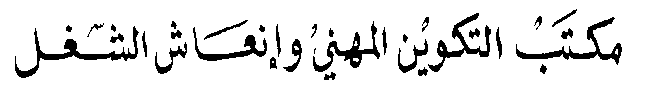
****

**Office de la Formation Professionnelle**

**et de la Promotion du Travail**

**OFPPT**

**Rapport du projet de fin d’étude.**

**Présenté en vue d'obtenir le diplôme technicien spécialisé**

**en développement informatique.**

**Présenté par :   
El Ferdaous Ayoub**

**Application de gestion en Web**

**‘Gestion d’un Pharmacie’**

**Encadré par**

**M. Fouad Essarraj**

**2019 – 2020**

**Sommaire**

[I. Introduction: 3](#_Toc41724530)

[II. Remerciements : 4](#_Toc41724531)

[I. Menu Principal : 5](#_Toc41724532)

[II. Architecture 3 tiers: 5](#_Toc41724533)

[III. Analyse et conception : 6](#_Toc41724534)

[1. Introduction : 6](#_Toc41724535)

[2. Cas d’utilisation : 6](#_Toc41724536)

[3. Classes utilisé : 7](#_Toc41724537)

[IV. Codage 10](#_Toc41724538)

[Introduction : 10](#_Toc41724539)

[1. Langages et Framework utilises : 10](#_Toc41724540)

[a. ASP.NET MVC : 10](#_Toc41724541)

[b. Model-View-Controller : 10](#_Toc41724542)

[c. Entity Framework : 11](#_Toc41724543)

[V. Présentation 12](#_Toc41724544)

[1. Introduction 12](#_Toc41724545)

[2. Présentation des exemples des interfaces 12](#_Toc41724546)

[VI. Conclusion générale 19](#_Toc41724547)

# Dédicace:

**Je dédie ca travail :**

**A ma famille, elle qui m’a doté d’une éducation digne, son amour a fait de moi ça que je suis aujourd’hui :**

**Particulièrement à mes chers parents et mes sœurs pour le soutien.**

**A tous mes amis, et a tous ceux que j’aime et a toutes les personnes qui se sont données la peine de me soutenir durant ce projet.**

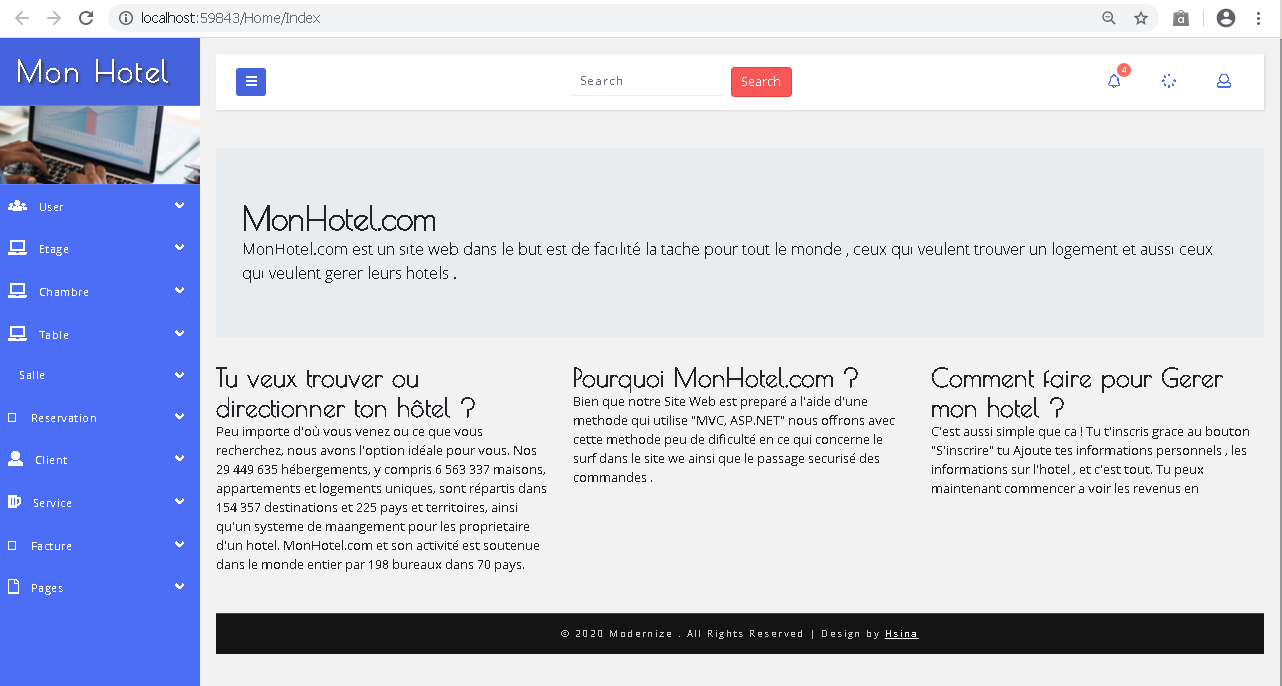
# Remerciements :

