24444 Bertrand RWOGERA

***Hotel room reservation system***

The role of Hotel reservation System:

the Hotel reservation System streamlines the process of registering user for reserving aa room and provides a centralized platform for managing reservation information. It enhances communication between the hotel and individuals seeking rooms, making the entire process more efficient and organized.

1. **Here are some KPI (Key performance Indicators) against Hotel room system such as:**

* **Registration:** The system allows customer to register for reservation by providing their personal information such as name, email, password, and confirm password. The registration form may also include additional fields specific to the hotel's requirements.
* **User Management:** The system maintains a database of registered user. It enables administrators to view and manage user accounts, including adding new users, editing user details, and deleting user accounts if needed.
* **Course Information:** The system provides information about the hotel rooms available within the hotel. This includes room descriptions, schedules, locations, and any prerequisites or requirements for participating in the rooms.

Here are steps I have followed to develop my spring mvc(Model-View-Controller) web application

Open Intellij and select "File" -> "New" -> "Other".

In the "New" dialog box, expand the "Maven" folder and select "Maven Project". Click "Next".

In the "New Maven Project" dialog, specify the Group Id, Artifact Id, and Version for your project. These parameters help identify your project and generate a unique identifier for it.

When creating a Spring MVC project, there are several frameworks and best practices you can follow to enhance your development process and improve the overall structure and efficiency of your application.

* **Thymelea**f: it allow you to create dynamic web pages by combining HTML templates with server-side data.
* **Spring Security**: If your application requires authentication and authorization, Spring Security provides robust security features.
* **Hibernate or Spring Data JPA**: If your application interacts with a relational database, Hibernate or Spring Data JPA can simplify database operations.
* **Spring web:** Spring Web provides a Model-View-Controller (MVC) architecture that allows for clear separation of concerns between the application's data, presentation, and user interaction.
* **Spring DevTools:** this framework support remote development scenarios. It allows you to run your application on a remote server, provides automatic application restart capabilities.
* **Compatibility:** chosen database driver must be compatible with the database management system (DBMS) you are using. Different databases (e.g., MySQL, PostgreSQL, Oracle, SQL Server) have their own specific drivers. For me I use PostgreSQL

Configure the project dependencies by:

Open the project's pom.xml file.

Add the required Spring MVC dependencies, such as spring-webmvc, spring-core, spring-context, and any additional dependencies you need.

Save the pom.xml file, and the dependencies will be downloaded automatically.

When developing a Spring MVC project, you typically need to include several packages from the Spring framework such as:

* controller
* model
* repository
* service layer

a controller plays a crucial role in handling incoming HTTP requests, processing the requests, and r

returning an appropriate response to the client.

The main controller class is typically annotated with some annotations to indicate that it is responsible for handling HTTP requests and serving as a component of the MVC architecture.

