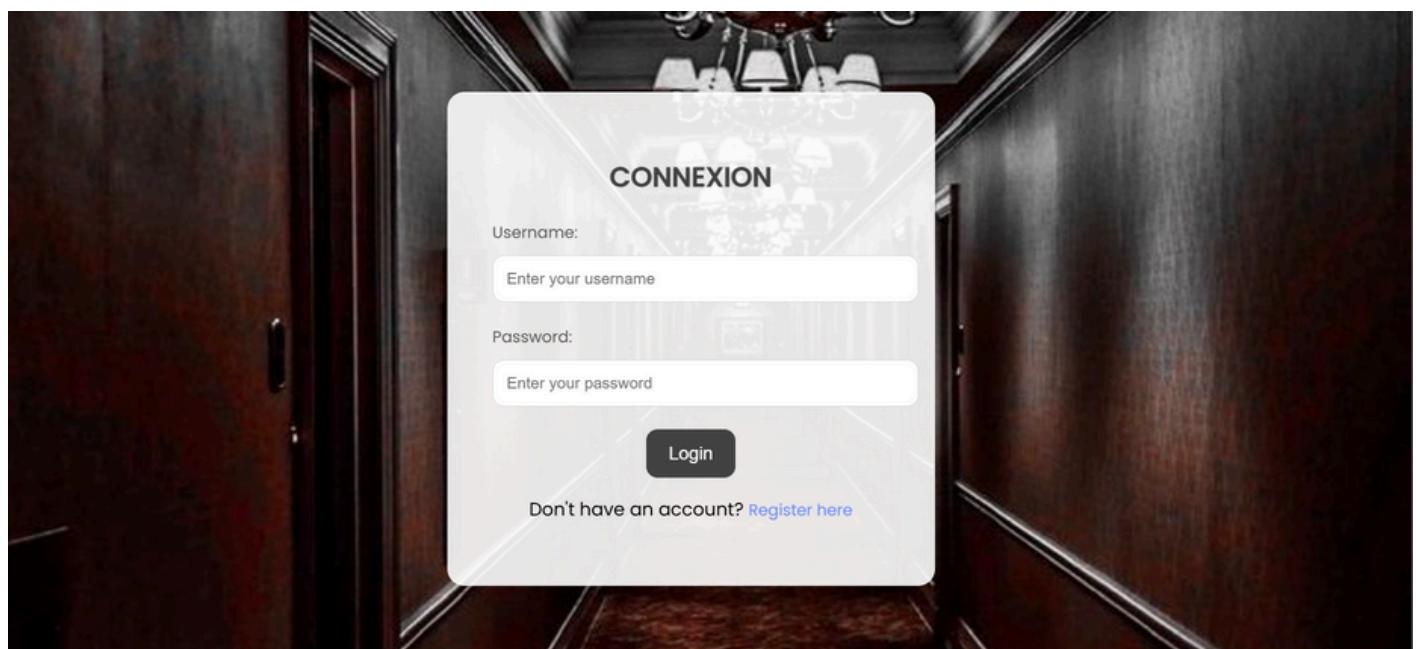


# Rapport du Project :

# “Gestion d'un Système de Réservation de Salles”



**Réalisé par: Manal Haggoud  
Sous l'encadrement de Monsieur Ahmed  
Amamou.**

### Objectif :

Créer une application web pour la gestion des réservations de salles dans un centre d'affaires.

Technologies utilisées:

- Hibernate pour la gestion des données.
- Servlets pour la logique métier.
- JSP pour l'interface utilisateur.

Utilisateurs :

- Admin :

Gère les salles et les utilisateurs.

- Client :

Réserve des salles et consulte les disponibilités.

- Authentification :

Connexion et déconnexion pour les administrateurs et les clients.

- Gestion des salles (Admin) :

Ajouter, modifier, et supprimer des salles.

Consulter la liste des salles disponibles.

- Réservation des salles (Client) :

Rechercher des salles selon des critères (date, capacité, équipements).

Réserver une salle disponible.

Annuler une réservation.

Afficher l'historique des réservations.

- Tableau de bord (Admin) :

Consulter des statistiques telles que :

Nombre de réservations.