**Au terme de ce travail je tiens tout d’abord à remercier :**

**Mon DIEU de m’avoir donné le courage, la force et la volonté pour achever ce modeste travail.**

**Mon encadreur M. Essarraj Fouad, enseignant à ISMONTIC, pour ses qualités professionnelles, ses conseils son encadrement, ses remarques constructives tout le long de mon travail.**

# Menu Principal :



# NSY107 - Intégration des systèmes client-serveur - ppt téléchargerArchitecture 3 tiers:

L'**architecture trois tiers**est l'application du modèle plus général qu'est le multi-tiers. L'architecture logique du système est divisée en trois niveaux ou couches :

* couche de *présentation*
* couche de *traitement*
* couche d'*accès aux données*

C'est une architecture basée sur l'environnement client-serveur

# Analyse et conception :

## Introduction :

**Le but de cette étape dans le rapport est pour simplifier la lecture du rapport pour tout le monde.**

## Cas d’utilisation :

**MonHotel.com**

Créer un compte

Gérer tous le bâtiment d’hôtel

Reserver une chambre

Suivre les revenues ainsi que les dépenses

Reserver une table dans le restaurant de l’hôtel

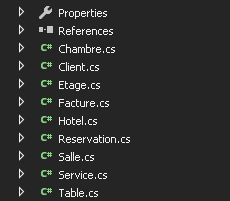
Utilisateur

Admin

Reserver une salle  
« Conférence ou fête»

Demander un service supplémentaire

## Classes utilisé :



**Chambre :**

public class Chambre

{

[Key,Required]

public string Num\_Chambre { get; set; }

[Range(1,3)]

public int NbrPlaces { get; set; }

public float Prix { get; set; }

public virtual Etage Etage { get; set; }

}

**Client:**

public class Client

{

[Key, Required]

public string CNI { get; set; }

public string Nom { get; set; }

public string Prénom { get; set; }

public string Nationalite { get; set; }

public string Adresse { get; set; }

[DataType(DataType.PhoneNumber)]

public string Telephone { get; set; }

[EmailAddress]

public string Email { get; set; }

public virtual ICollection<Reservation> Reservations { get; set; }

}

**Etage:**

public class Etage

{

[Key,Required]

public string Num\_Etage { get; set; }

}

**Facture:**

public class Facture

{

[System.ComponentModel.DataAnnotations.Key]

public string Id\_Facture { get; set; }

public System.DateTime DateFacture { get; set; }

public float Montant { get; set; }

public bool Payer { get; set; }

public virtual Client Client { get; set; }

}

**Hôtel:**

public class Hotel

{

[Key,Required]

public string Nom\_Hotel { get; set; }

public string Adresse { get; set; }

[DataType(DataType.PhoneNumber)]

public string Télephone { get; set; }

[DataType(DataType.EmailAddress)]

public string Email { get; set; }

}

**Reservation:**

[Table("Reservation")]

public class Reservation

{

[Key]

public string Id\_Reservation { get; set; }

[DataType(DataType.Date)]

public DateTime DateReservation { get; set; }

[DataType(DataType.Date)]

public DateTime DateArrive { get; set; }

[DataType(DataType.Date)]

public DateTime DateDepart { get; set; }

[Range(1,2)]

public int NbrPersonnes { get; set; }

public virtual Client Client { get; set; }

public virtual Service Service { get; set; }

public virtual Chambre Chambre { get; set; }

public virtual Salle Salle { get; set; }

public virtual Table Table { get; set; }

}

**Salle:**

public class Salle

{

[Key,Required]

public string Num\_Salle { get; set; }

public int Nbr\_Pirsonnes { get; set; }

public float Prix { get; set; }

public virtual Etage Etage { get; set; }

}

**Service:**

public class Service

{

[Key,Required]

public string TypeService { get; set; }

public float Prix { get; set; }

}

**Table :**

public class Table

{

[Required,Key]

public string Num\_Table { get; set; }

public int Nbr\_Places { get; set; }

public float Prix { get; set; }

}

# Codage

Introduction :   
**Ce chapitre a pour but de présenté tous les logiciels et sources utilisé pour le développement de cette application.**

