



République Tunisienne  
Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique  
Université de Tunis El Manar



## Rapport de projet

---

*Sujet : Développement d'une application de gestion  
d'hôtel*

---

*Réalisé par :*  
Seifeddine Touj

---

Année universitaire : 2023/2024

# Sommaire

<b>I. Introduction</b>	<b>3</b>
<b>II. Description du Projet</b>	<b>3</b>
A. Objectifs du projet	3
B. Technologies utilisées	3
C. Architecture de l'application	3
<b>III. Fonctionnalités Implémentées</b>	<b>4</b>
A. Recherche et Réservation de Chambres	4
B. Gestion des Réservations	6
C. Authentification et Gestion des Utilisateurs	6
D. Gestion des Chambres (pour les administrateurs)	7
<b>IV. Implémentation Technique</b>	<b>8</b>
A. Utilisation de Spring Boot	8
B. Modèle MVC avec Spring MVC	9
C. Intégration de Spring Data JPA	10
D. Sécurité avec Spring Security	11
E. Tests avec Postman	13
G. Frontend en Angular	14
<b>VIII. Conclusion</b>	<b>16</b>

## **I. Introduction**

Mon projet vise à répondre au besoin croissant d'un système de gestion de réservation de chambres d'hôtel en ligne. Avec la montée en puissance des réservations en ligne, il est devenu essentiel pour les hôtels de disposer d'une plateforme numérique permettant aux clients de rechercher, réserver et gérer leurs séjours de manière efficace et sécurisée.

Mon système permettra aux utilisateurs de rechercher des chambres d'hôtel disponibles, de consulter les détails et les photos des chambres, et de réserver en ligne. Les administrateurs auront accès à des fonctionnalités supplémentaires telles que la gestion des réservations et des chambres.

## **II. Description du Projet**

### **A. Objectifs du projet**

Le principal objectif de mon projet est de fournir une plateforme complète de gestion de réservation de chambres d'hôtel, offrant une expérience utilisateur fluide et sécurisée. Je vise à simplifier le processus de réservation et à offrir aux utilisateurs un accès facile à toutes les informations nécessaires.