Salles les plus utilisées.

- Base de données :

Créer une base de données relationnelle pour stocker les informations sur :

- Les utilisateurs.
- Les salles.
- Les réservations.
- Utiliser Hibernate pour le mapping objet-relationnel (ORM).

- Back-End :

Utiliser des Servlets pour gérer la logique métier.

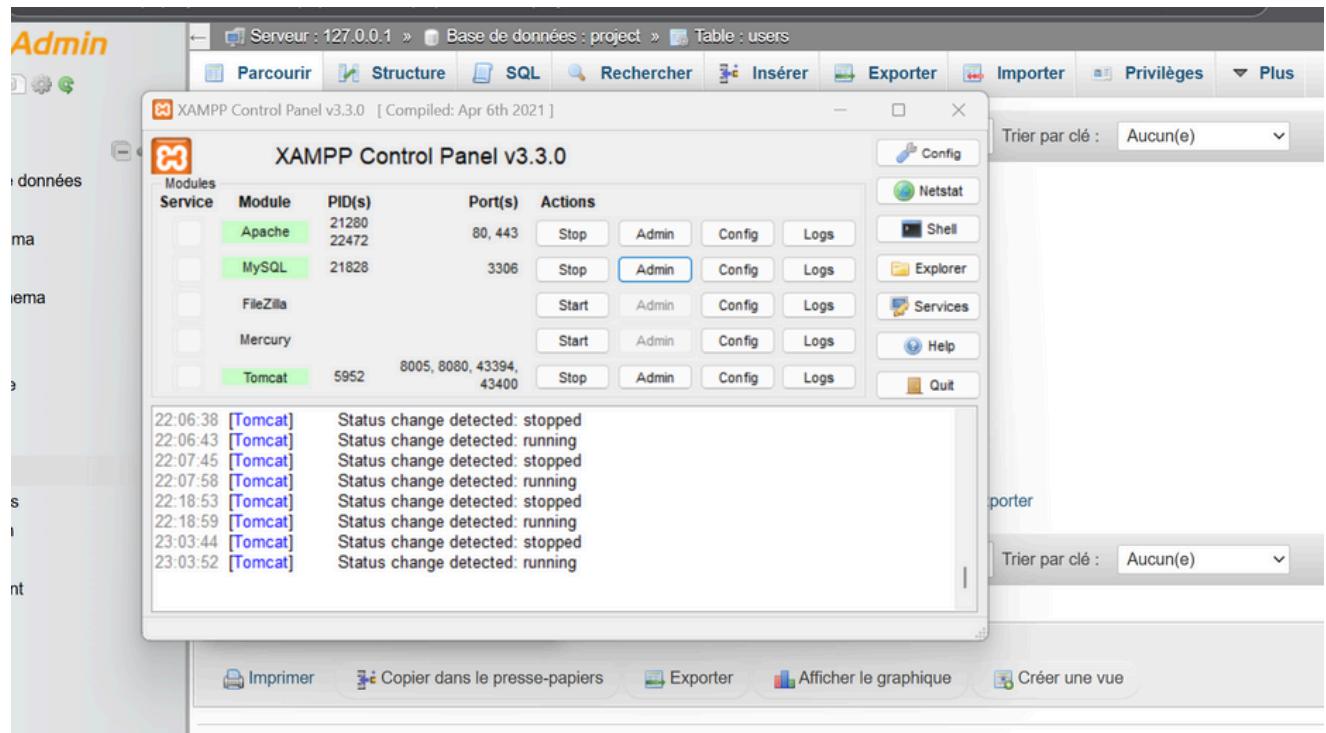
Implémenter un modèle MVC (Modèle-Vue-Contrôleur).

- Front-End :

Utiliser JSP pour les vues et les formulaires.

Ajouter un minimum de CSS pour rendre l'interface agréable.

