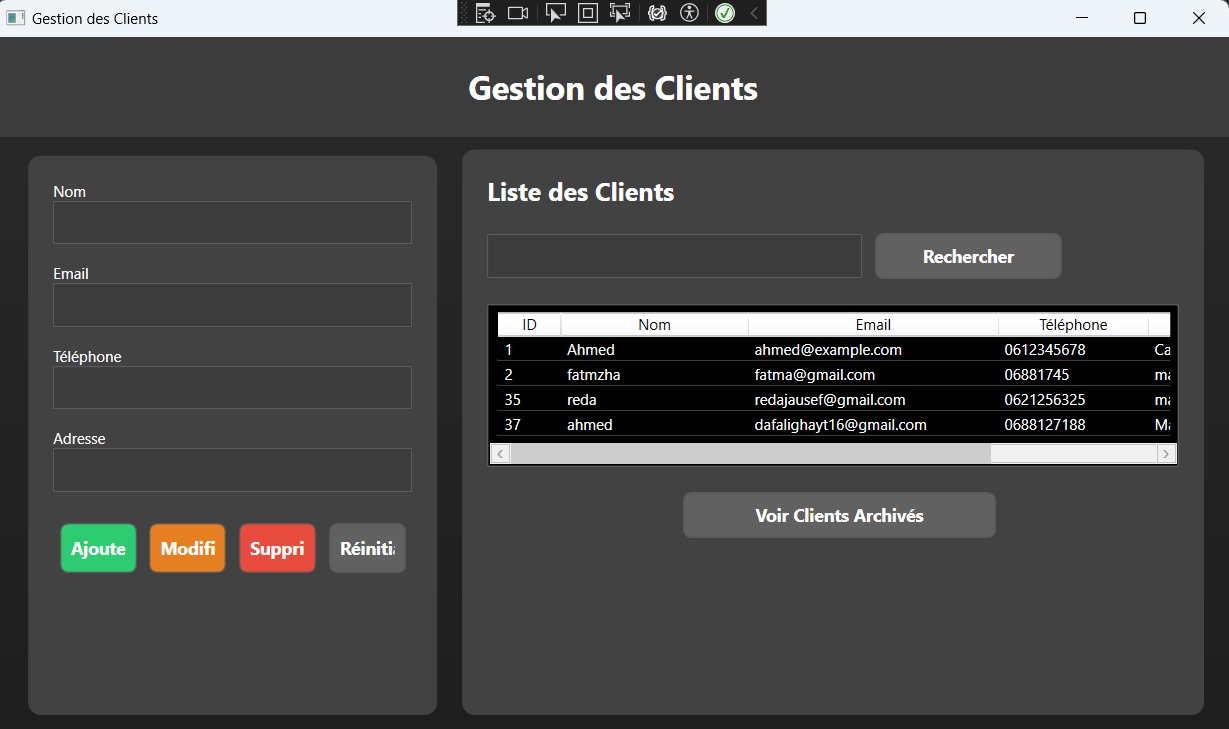
L'administrateur peut consulter les informations des clients, comme leur nom, email, et historique de réservation. Cela permet une gestion efficace des utilisateurs et une assistance rapide.

* Modifier les informations du client : L'administrateur peut mettre à jour les coordonnées ou d'autres détails du client.
* Supprimer un client : Si un client n'est plus actif ou souhaite être supprimé, l'administrateur peut gérer cette action.

3. Gérer les Chambres

L'administrateur peut gérer l'inventaire des chambres disponibles, leur disponibilité et les détails associés. Cela permet une gestion dynamique des chambres en fonction de la demande.

* Ajouter/modifier/supprimer une chambre : L'administrateur peut ajouter une nouvelle chambre, mettre à jour ses détails ou supprimer une chambre.
* Mettre à jour la disponibilité : L'administrateur peut ajuster la disponibilité des chambres selon les réservations existantes.

****

Dans cette section, l'administrateur peut gérer efficacement les informations des clients, telles que leur nom, email, numéro de téléphone, et adresse. Voici les principales fonctionnalités disponibles pour l'administrateur :

1. Recherche de Client

L'administrateur peut rechercher un client en utilisant différents critères, tels que son nom, son email ou son numéro de téléphone. Cela permet de trouver rapidement un client spécifique et d’accéder à ses informations ou à son historique de réservation.

2. Ajouter un Client

L'administrateur peut ajouter de nouveaux clients à la base de données en saisissant leurs informations personnelles telles que le nom, l'email, le téléphone, et l'adresse. Une fois ajoutés, ces clients pourront effectuer des réservations et être suivis dans le système.