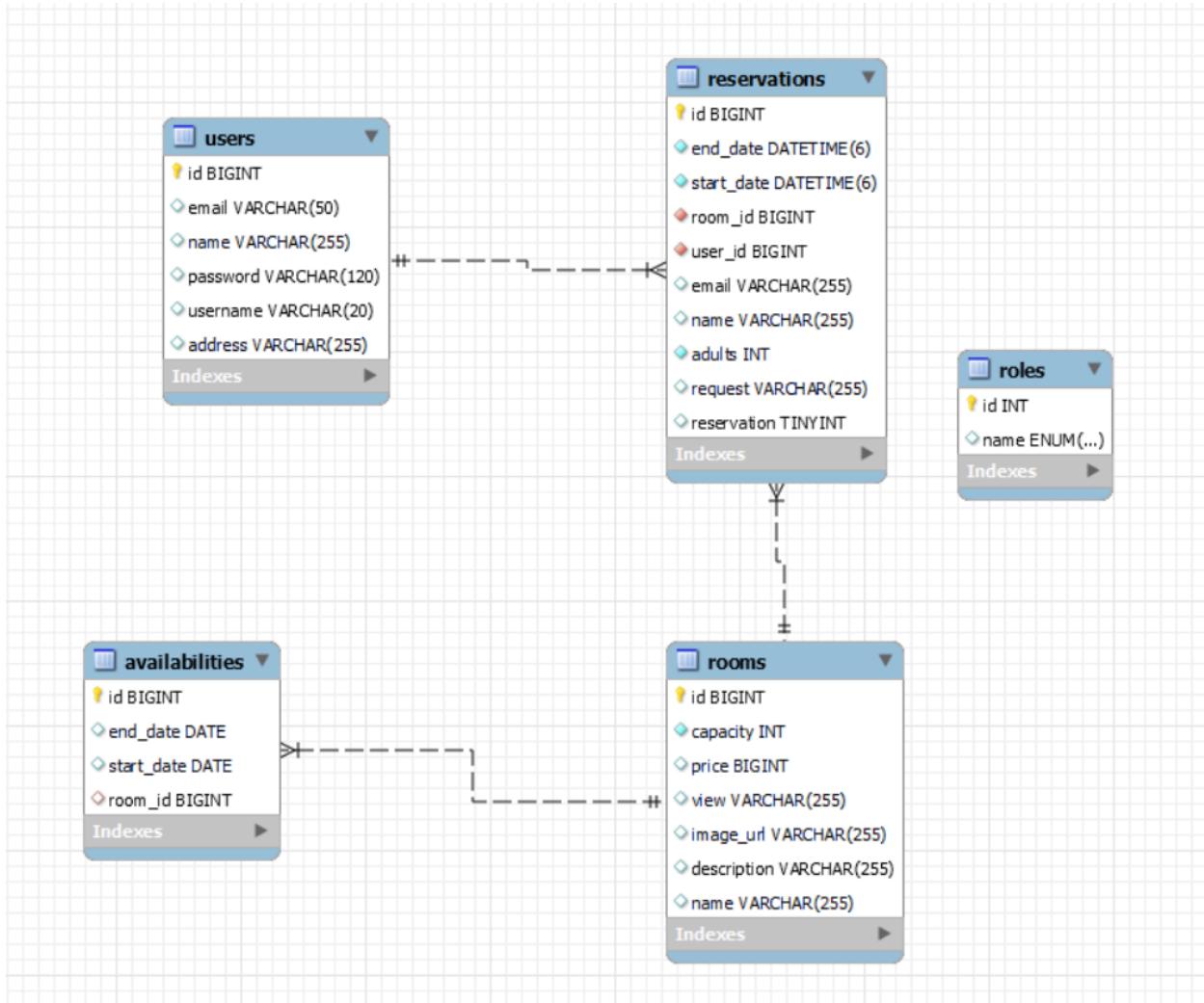
### **B. Technologies utilisées**

Pour atteindre mes objectifs, j'ai choisi d'utiliser le framework Spring pour le développement de mon application. Spring offre une gamme complète d'outils et de fonctionnalités qui simplifient le développement d'applications Java, notamment Spring Boot, Spring MVC, Spring Data JPA et Spring Security. Je vais également utiliser Postman pour tester mes API, et Angular pour le développement du frontend.

### **C. Architecture de l'application**

Mon application suivra une architecture MVC (Modèle-Vue-Contrôleur), avec Spring MVC pour gérer les interactions utilisateur. Spring Data JPA sera utilisé pour interagir avec la base de données, tandis que Spring Security assurera la sécurité de la

plateforme. L'application sera développée de manière modulaire et extensible, permettant une évolutivité facile à l'avenir.



### III. Fonctionnalités Implémentées

#### A. Recherche et Réservation de Chambres

Les utilisateurs pourront rechercher des chambres d'hôtel disponibles en spécifiant des critères tels que la date d'arrivée, la date de départ. Ils pourront ensuite visualiser les détails des chambres et réserver en ligne en fournissant les informations nécessaires.

**HOTELIER**

HOME ABOUT PROFILE

/

/
SUBMIT

OUR ROOMS

## Explore Our ROOMS



Serene Haven Suite test

2 Adults | sea view

Embrace tranquility in our Serene Haven Suite, where luxury meets serenity. With breathtaking views and elegant amenities, unwind in style amidst peaceful surroundings.

[VIEW DETAIL](#)

[BOOK NOW](#)



Royal Oasis Retreat:

5 Adults | Garden view

Indulge in opulence at our Royal Oasis Retreat, where every detail exudes regal sophistication. Experience unparalleled comfort and impeccable service in a setting fit for royalty.

[VIEW DETAIL](#)

[BOOK NOW](#)



Sunset Horizon Hideaway

5 Adults | Houses view

Surrender to the beauty of nature from our Sunset Horizon Hideaway, where each evening paints the sky with vibrant hues. Immerse yourself in serenity and let the world fade away.

[VIEW DETAIL](#)

[BOOK NOW](#)



Azure Skyline Suite test

4 Adults | sea view

Elevate your stay with panoramic views from our Azure Skyline Suite, where the skyline meets the horizon in a seamless blend of urban charm and natural beauty. Experience the city from a new perspective.

[VIEW DETAIL](#)

[BOOK NOW](#)



Test Room

6 Adults | sea view

aaaaaaaaaaaaaaaaaaaaaa

[VIEW DETAIL](#)



TEST 2

7 Adults | Houses view

hhhhhhhhhhhhhh

[VIEW DETAIL](#)

[BOOK NOW](#)

ROOM BOOKING

## Book A LUXURY ROOM







CALENDAR

CALENDAR

Select Adult

Special Request

BOOK NOW

### B. Gestion des Réservations

Les utilisateurs auront accès à un historique de leurs réservations et pourront gérer les détails tels que les dates et les chambres. Les administrateurs auront un accès complet à toutes les réservations effectuées et pourront gérer leur statut.