# base de donee:



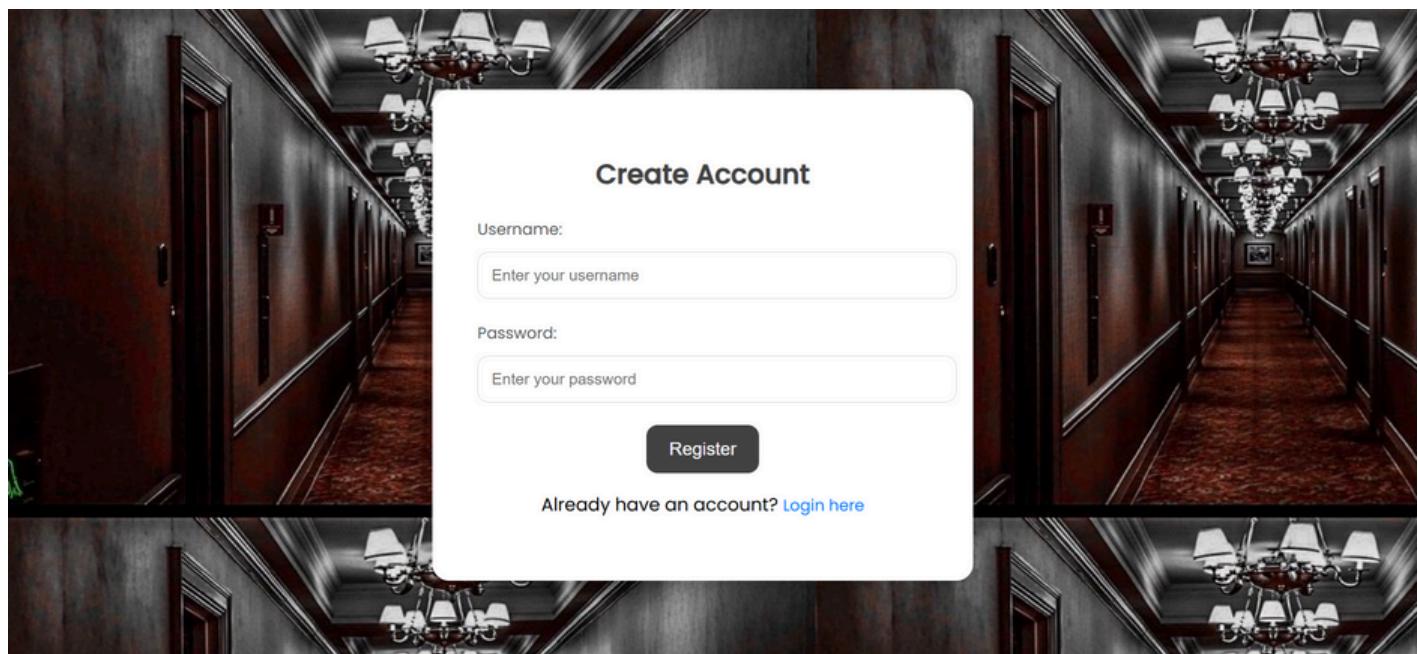
The screenshot shows the phpMyAdmin interface for the 'project' database. The left sidebar lists various databases and tables. The 'users' table is selected, displaying 10 rows of data. The columns are id, username, password, and role. The data includes users like admin, user1, manal, hala, aya, and rihab, each assigned specific roles (ADMIN, USER, user).

	id	username	password	role
1	1	admin	admin123	ADMIN
2	2	user1	password1	USER
3	3	manal	1234	user
5	5	hala	123	user
7	7	aya	123	user
10	10	rihab	123	user