3. Modifier un Client

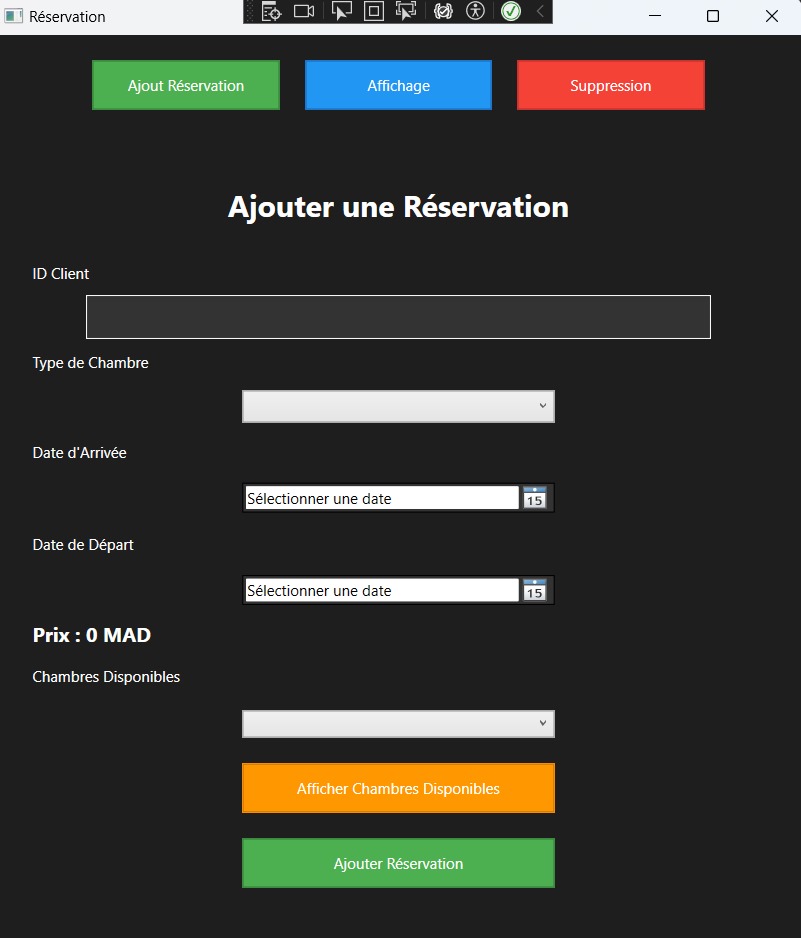
L'administrateur peut mettre à jour les informations d'un client existant. Cela inclut la possibilité de modifier les coordonnées (nom, email, téléphone, adresse) si ces informations sont incorrectes ou ont changé.

4. Supprimer un Client

Si un client n'est plus actif ou demande la suppression de ses informations, l'administrateur peut supprimer ce client de la base de données nous le cas ou le client est supprimer définitivement de la base de données c’est que il n’a jamais passer une réservations il est considérer comme non client, mais si le client a des réservations et on doit procéder a sa suppression il passe dans un état archiver .

5. Réinitialiser les Informations

L'administrateur peut réinitialiser certains détails du client, comme réinitialiser son mot de passe ou ses préférences, en cas de besoin.

****

Cette image présente une interface graphique d'un système de gestion de réservations. L'interface est divisée en plusieurs sections et boutons, chacun ayant une fonction spécifique. Voici une description des éléments visibles :

1. Boutons en haut de l'interface :
   * Ajout Réservation (vert) : Permet d'accéder à l'interface pour ajouter une nouvelle réservation.
   * Affichage (bleu) : Permet de visualiser les réservations existantes.
   * Suppression (rouge) : Permet de supprimer une réservation existante.
2. Section "Ajouter une Réservation" :
   * ID Client : Un champ de texte où l'utilisateur peut entrer l'identifiant du client.
   * Type de Chambre : Un menu déroulant permettant de sélectionner le type de chambre.
   * Date d'Arrivée : Un sélecteur de date pour choisir la date d'arrivée du client.
   * Date de Départ : Un sélecteur de date pour choisir la date de départ du client.
   * Prix : Affiche le prix total de la réservation, initialement à "0 MAD" (monnaie marocaine, dirham).
   * Chambres Disponibles : Un champ qui semble afficher les chambres disponibles une fois recherchées.
3. Boutons en bas de l'interface :
   * Afficher Chambres Disponibles (orange) : Permet de rechercher et d'afficher les chambres disponibles en fonction des critères de réservation.
   * Ajouter Réservation (vert) : Permet de confirmer et d'ajouter la réservation dans le système.