## Langages et Framework utilises :

### ASP.NET MVC :

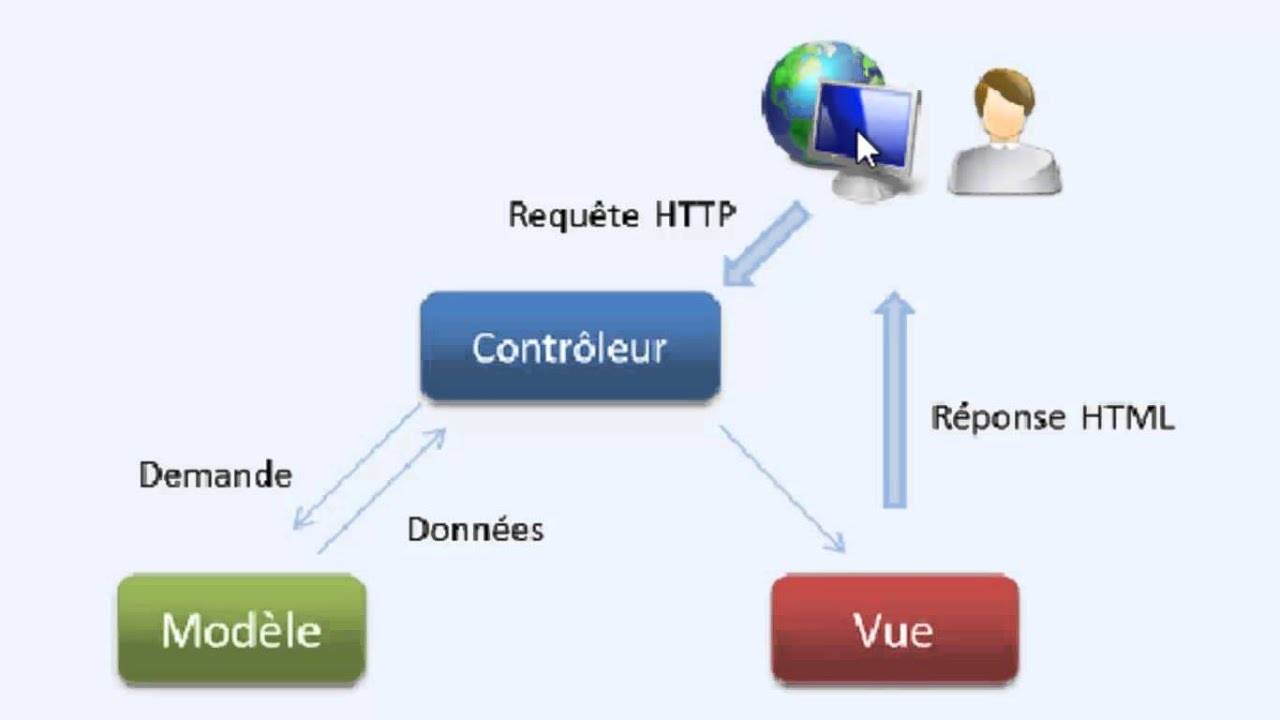
**ASP.NET MVC est un Framework de programmation des application web en style MVC ajouté à ASP.NET en 2009. Il a été créé en 2007 par Scott Guthrie, un des auteurs d'ASP.Net puis a été incorporé à ASP.NET et est devenu un produit officiel de Microsoft avec la sortie d’ASP.Net MVC.**

### Model-View-Controller :

**“MVC” : ”Model-View-Controller” (Modele / Vue / Controleur en français donc). C’est un design pattern (patron de conception), c’est à dire un concept d’architecture logicielle pour son application. Il permet d’avoir un code plus structure, plus évolutif, plus maintenable, permettant de profiter de plusieurs mécanismes, d’avoir de la persistance de données, et bien d’autres choses encore.**

**Le “Modele” est la représentation interne des données. Il permet comme son nom l’indique de modéliser les données que l’on va manipuler dans l’application. Le modèle représente les véritables données avec toutes les informations qu’elles véhiculent.**