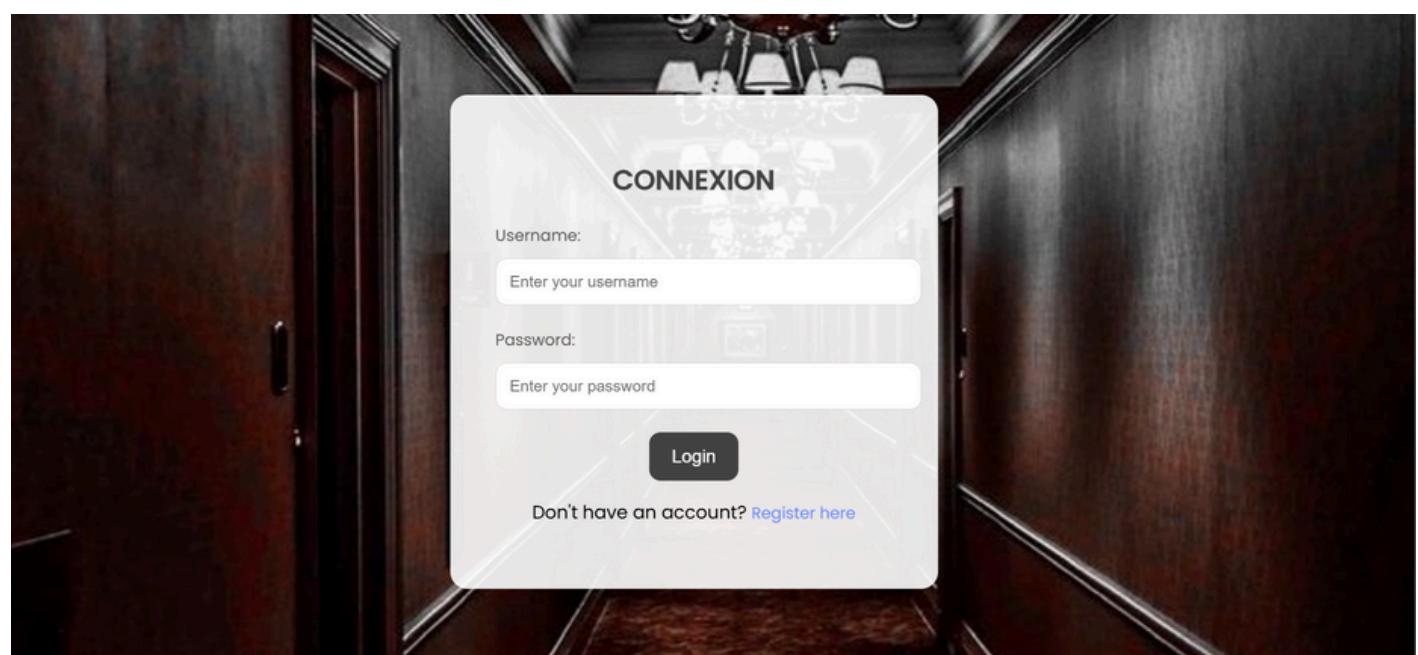
```
<class>model.Room</class>
<class>model.Reservation</class>
<properties>
    <property name="javax.persistence.jdbc.driver" value="com.mysql.cj.jdbc.Driver"/>
    <property name="javax.persistence.jdbc.url" value="jdbc:mysql://localhost:3306/project"/>
    <property name="javax.persistence.jdbc.user" value="root"/>
    <property name="javax.persistence.jdbc.password" value="" />

    <property name="hibernate.dialect" value="org.hibernate.dialect.MySQL8Dialect"/>
    <property name="hibernate.hbm2ddl.auto" value="update"/>
```

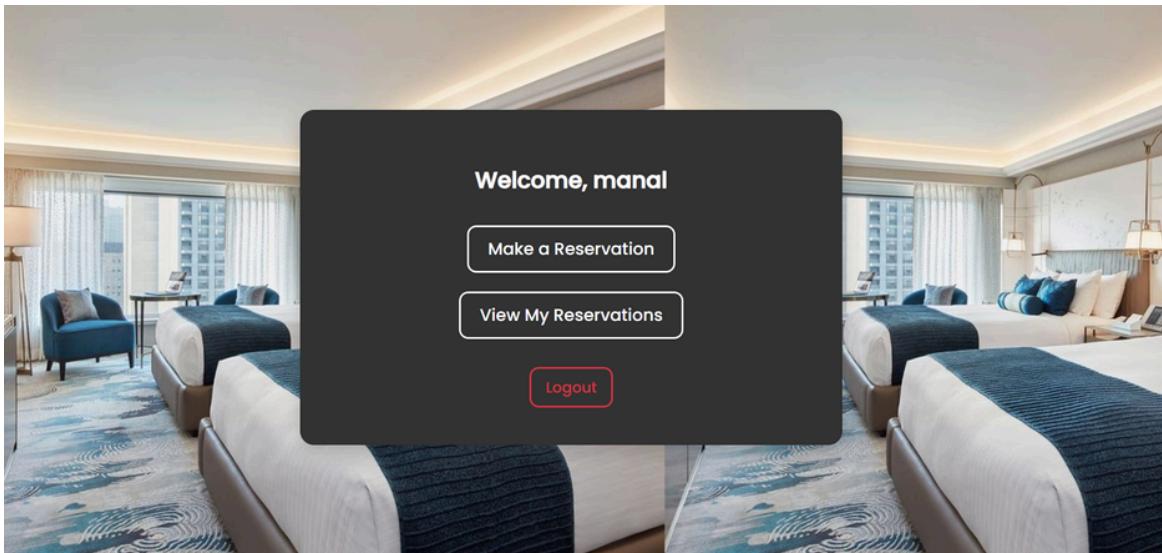
## Interface d'inscription:



## Interface de connexion:



# Interfaces pour l'utilisateur:



**Make a Reservation**

**Available Rooms**

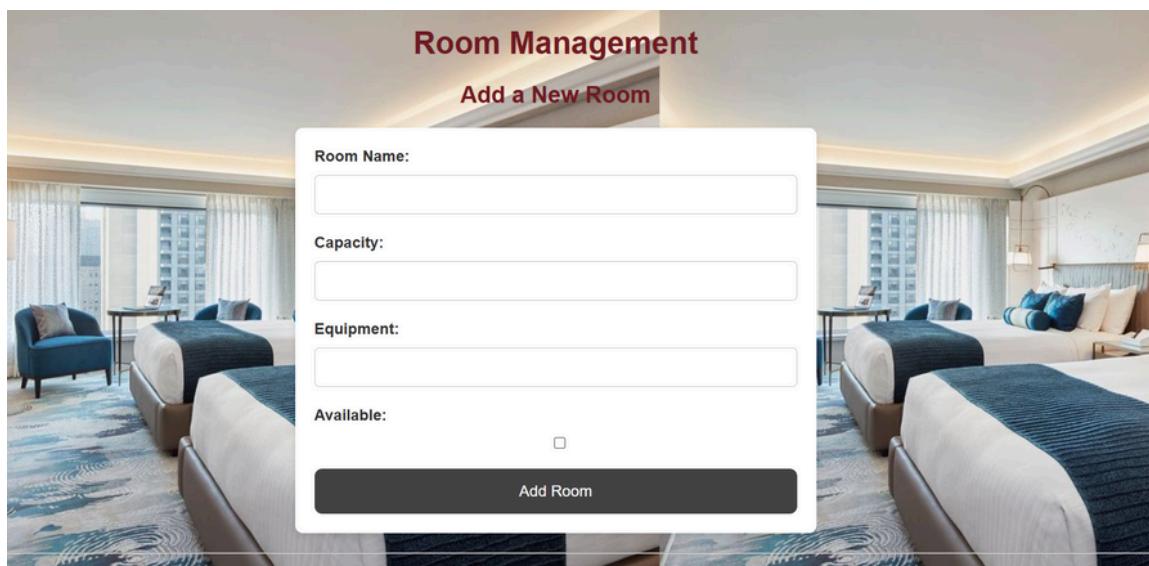
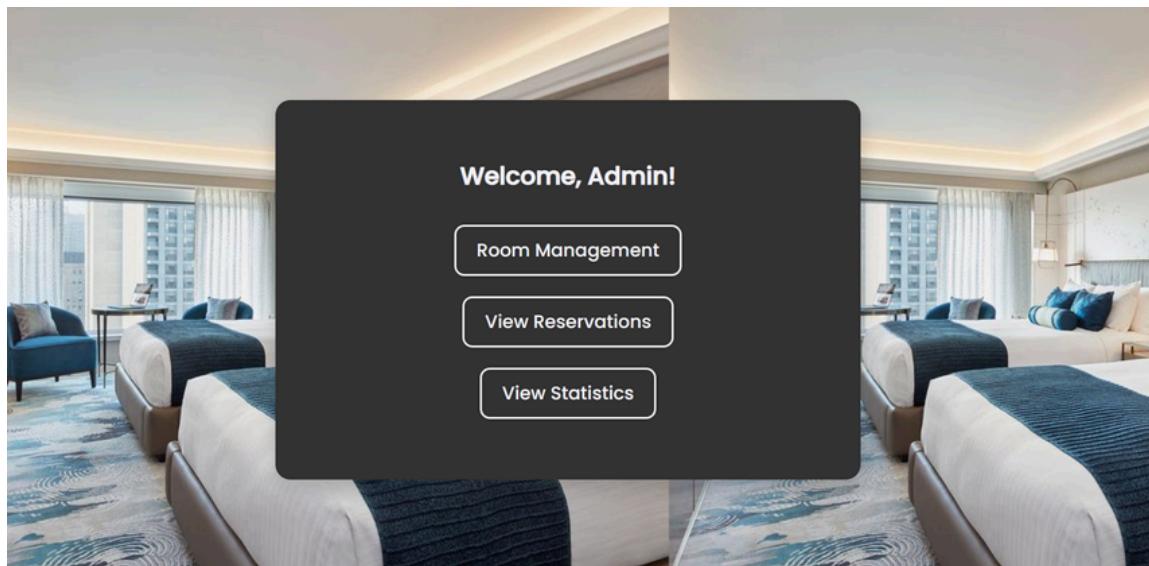
ID	Name	Capacity	Equipment	Availability	Actions
1	salle vip5(famille)	5	luxe	Available	Date: <input type="text" value="jj / mm / aaaa"/> <input type="button" value="..."/> Start Time: <input type="text" value="-- : --"/> <input type="button" value="..."/> End Time: <input type="text" value="-- : --"/> <input type="button" value="..."/> <input type="button" value="Reserve"/>
2	salle vip4	4	luxe	Available	Date: <input type="text" value="jj / mm / aaaa"/> <input type="button" value="..."/> Start Time: <input type="text" value="-- : --"/> <input type="button" value="..."/> End Time: <input type="text" value="-- : --"/> <input type="button" value="..."/> <input type="button" value="Reserve"/>