HOME
ABOUT
PROFILE

Checkin	Checkout	Price	Status
4/13/24, 2:00 AM	4/20/24, 2:00 AM	50	<a href="#" style="color: blue; border: 1px solid #0072BD; padding: 2px 5px;">Edit</a> <a href="#" style="color: red; border: 1px solid red; padding: 2px 5px;">Delete</a>
4/15/24, 2:00 AM	4/12/24, 2:00 AM	70	<a href="#" style="color: blue; border: 1px solid #0072BD; padding: 2px 5px;">Edit</a> <a href="#" style="color: red; border: 1px solid red; padding: 2px 5px;">Delete</a>

### C. Authentification et Gestion des Utilisateurs

Les utilisateurs pourront créer un compte, se connecter et gérer leurs informations personnelles. Les administrateurs auront également accès à un tableau de bord d'administration où ils pourront gérer les comptes utilisateurs et les rôles associés.

## WELCOME admin

Checkin	Checkout	Price	Status
4/13/24, 2:00 AM	4/20/24, 2:00 AM	50	
4/15/24, 2:00 AM	4/12/24, 2:00 AM	70	

Address  
19,lzlzlz

UPDATE

ID	Name	Email	Actions
1	admin	admin@c.bbb	<button style="background-color: blue; color: white; border: none; padding: 2px 5px;">Edit</button> <button style="background-color: red; color: white; border: none; padding: 2px 5px;">Delete</button>

### Update User

Address  
19,lzlzlz

ROLE\_USER

Update

### D. Gestion des Chambres (pour les administrateurs)

Les administrateurs auront la possibilité d'ajouter de nouvelles chambres, de modifier les détails des chambres existantes et de supprimer des chambres

obsolesques. Ils auront également la possibilité de gérer les disponibilités des chambres en mettant à jour les dates de disponibilité.

#	name	capacity	view	price	Edit	Delete
2	Serene Haven Suite test	2	sea	60	<button>Edit</button>	<button>Delete</button>
103	Royal Oasis Retreat:	5	Garden	70	<button>Edit</button>	<button>Delete</button>
104	Sunset Horizon Hideaway	5	Houses	70	<button>Edit</button>	<button>Delete</button>
152	Azure Skyline Suite test	4	sea	50	<button>Edit</button>	<button>Delete</button>
352	Test Room	6	sea	59	<button>Edit</button>	<button>Delete</button>
353	TEST 2	7	Houses	101	<button>Edit</button>	<button>Delete</button>

### Room Edit/Create

Room Name  
Royal Oasis Retreat:

Room View  
Garden

Room Capacity  
5

Room Price  
70

Room Description  
Indulge in opulence at our Royal Oasis Retreat, where every detail exudes regal sophistication. Experience unparalleled comfort and impeccable service in a setting fit for royalty.

[Book Now](#)

## IV. Implémentation Technique

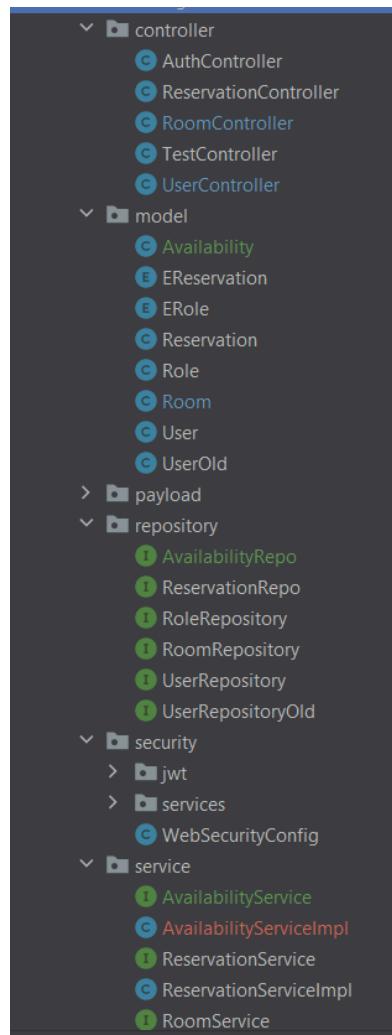
### A. Utilisation de Spring Boot

L'utilisation de Spring Boot dans mon projet m'a permis de simplifier considérablement la configuration et le déploiement de l'application. Grâce à ses fonctionnalités de configuration automatique et de gestion des dépendances, j'ai pu

créer une application autonome prête à être exécutée avec un minimum d'effort. Spring Boot offre également une intégration transparente avec d'autres modules Spring, ce qui facilite le développement et la maintenance de l'application.