Cette interface est conçue pour faciliter la gestion des réservations en permettant aux utilisateurs de saisir les détails nécessaires, de vérifier les disponibilités et d'ajouter de nouvelles réservations.

**Une image contenant texte, capture d’écran, logiciel, conception

Description générée automatiquement**

Cette image représente une interface utilisateur pour la gestion des chambres, probablement dans un système de gestion hôtelière. Voici une description détaillée des différents éléments de l'interface :

1. Titre de l'interface :
   * "Gérer les Chambres" : Indique que cette section est dédiée à la gestion des chambres disponibles dans l'établissement.
2. Champs de saisie :
   * Numéro de Chambre : Un champ de texte où l'utilisateur peut entrer le numéro de la chambre à gérer.
   * Type de Chambre : Un menu déroulant pour sélectionner le type de chambre (par exemple, simple, double, suite).
   * Statut de Chambre : Un menu déroulant permettant de définir le statut de la chambre (par exemple, disponible, occupée, en maintenance).
3. Boutons d'action :
   * Ajouter Chambre (vert) : Permet d'ajouter une nouvelle chambre avec les informations saisies.
   * Afficher Chambres (orange) : Permet de visualiser la liste des chambres existantes.
   * Supprimer Chambre (rouge) : Permet de supprimer une chambre existante à partir du système.
4. Zone d'affichage :
   * Une section, probablement dédiée à l'affichage des informations sur les chambres, est située en bas de l'interface. Elle pourrait contenir une liste ou des détails des chambres gérées via cette interface.

Cette interface est conçue pour permettre aux utilisateurs de gérer les chambres de manière efficace, en leur offrant des options pour ajouter, afficher et supprimer des chambres en fonction de leurs besoins.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Sur cette interface, nous avons plusieurs options, telles que : voir toutes les réservations, trier et afficher les réservations payées et non payées, modifier le paiement des réservations (les marquer comme payées ou non payées), et filtrer les réservations d’un client par un ID.

**Une image contenant capture d’écran, texte, logiciel, Logiciel multimédia

Description générée automatiquement**

Cette partie permet aux admins de supprimer les réservations pour libérer les chambres associées en cas d'annulation. Nous avons plusieurs conditions pour accéder à la suppression : d'abord, si la réservation est terminée, elle peut être supprimée, mais pas de la base de données ; son statut sera simplement modifié en "confirmé", ce qui signifie qu'elle est non active. Si la réservation est en cours, elle ne peut pas être supprimée car le client occupe déjà la chambre. Enfin, si la réservation est active et en attente, elle peut être supprimée en modifiant son statut à "annulé", afin de préserver l'historique des réservations.

# Conclusion

Ce projet, réalisé en groupe dans le cadre de notre formation, a permis de développer une application de gestion des réservations pour un hôtel, ainsi qu'un site web destiné aux clients pour effectuer leurs réservations en ligne. Nous avons utilisé les technologies **WPF**, **.NET Core**, **ASP.NET Core MVC**, **SSMS** et **Entity Framework** dans le but d'acquérir une expérience concrète dans le développement d'applications, la gestion des bases de données et l'intégration de systèmes complexes.

L'application permet de gérer facilement les réservations, avec des fonctionnalités telles que la consultation, la modification, l'annulation des réservations et la gestion du paiement. Du côté administrateur, nous avons mis en place des options pour filtrer et modifier les réservations selon différents critères. Le site web permet aux clients de réserver des chambres en ligne, de consulter leur historique et d'effectuer des paiements de manière sécurisée.

Bien que ce projet ne soit pas destiné à être déployé dans un cadre professionnel, il nous a permis de mettre en pratique nos connaissances et de développer nos compétences dans un environnement réel de développement logiciel. L'utilisation de **ASP.NET Core MVC** et **Entity Framework** nous a permis de structurer l'application de manière modulaire et d'assurer une gestion efficace des données via **SSMS**.

Ce projet a également été une excellente occasion de travailler en équipe, de partager nos idées et de coordonner nos efforts pour parvenir à une solution fonctionnelle et cohérente. Chaque membre du groupe a contribué à la réalisation des différentes parties de l’application, ce qui a permis d'améliorer nos compétences en collaboration et en gestion de projet.

En conclusion, bien que ce projet soit destiné à des fins scolaires et ne soit pas prévu pour un déploiement professionnel, il représente un excellent exercice pour renforcer nos compétences techniques et nous préparer aux défis futurs dans notre parcours de développement logiciel. Cette expérience a été enrichissante, tant sur le plan technique que sur le plan de la collaboration en équipe, et elle nous a permis de mieux appréhender les étapes du cycle de vie d'un projet logiciel.

**Lien Vers GitHub :**

**https://github.com/ghayted/SiteHotelll**