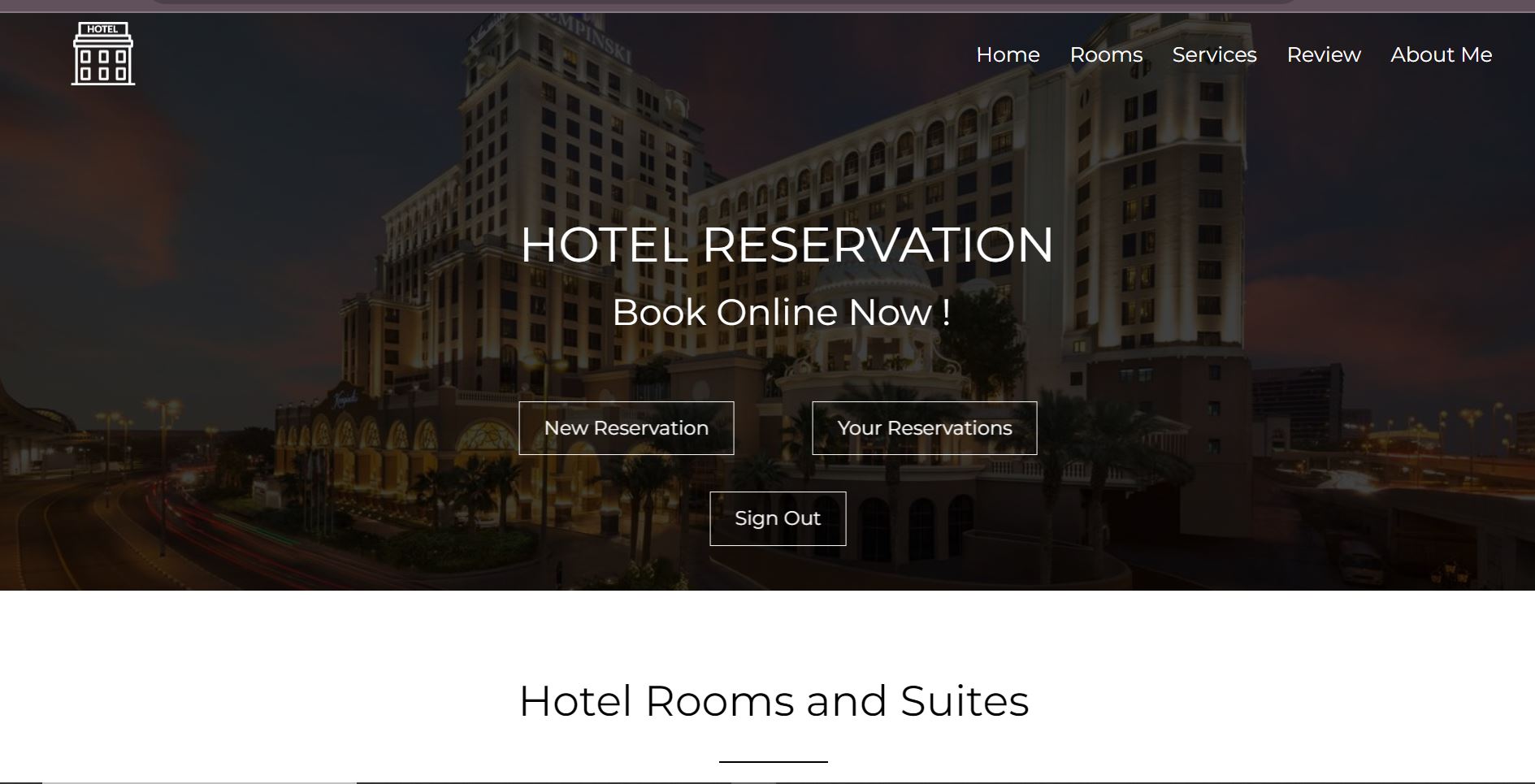
@**Controller**: This annotation marks a class as a controller component in Spring MVC.

@**RequestMapping**: This annotation is used to define the base URL or URL patterns that the controller handles.

@**GetMapping**, @**PostMapping**, @**PutMapping**, @**DeleteMapping**: These annotations are used to handle specific HTTP methods (GET, POST, PUT, DELETE) for a given URL pattern.

* **a mode**l represents the data that is passed between the controller and the view. It plays a crucial role in facilitating the exchange of information, enabling the controller to prepare the data and the view to render it appropriately.
* **repository**  is responsible for defining and implementing methods to access and manipulate data stored in the database.
* **The service layer** plays a crucial role in implementing the business logic and coordinating the interactions between the controller and the repository (data access layer).
* **@PathVariable**: This annotation is used to bind a URL variable to a method parameter. It extracts the value from the URL and maps it to the corresponding method argument.
* Templates are responsible for rendering the dynamic content that is displayed to the user.
* Templates use templating languages or frameworks (e.g., JSP, Thymeleaf, FreeMarker) to access and display the data received from the controller.

# My home page view



This is the main entry point of my web application, it serves as the initial page that users see after they logged in.