**My Reservations**

Room	Date	Start Time	End Time	Status
9	2024-12-30	02:26	21:26	Pending

[Back to Dashboard](#)

# Interfaces pour l'administrateur:



ID	NAME	CAPACITY	EQUIPMENT	AVAILABILITY	ACTIONS	
1	salle vip5(famille)	5	luxe	Available	<button>Delete</button>	<button>Edit</button>
2	salle vip4	4	luxe	Available	<button>Delete</button>	<button>Edit</button>
4	salle vip3	4	luxe	Available	<button>Delete</button>	<button>Edit</button>
5	salle vip1	2	luxe	Available	<button>Delete</button>	<button>Edit</button>
6	salle vip2	2	luxe	Available	<button>Delete</button>	<button>Edit</button>
7	chambre 1	2	standard	Available	<button>Delete</button>	<button>Edit</button>
8	chambre 2	2	standard	Available	<button>Delete</button>	<button>Edit</button>
9	chambre 3	2	standard	Available	<button>Delete</button>	<button>Edit</button>

## Room Reservations

[← Go Back to Menu](#)

ID	DATE	START TIME	END TIME	STATUS	USER	ROOM	ACTIONS
1	2024-12-25	10:00	12:00	Confirmed	admin	salle vip5(famille)	<button>Confirmed</button> <button>Update</button>
2	2024-12-26	08:39	11:39	Confirmed	user1	salle vip5(famille)	<button>Confirmed</button> <button>Update</button>
3	2024-12-12	04:40	01:44	Confirmed	user1	salle vip4	<button>Confirmed</button> <button>Update</button>
4	2024-12-27	06:02	08:02	Pending	user1	salle vip1	<button>Pending</button> <button>Update</button>
5	2024-12-26	00:44	04:41	Rejected	aya	salle vip5(famille)	<button>Rejected</button> <button>Update</button>