## B. Modèle MVC avec Spring MVC

J'ai adopté le modèle MVC (Modèle-Vue-Contrôleur) pour organiser mon application, avec Spring MVC comme framework de contrôle. Cette approche m'a permis de séparer clairement les préoccupations liées à la logique métier, à la présentation des données et au traitement des requêtes utilisateur. Les contrôleurs Spring MVC ont été utilisés pour gérer les différentes interactions utilisateur, en orchestrant les actions appropriées et en renvoyant les vues nécessaires.



## C. Intégration de Spring Data JPA

L'intégration de Spring Data JPA a été un élément clé de mon projet pour interagir avec la base de données relationnelle. Voici comment j'ai mis en place cette intégration :

### Configuration de la source de données :

Tout d'abord, j'ai configuré la source de données dans mon application Spring, en spécifiant les détails de connexion à ma base de données relationnelle. Cela peut inclure des paramètres tels que l'URL JDBC, le nom d'utilisateur et le mot de passe.

### Définition des entités :

Ensuite, j'ai défini des entités Java qui représentent les tables de ma base de données. Chaque entité correspond à une table et chaque champ de l'entité correspond à une colonne de la table. J'ai utilisé des annotations JPA telles que @Entity, @Table, @Id, @Column, etc., pour définir les entités et leurs propriétés.

### Création de référentiels :

J'ai créé des interfaces de référentiel (repository interfaces) pour chaque entité, en étendant l'interface JpaRepository fournie par Spring Data JPA. Ces interfaces de référentiel m'ont permis de définir des méthodes de requête personnalisées pour accéder aux données de manière efficace.

### Utilisation des requêtes dérivées :

J'ai utilisé les requêtes dérivées de Spring Data JPA pour définir des requêtes de base de données en utilisant la convention de nommage des méthodes. Par exemple, en déclarant une méthode avec un nom tel que findByNom, Spring Data JPA génère automatiquement une requête SQL pour rechercher des enregistrements en fonction du champ "nom".

### Définition de requêtes JPQL :

Pour des requêtes plus complexes, j'ai également utilisé des requêtes JPQL (Java Persistence Query Language). Ces requêtes sont définies en utilisant des annotations @Query dans mes interfaces de référentiel, et elles offrent une flexibilité supplémentaire pour interroger la base de données.

```
👤 Seifeddine Touj
@Entity
@Table(name = "users",
       uniqueConstraints = {
           @UniqueConstraint(columnNames = "username"),
           @UniqueConstraint(columnNames = "email")
       })
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    3 usages
    @NotBlank
    @Size(max = 20)
    private String username;
```

```
3 usages 👤 Seifeddine Touj
public interface RoomRepository extends JpaRepository<Room, Long> {
    1 usage 👤 Seifeddine Touj
    @Query("SELECT r FROM Room r WHERE r NOT IN (SELECT res.room FROM Reservation res WHERE res.startDate <= :endDate AND res.endDate >= :startDate)")
    List<Room> findAvailableRooms(Date startDate, Date endDate);
}
```

```
3 usages 👤 Seifeddine Touj
public interface ReservationRepo extends JpaRepository<Reservation, Long> {
    1 usage 👤 Seifeddine Touj
    List<Reservation> findByUser(Optional<User> user);
}
```

## D. Sécurité avec Spring Security

L'intégration de Spring Security a été une étape cruciale pour assurer la sécurité de mon application. Voici comment j'ai mis en place la sécurité avec Spring Security :

### Configuration initiale :

J'ai ajouté la dépendance Spring Security à mon projet, ce qui m'a permis d'accéder à toutes les fonctionnalités de sécurité offertes par Spring. Ensuite, j'ai configuré

Spring Security en définissant un fichier de configuration de sécurité où j'ai spécifié les règles de sécurité pour mon application.