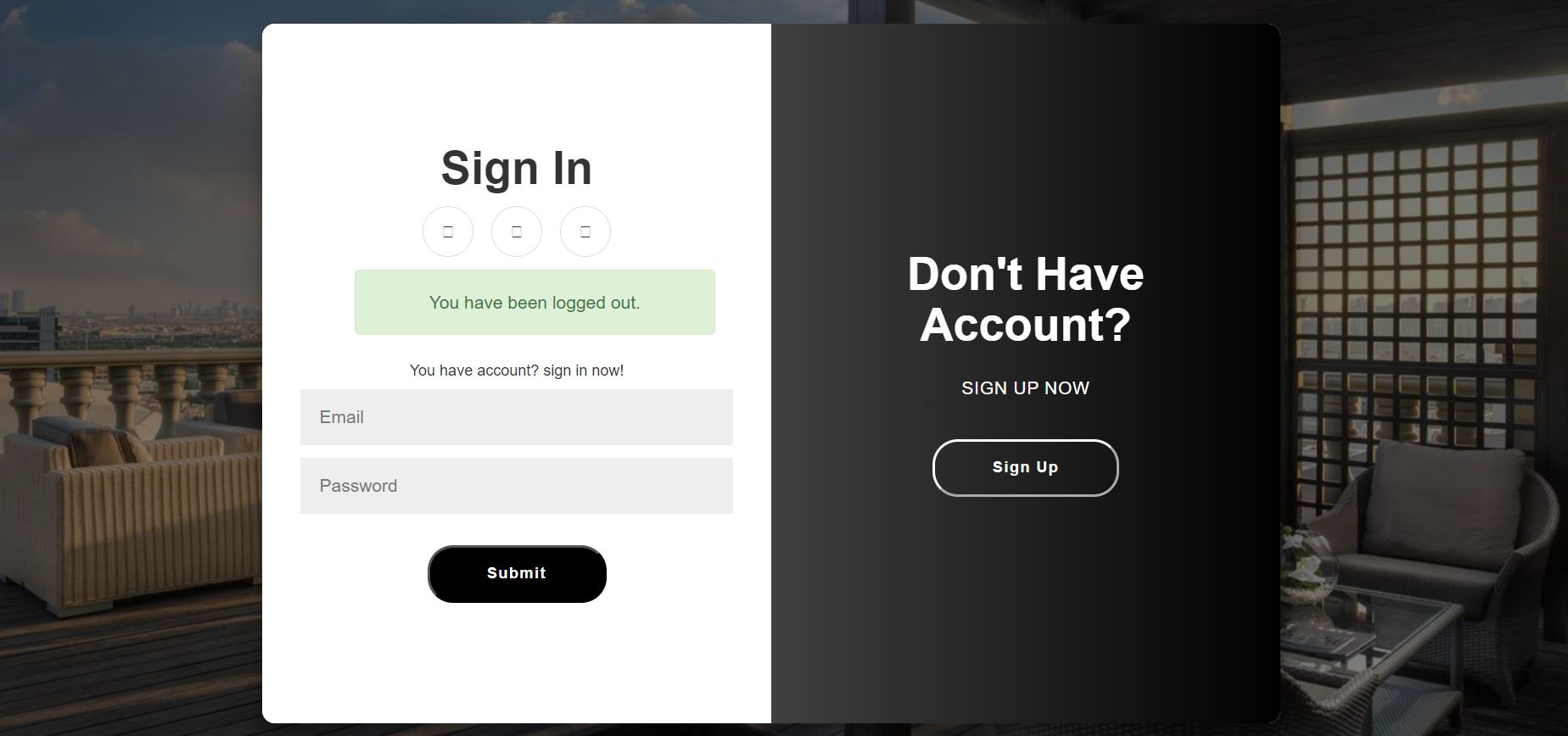
Navigation links that allows you to manage my web application and gives access to your new reservation and the reservation you already have or signout.

Navigation link that allows you to Log in case you are the user who have an account.

Navigation link that allows to manage all the contents in my web application.