**La “Vue” quant à elle est la représentation visuelle de ces données à l’écran. Le contrôleur enfin, sert à faire l’interface entre le modèle et la vue. En effet, puisque le modèle et la vue sont sensés être au maximum indépendants, le contrôleur sert à faire le lien pour faire communiquer l’un (M) avec l’autre (V).**



### Entity Framework :

**Entity Framework (EF) est un mappeur objet-relationnel (ORM) qui permet aux développeurs .NET de travailler avec des données relationnelles à l'aide d'objets spécifiques à un domaine. Cela élimine le besoin de la plupart du code d'accès aux données que les développeurs doivent généralement écrire.**

**Entity Framework vous permet de créer un modèle en écrivant du code ou en utilisant des boîtes et des lignes dans EF Designer. Ces deux approches peuvent être utilisées pour cibler une base de données existante ou créer une nouvelle base de données.**

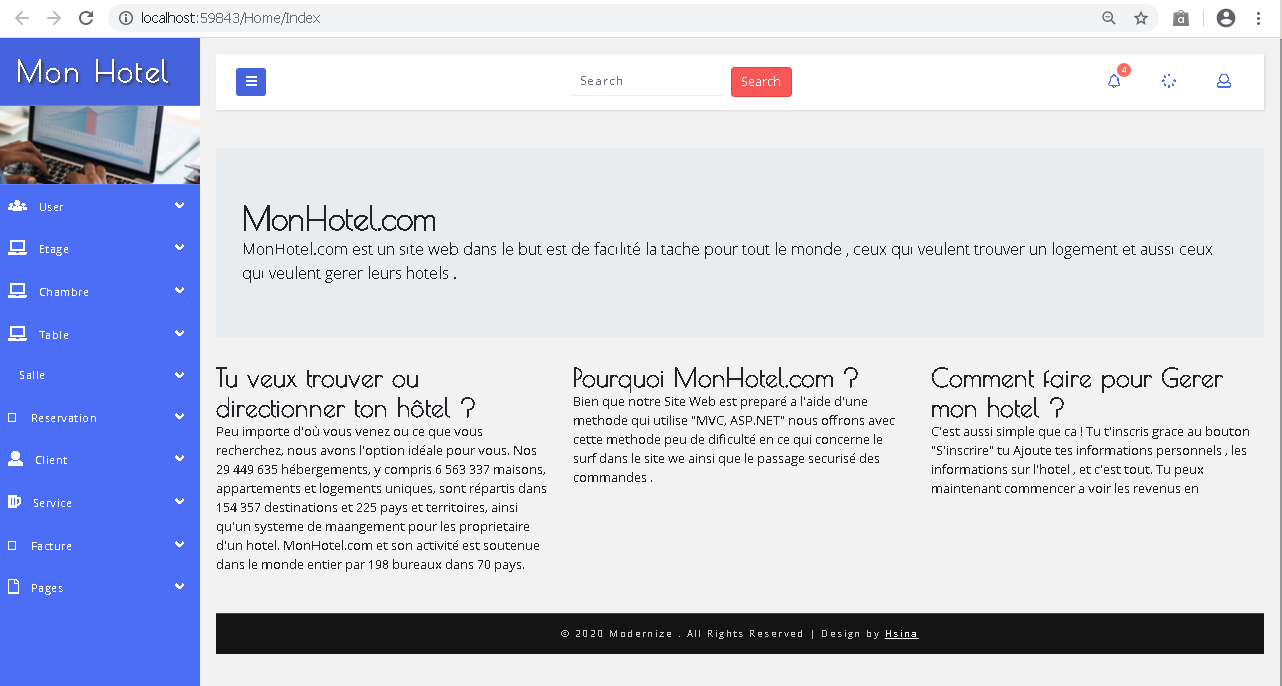
**Entity Framework est le principal ORM fourni par Microsoft pour .NET Framework et la technologie d'accès aux données recommandée par Microsoft.**