#### Gestion de l'authentification :

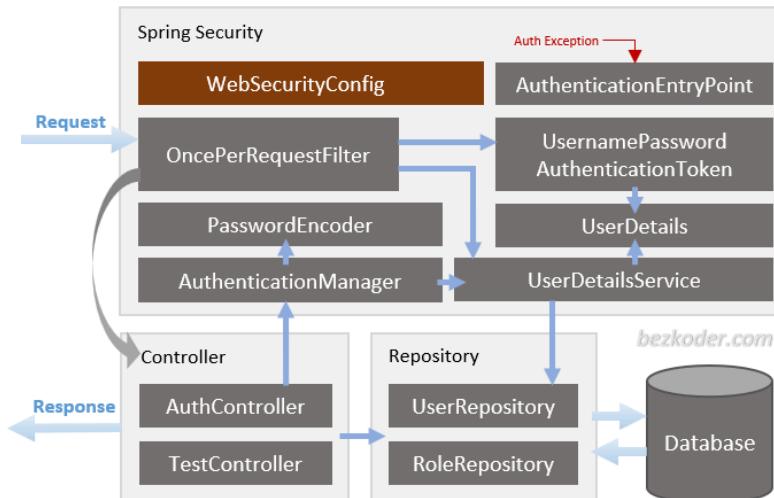
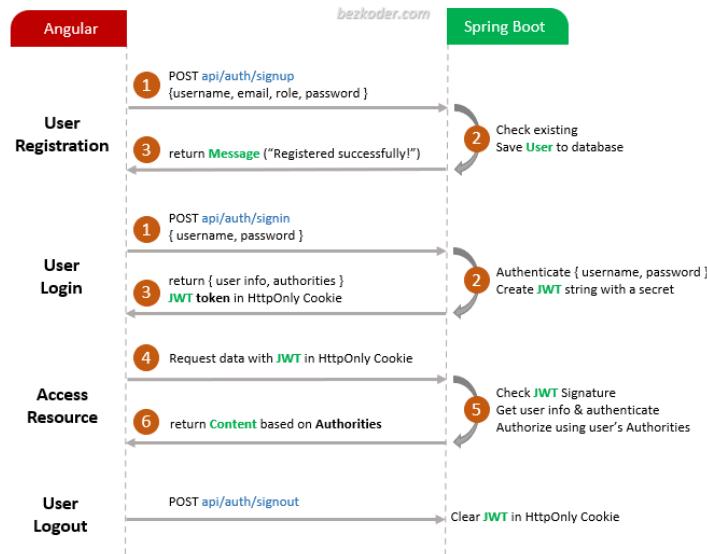
J'ai configuré Spring Security pour gérer l'authentification des utilisateurs. Cela inclut la définition des sources d'authentification, la configuration des filtres d'authentification, et la définition du mécanisme d'authentification avec les tokens JWT.

#### Gestion des autorisations :

J'ai défini des règles d'autorisation pour contrôler l'accès aux différentes fonctionnalités de mon application. Cela inclut l'attribution de rôles aux utilisateurs et la configuration des autorisations associées à chaque rôle. Par exemple, j'ai spécifié que seuls les utilisateurs avec le rôle d'administrateur peuvent accéder aux fonctionnalités d'administration de l'application.

#### Tests de sécurité :

J'ai effectué des tests de sécurité pour vérifier l'efficacité de ma configuration de sécurité. Cela inclut des tests d'authentification pour m'assurer que seuls les utilisateurs authentifiés peuvent accéder aux fonctionnalités protégées, des tests d'autorisation pour vérifier que les autorisations sont correctement appliquées, et des tests de vulnérabilités pour identifier et corriger les éventuelles failles de sécurité.



## E. Tests avec Postman

Les tests avec Postman ont été essentiels pour garantir le bon fonctionnement de mes API développées. Voici comment j'ai utilisé Postman pour tester mon application

### Création de collections d'API :

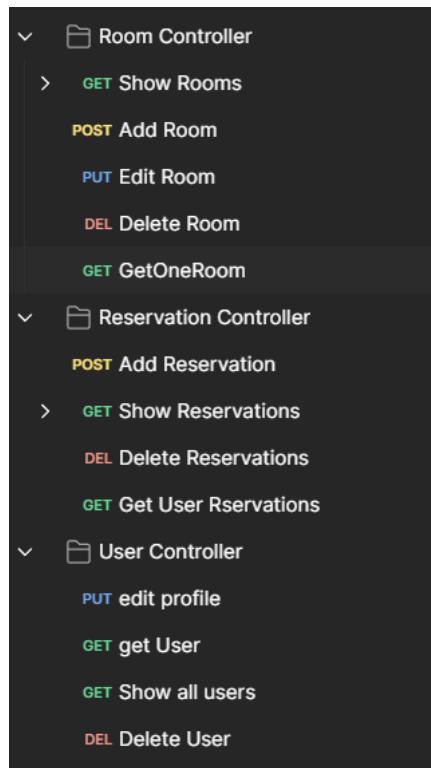
J'ai regroupé les différentes requêtes API liées à chaque fonctionnalité de mon application dans des collections distinctes. Par exemple, j'ai créé une collection pour la recherche de chambres, une autre pour la gestion des réservations, etc.