## ADMIN STATISTICS

### TOTAL RESERVATIONS

7

### MOST BOOKED ROOMS

ROOM ID	NUMBER OF RESERVATIONS
1	3
9	2

### MOST BOOKED ROOMS

ROOM ID	NUMBER OF RESERVATIONS
1	3
9	2
2	1
5	1

### RESERVATIONS PER DAY

DATE	NUMBER OF RESERVATIONS
2024-12-12	1

## RESERVATIONS PER DAY

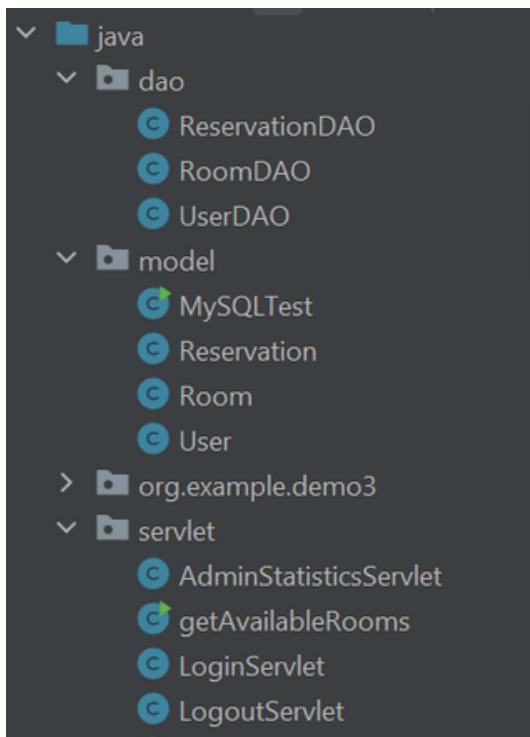
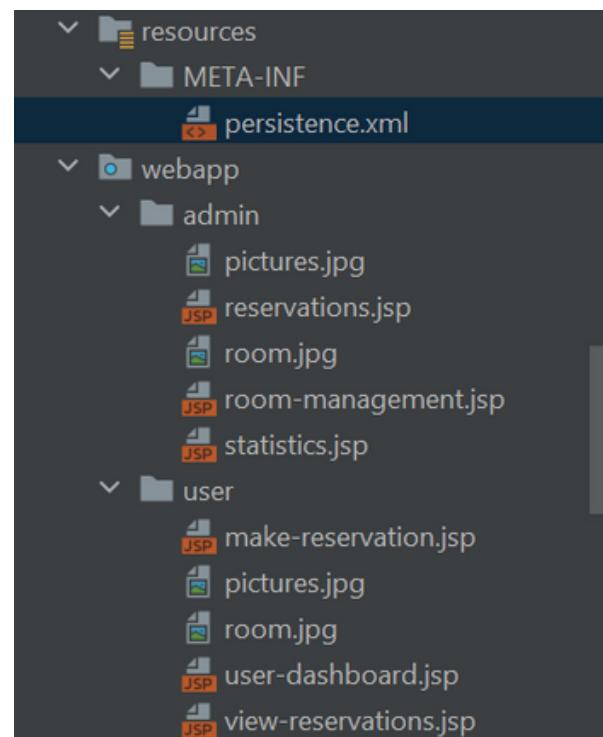
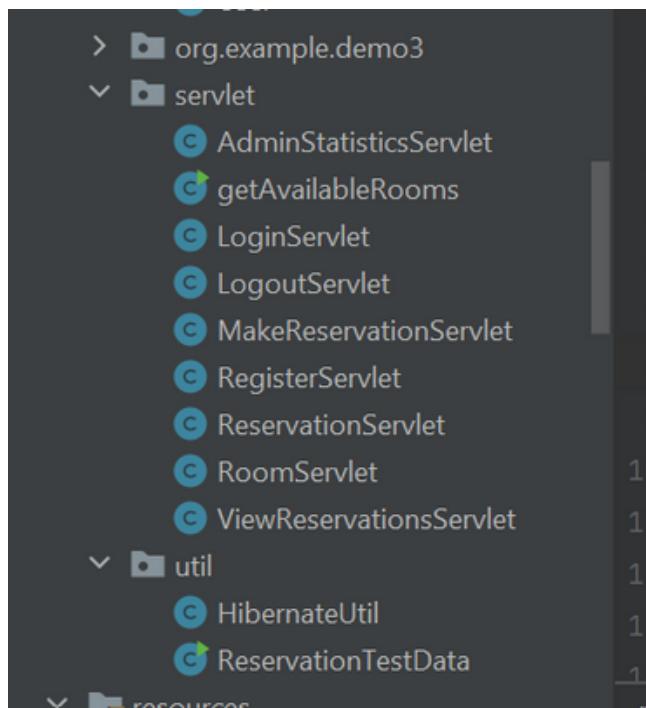
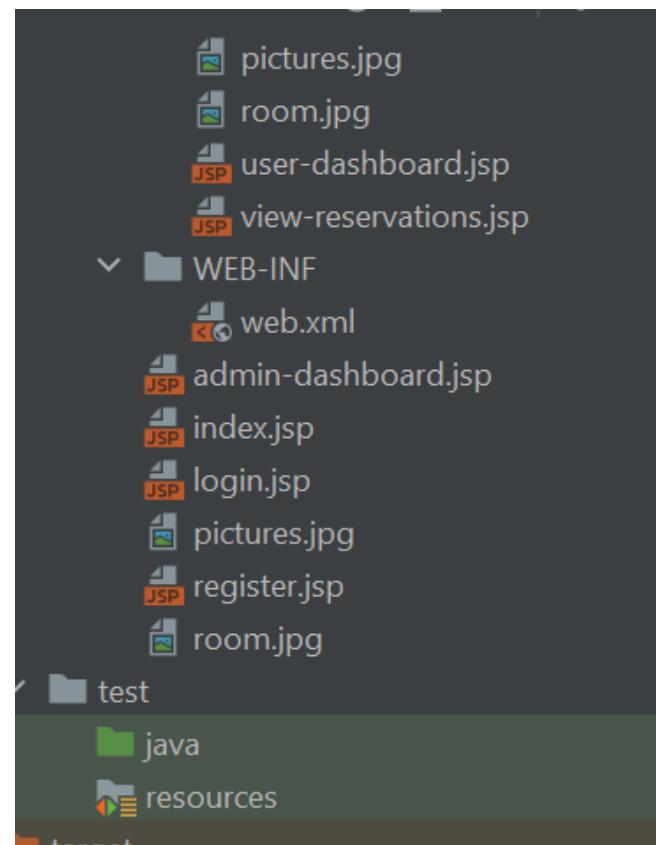
DATE	NUMBER OF RESERVATIONS
2024-12-12	1
2024-12-25	1
2024-12-26	2
2024-12-27	1
2024-12-30	1
2025-01-01	1