# Présentation

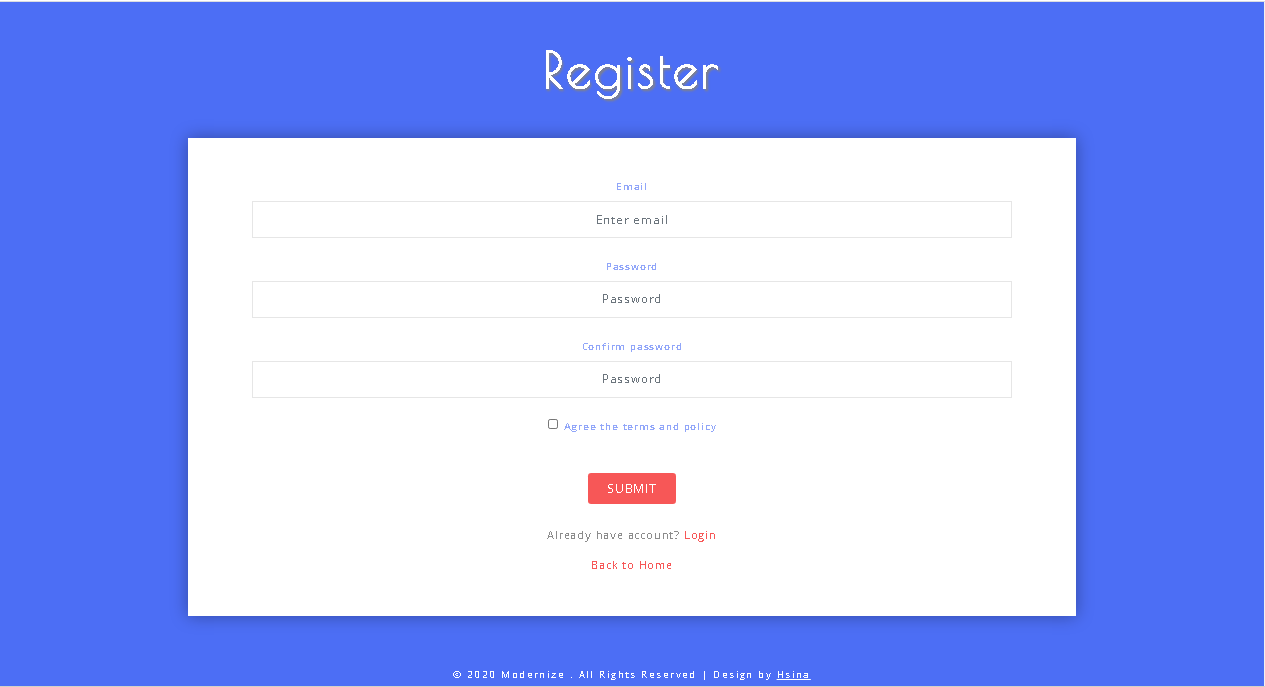
## Introduction

**Apres avoir affecté l’étude et la conception de notre application, nous passons à la phase d’implémentation. Ce chapitre présent le résultat du travail effectue durant ce projet de fin d’études. Nous allons présenter, aussi, quelques captures d’écran démontrant les fonctionnalités de notre application.**

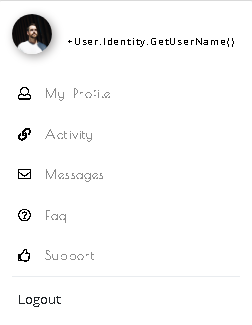
## Présentation des exemples des interfaces