### Configuration des requêtes :

Pour chaque endpoint API, j'ai configuré les requêtes HTTP appropriées dans Postman. Cela inclut la spécification de la méthode HTTP (GET, POST, PUT, DELETE), l'URL de l'endpoint, les paramètres de requête, les en-têtes HTTP nécessaires (comme les jetons d'authentification), et éventuellement le corps de la requête pour les méthodes POST et PUT.

### Validation des réponses :

J'ai utilisé les fonctionnalités de validation de réponse de Postman pour vérifier que les réponses renvoyées par mes API étaient correctes. Cela inclut la vérification des codes de statut HTTP, des données renvoyées dans le corps de la réponse, et des en-têtes HTTP si nécessaire.



## G. Frontend en Angular

Pour le développement du frontend de mon application, j'ai choisi d'utiliser Angular. Angular est un framework JavaScript moderne développé et maintenu par Google, qui offre une structure robuste pour la création d'applications web évolutives et performantes. Voici comment j'ai intégré Angular dans mon projet :

### **Configuration de l'environnement de développement :**

J'ai configuré l'environnement de développement en installant Node.js et npm (Node Package Manager), nécessaires pour gérer les dépendances et exécuter des commandes Angular CLI (Command Line Interface).

### **Création du projet Angular :**

En utilisant Angular CLI, j'ai créé un nouveau projet Angular avec la commande `ng new nom-du-projet`. Cela a généré une structure de dossier initiale avec les fichiers et répertoires nécessaires pour démarrer le développement.

### **Développement des composants :**

J'ai développé des composants Angular pour chaque partie de l'interface utilisateur, tels que la page de recherche de chambres, la page de détails de la chambre, le formulaire de réservation, etc. Chaque composant a son propre fichier TypeScript, fichier HTML et fichier de style CSS.

### **Gestion de l'état de l'application :**

J'ai utilisé les services et les observables d'Angular pour gérer l'état de l'application de manière efficace. Les services ont été utilisés pour encapsuler la logique métier et partager des données entre les composants, tandis que les observables ont été utilisés pour écouter les changements de données asynchrones.

### **Routage :**

J'ai configuré le routage dans mon application Angular en définissant des routes pour chaque vue ou page de l'application. Cela m'a permis de naviguer entre les différentes parties de l'application en fonction de l'URL.

### **Communication avec l'API backend :**

J'ai utilisé le module HttpClient d'Angular pour effectuer des requêtes HTTP vers mon API backend. J'ai développé des services Angular pour encapsuler les appels API et gérer les réponses de manière cohérente dans toute l'application.

#### Gestion des formulaires :

Angular offre des fonctionnalités puissantes pour la gestion des formulaires, y compris la validation côté client. J'ai utilisé les modules FormsModule pour créer et valider les formulaires de réservation, de connexion, etc.

## **VIII. Conclusion**

En conclusion, mon projet de développement d'une plateforme de gestion de réservation de chambres d'hôtel en ligne a été une expérience enrichissante et gratifiante. J'ai réussi à créer une application robuste et fonctionnelle, offrant aux utilisateurs une expérience fluide et sécurisée lors de la recherche, de la réservation et de la gestion des chambres d'hôtel.

En utilisant les fonctionnalités avancées du framework Spring, notamment Spring Boot, Spring MVC, Spring Data JPA et Spring Security, j'ai pu développer rapidement et efficacement mon application tout en garantissant sa fiabilité et sa sécurité. L'intégration de Postman pour les tests.

J'ai également offert la possibilité de développer le frontend de mon application en utilisant des technologies modernes telles Angular, ce qui permet une expérience utilisateur avancée et réactive.

En envisageant l'avenir, je vois des possibilités d'amélioration continue, notamment en optimisant les performances, en ajoutant de nouvelles fonctionnalités, en améliorant l'expérience utilisateur et en étendant la portée de mon application à un public plus large.