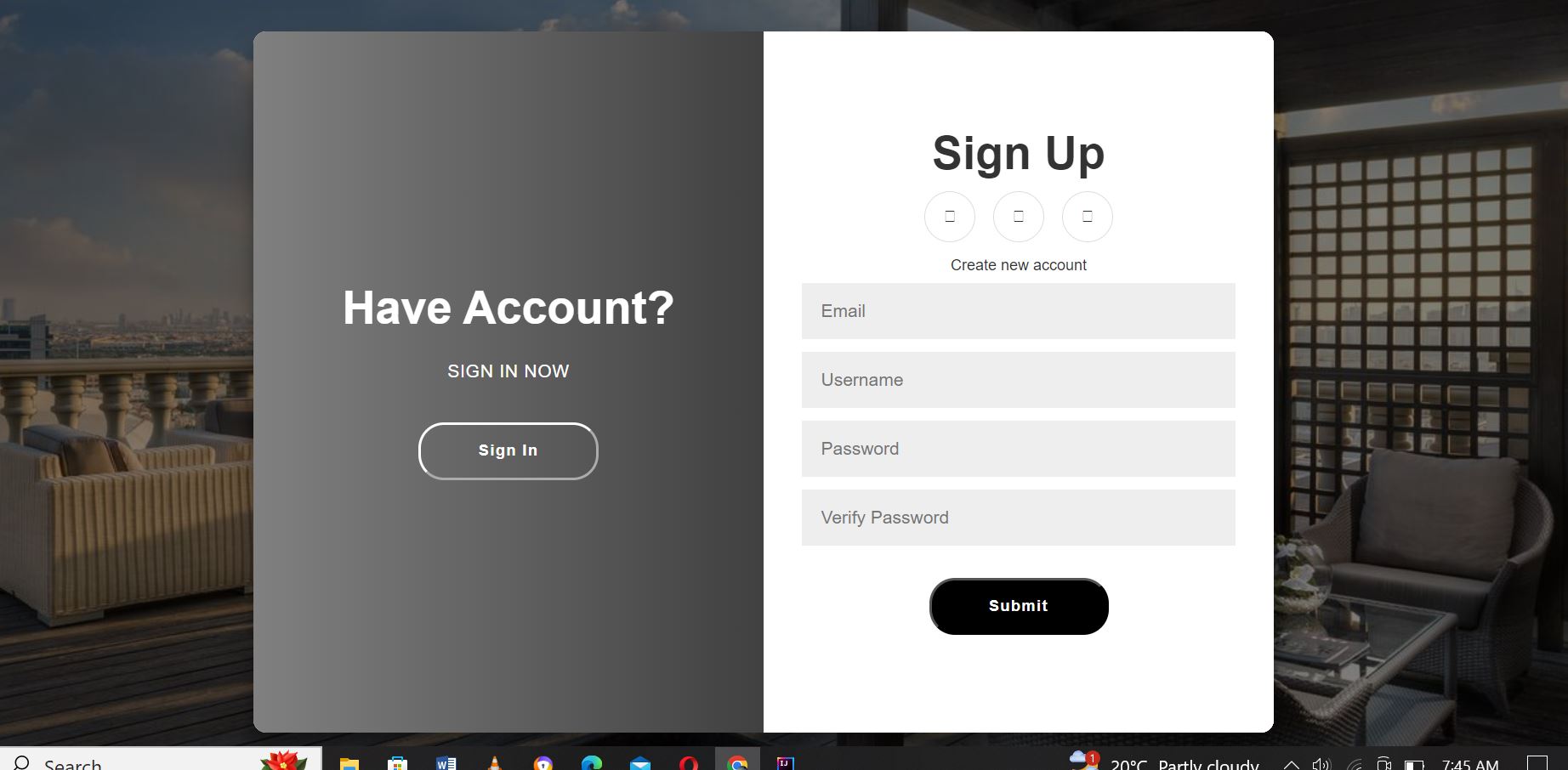
Here you can use localhost://8080

# User authentication and access control to Log in

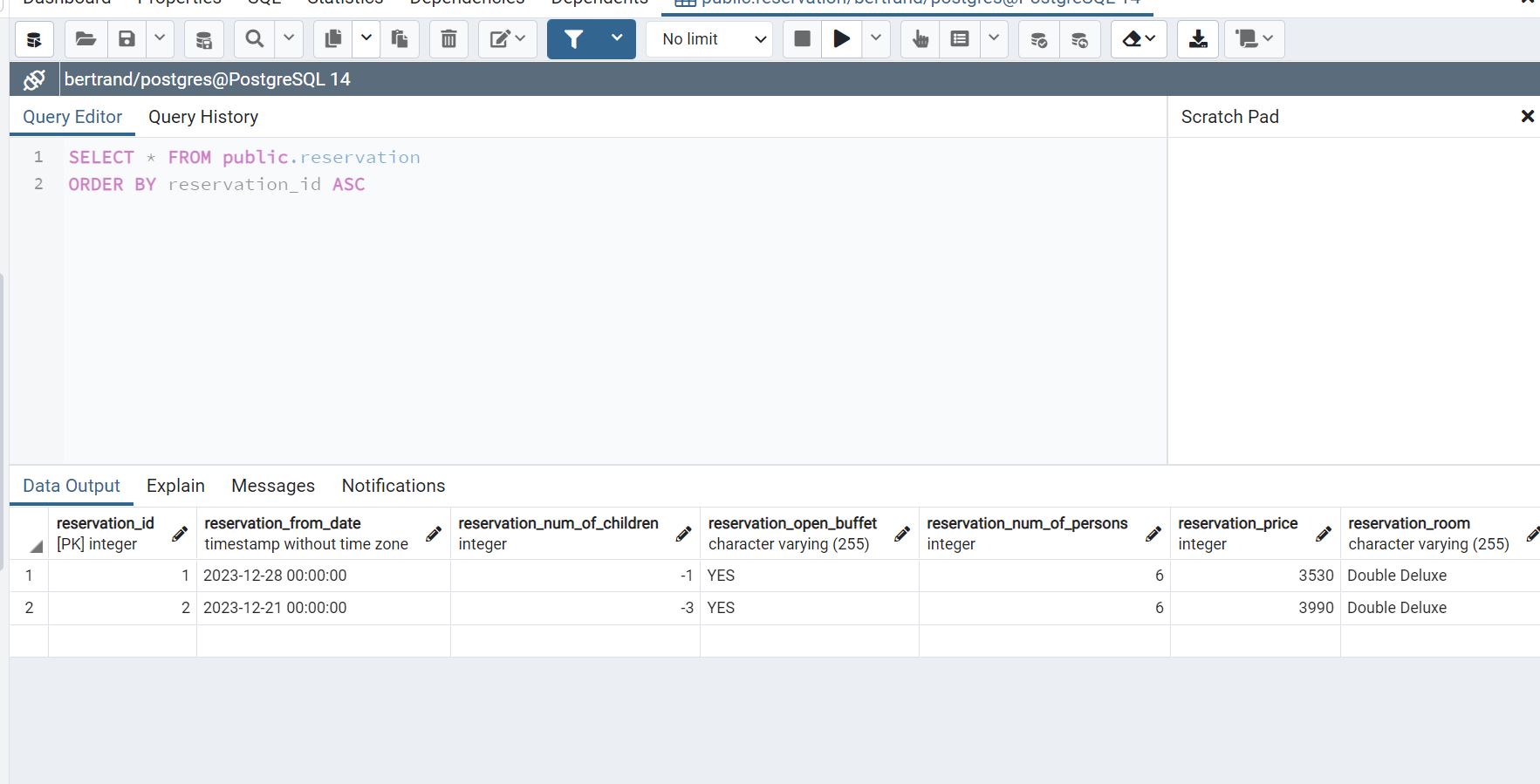


The login functionality verifies the identity of the user by authenticating their credentials, such as email and password.

It ensures that only authorized users can access protected resources and perform specific actions within the application.



* The sign-up feature enables users to register themselves by providing necessary information, such as username, email address, and password.
* It collects user details required to create a unique account and establish their identity within the application.
* The sign-up functionality creates a new user account in the application's user management system or database.



This the database



This how it Displayed in web page and user can be able to edit and Delete

Some of The room User can Access After Log In

Thank You