**Ci-dessous, l’interface de la page d’accueil :**

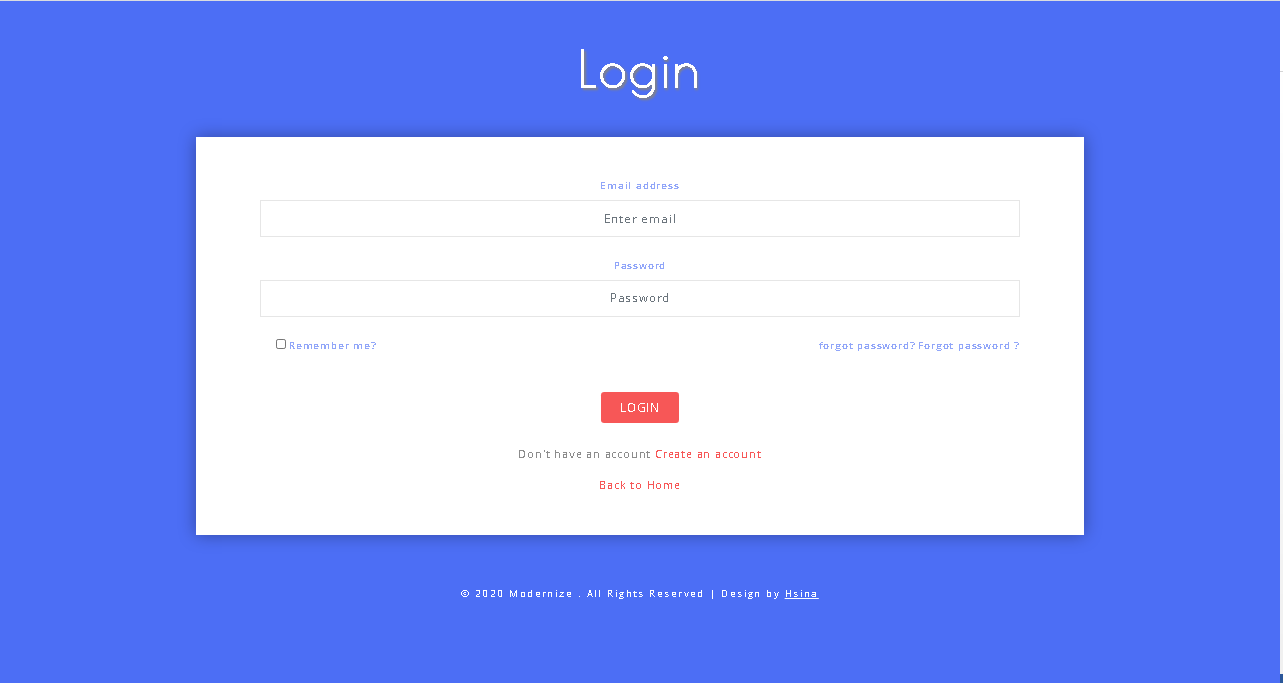
**Ci-dessous, l’interface de la page Inscription :**

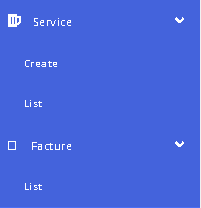
****

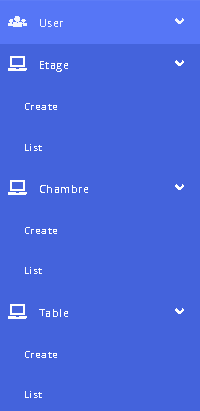
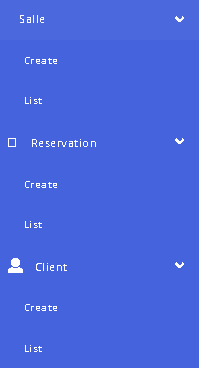
**Ci-dessous, l’interface de configuration de compte:**

****

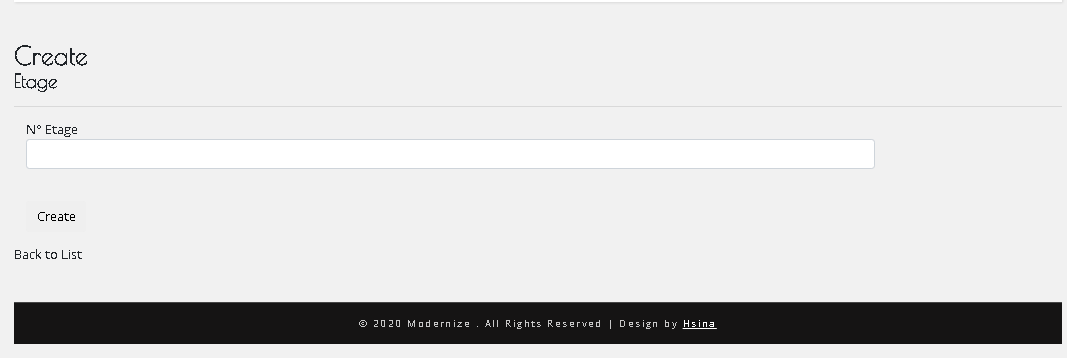
**Ci-dessous, l’interface d’authentification :**