## RESERVATIONS BY USER

USER ID	NUMBER OF RESERVATIONS
2	3
1	1
3	1
7	1
10	1

[← Back to Dashboard](#)

# ORGANISATION DES FICHIERS:

**1**

**3**

**2**

**4**


Composant	Type de fichier	Fonctionnalité principale	Exemple
<b>Modèle (Model)</b>	<b>Entités</b>	Représente les tables de la base de données sous forme d'objets Java. Définit les attributs et leurs mappings avec la base de données.	<code>Room.java</code>
	<b>Hibernate</b>		
	<b>DAO (Data Access Object)</b>	Fournit des méthodes CRUD (Create, Read, Update, Delete) pour interagir avec la base de données. Utilise Hibernate pour exécuter les opérations.	<code>RoomDAO.java</code>
<b>Vue (View)</b>	<b>JSP (JavaServer Pages)</b>	Génère des pages web dynamiques. Contient le HTML et le code pour afficher les données envoyées par le contrôleur.	<code>rooms.jsp</code>
<b>Contrôleur</b>	<b>Servlets</b>	Gère les requêtes HTTP. Contient la logique métier, appelle les DAO pour manipuler les données, et transmet les résultats aux JSP.	<code>RoomServlet.java</code>
<b>Utilitaires</b>	<b>Hibernate Util</b>	Configure et gère les sessions Hibernate pour les connexions à la base de données.	<code>HibernateUtil.java</code>
	<b>Filtre de sécurité</b>	Vérifie les sessions et protège les pages réservées aux utilisateurs authentifiés. Redirige vers la page de connexion si nécessaire.	<code>AuthFilter.java</code>