Hotel Customer Booking Portal

This project is a **Hotel Customer Booking Portal** developed using Spring MVC, Spring Boot, Hibernate, and MySQL. The application is controlled by an admin who can manage customer bookings, including adding, viewing, updating, and deleting records.

Features

1. **Add Customer Booking:**
   - Admin can add new customer booking details such as name, number of people, room type, check-in, and check-out dates.
2. **View Bookings:**
   - Admin can view a list of all customer bookings.
3. **Update Booking:**
   - Admin can update existing booking details.
4. **Delete Booking:**
   - Admin can delete a customer booking.

Technologies Used

- **Backend:**
  - Spring MVC
  - Spring Boot
  - Hibernate
- **Frontend:**
  - JSP Pages
- **Database:**
  - MySQL

Project Structure

Entity Layer

- Represents the customer booking table with attributes like:
  - `cid` (Primary Key)
  - `name`
  - `noofpeople`
  - `roomtype`
  - `checkin`
  - `checkout`

DAO Layer

- Handles database operations using Hibernate for:
  - Adding a customer booking
  - Retrieving all customer bookings
  - Deleting a booking by `cid`
  - Updating a booking

Service Layer

- Provides a service interface between the DAO layer and the Controller.

Controller Layer

- Handles HTTP requests and routes them to appropriate JSP pages for:
    - Home Page
    - Add Customer Page
    - Display Bookings Page
    - Update Booking Page
    - Delete Booking

Frontend Pages

- `index.jsp`: Home page.
- `add.jsp`: Page for adding customer bookings.
- `display.jsp`: Page for displaying all bookings.
- `updatepage.jsp`: Page for updating a specific booking.

Project Flow

```
Hotel Customer Booking Portal
|
|-- Entity Layer
|   |-- Customer.java
|
|-- DAO Layer
|   |-- CusDao.java
|
|-- Service Layer
|   |-- CusService.java
|
|-- Controller Layer
|   |-- PageController.java
|
|-- Frontend Pages (JSP)
|   |-- index.jsp (Home Page)
|   |-- add.jsp (Add Booking Page)
|   |-- display.jsp (Display Bookings Page)
|   |-- updatepage.jsp (Update Booking Page)
|
|-- Database
    |-- MySQL (Customer Table)
```

Prerequisites

1. **Java Development Kit (JDK)**: Version 8 or above.
2. **MySQL**: Database server to store customer bookings.
3. **Maven**: For managing dependencies and building the project.
4. **Spring Boot CLI**: Optional but useful for running Spring Boot applications.

Installation and Setup

1. Clone the repository:

2.
   ```
   git clone https://github.com/your-username/hotel-customer-booking-portal.git
   cd hotel-customer-booking-portal
   ```

3. Set up the database:
   - Create a database named `hotel_booking` in MySQL.
   - Update the `application.properties` file with your MySQL credentials:
   - `spring.datasource.url=jdbc:mysql://localhost:3306/hotel_booking`
   - `spring.datasource.username=your-username`

   -
     ```
     spring.datasource.password=your-password
     spring.jpa.hibernate.ddl-auto=update
     ```

4. Build and run the application:

5.
   ```
   mvn clean install
   mvn spring-boot:run
   ```

6. Access the application:
   - Open a web browser and navigate to `http://localhost:8080/home`.

API Endpoints

- **Home Page:** `GET /home`
- **Add Customer Page:** `GET /add-page`
- **Add Customer:** `POST /add-customer`
- **View All Customers:** `GET /get-customer`
- **Delete Customer:** `GET /delete-customer?cid={cid}`
- **Get Customer for Update:** `GET /get-to-update-customer?cid={cid}`
- **Update Customer:** `POST /update-customer`