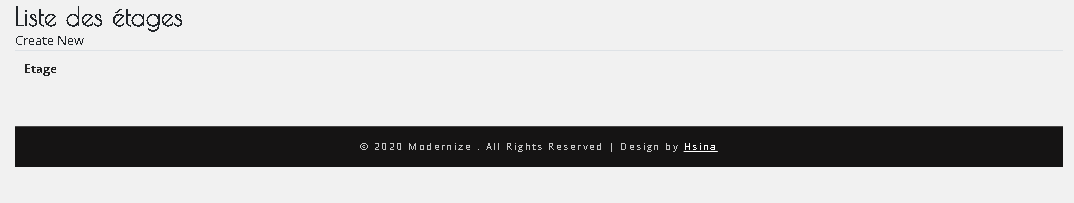
****

**Ci-dessous, l’interface des articles:**

****

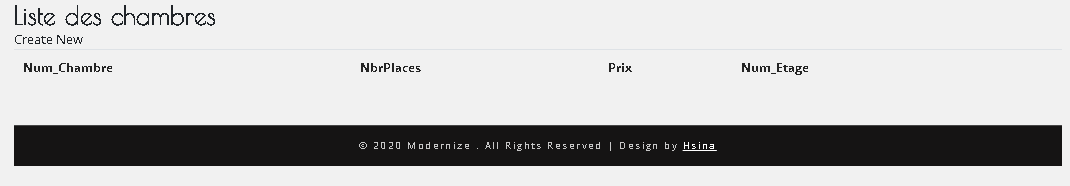
**Ci-dessous, l’interface de création d’étage et la liste des étages :**

****

****

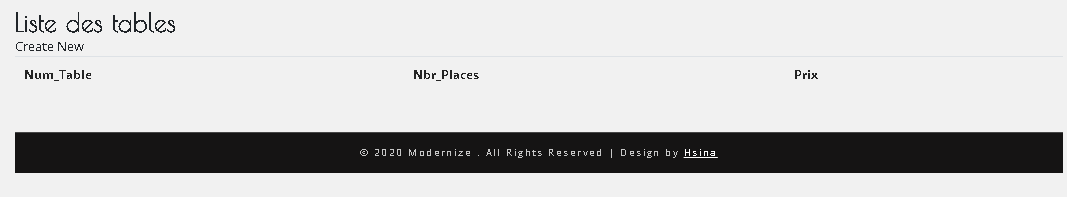
**Ci-dessous, l’interface des chambres et la liste des chambres:**

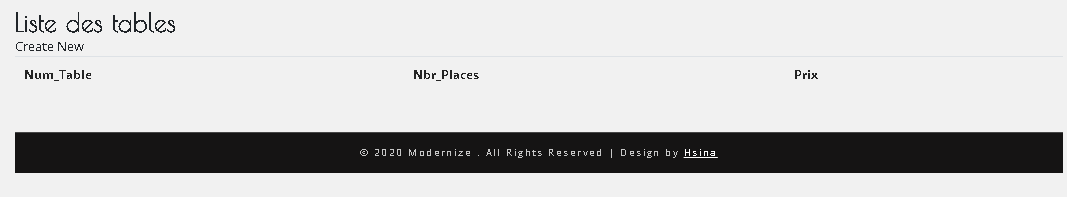
****

****

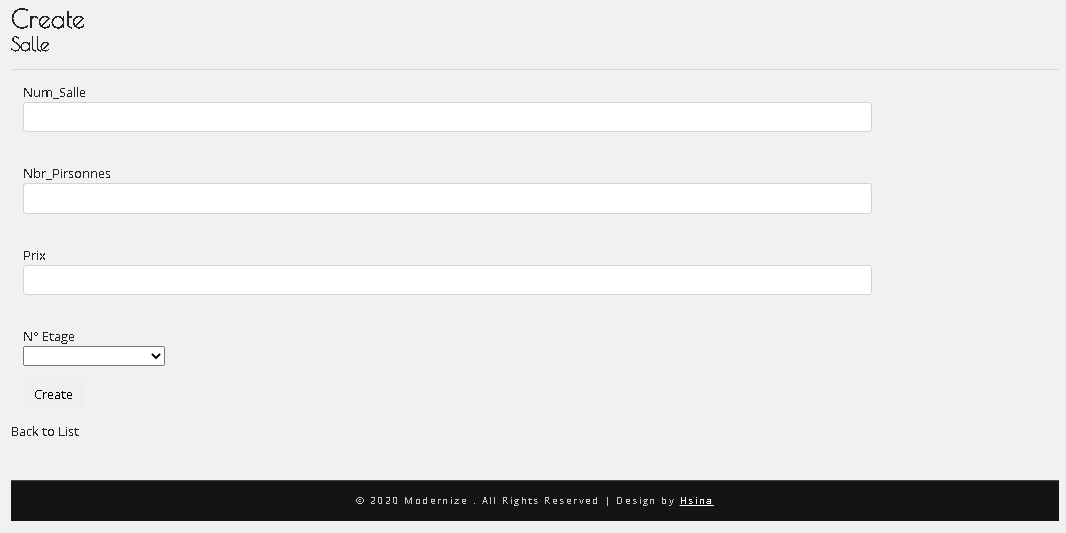
**Ci-dessous, l’interface de création de table et la liste des tables :**

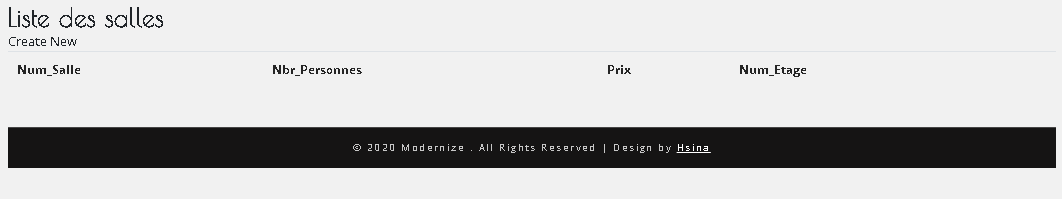
****

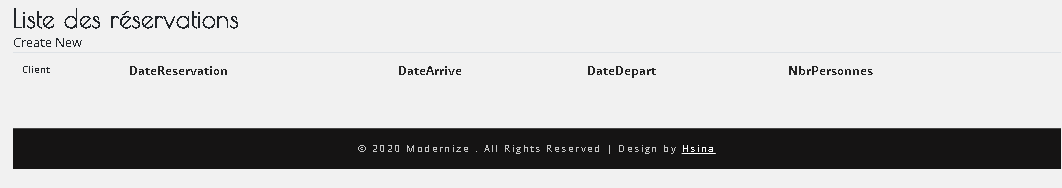
****

****

**Ci-dessous, l’interface de création des salles et liste de salle:**

****

****

**Ci-dessous, l’interface de la liste de réservation et de la création d’une réservation: **

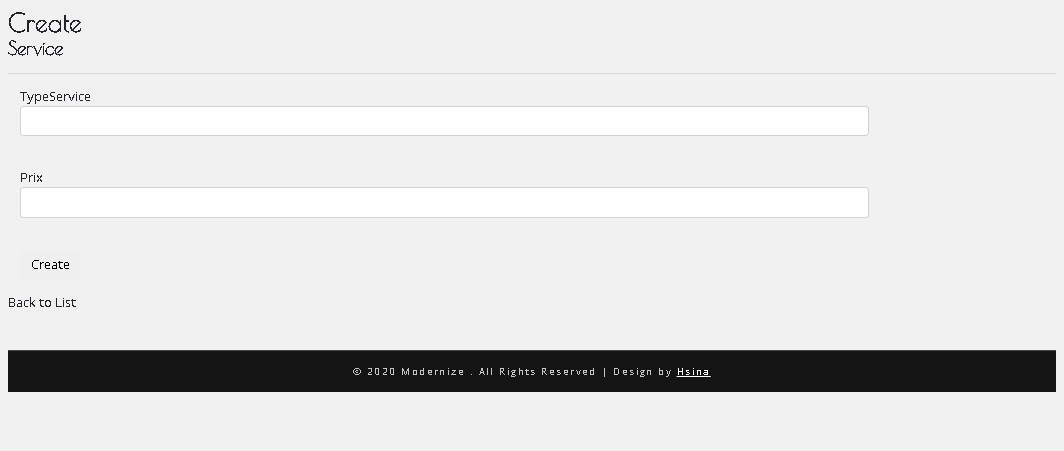


**Ci-dessous, l’interface de l’ajout d’un client et la liste des clients:**

****

****

**Ci-dessous, l’interface de la création d’un service et la liste des services:**

****

****

# Conclusion générale

**Ce travail était l’occasion d’appliquer dans un cadre professionnel les connaissances acquises durant notre formation à la NTIC. En effet, il mêlait ensemble plusieurs disciplines et m’a permis de mettre à profit les études des autres modules et de parfaire nos connaissances dans des modules déjà étudiées à la première année. Les acquis du cours de programmation objet étaient sans cesse sollicites et ce nouveau développement de projet en ASP.NET nous a encore permis d’aller plus loin dans les possibilités du langage et d’acquérir de nouvelles connaissances surtout en Framework ASP qui est le MVC.**