

HOTEL MANAGEMENT SYSTEM PROJECT

Presented by - Nitish Veni

Trained By- Subbu



OVERVIEW



01

Problem
Statement

02

Key Features

03

Goals

04

Technologies
Used

05

Project
Architecture

06

API
Testing

07

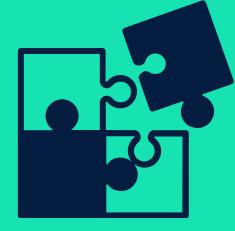
Challenges
Faced

08

Outcomes

01

PROBLEM STATEMENT



TO DEVELOP A ROBUST AND USER-FRIENDLY
HOTEL MANAGEMENT SYSTEM FOR
ADMINISTRATORS TO EFFICIENTLY MANAGE
HOTEL OPERATIONS.

02 KEY FEATURES

01

Login and Authentication-
Secure login for administrators with role-based access.

02

Data Management- CRUD Operations for Bookings, Hotels, Guests & Facilities

03

Search Functionality & Administrator registration-
Enable searching of records using IDs and key fields. Support registration of new administrators.

04

Search Functionality-
Enable searching of records using IDs and key fields.



03 GOALS & OBJECTIVES

Goals

To streamline hotel management operations with a scalable and secure system for administrators.

Objectives

Implement secure authentication, manage data efficiently, enable advanced search, onboard new administrators, and develop a user-friendly Angular interface with a robust Spring Boot backend.



04 TECHNOLOGIES USED



FRONT-END -- ANGULAR-18

BACK-END -- SPRINGBOOT

DATABASE -- MYSQL

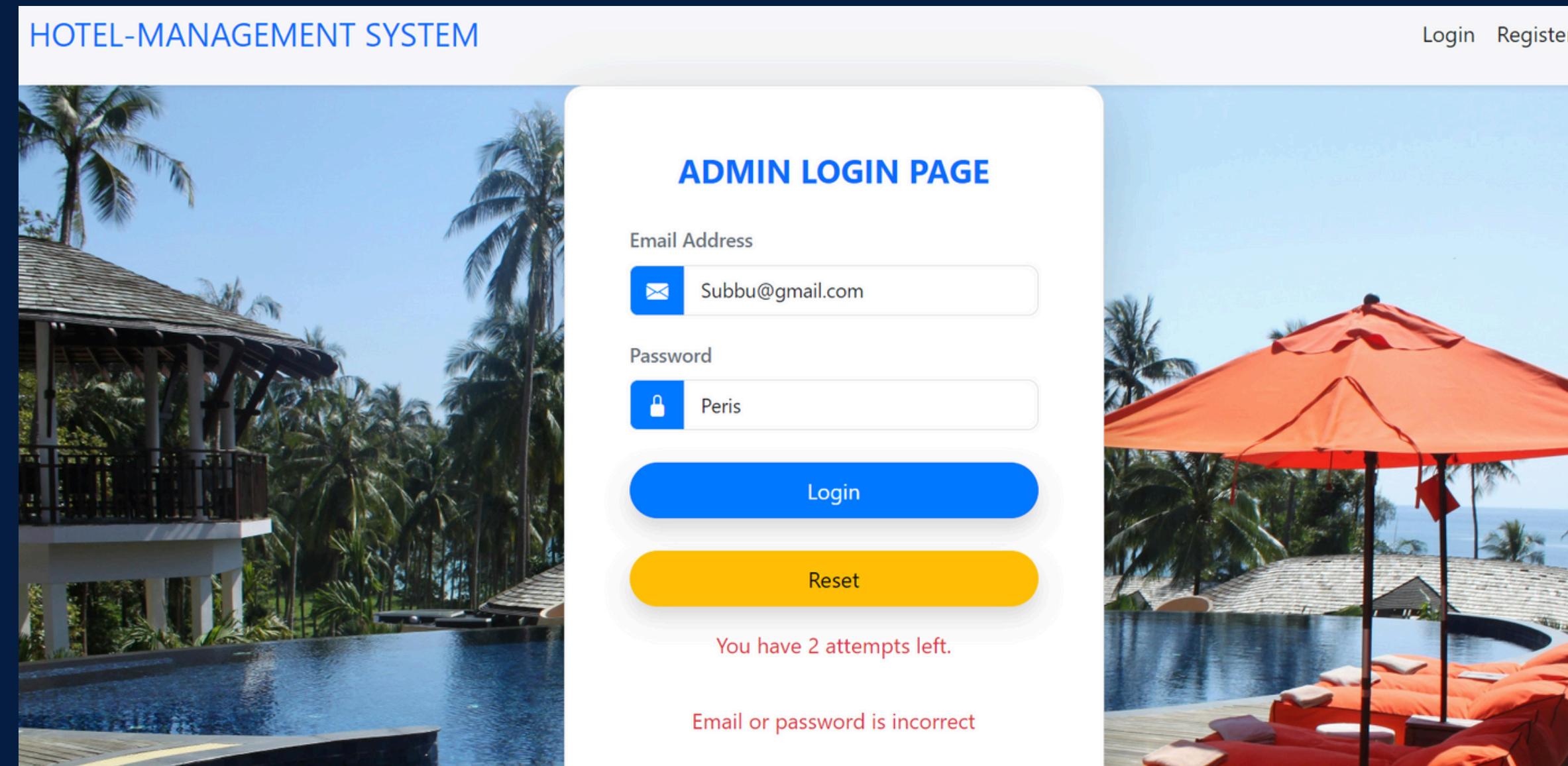
API TESTING -- POSTMAN

DESIGN -- BOOTSTRAP

IDE -- VS CODE & INTELLIJ IDEA

05

PROJECT ARCHITECTURE



Login Page with Validation

05

PROJECT ARCHITECTURE

Register Here!!

Full Name

Email

Password

Confirm Password

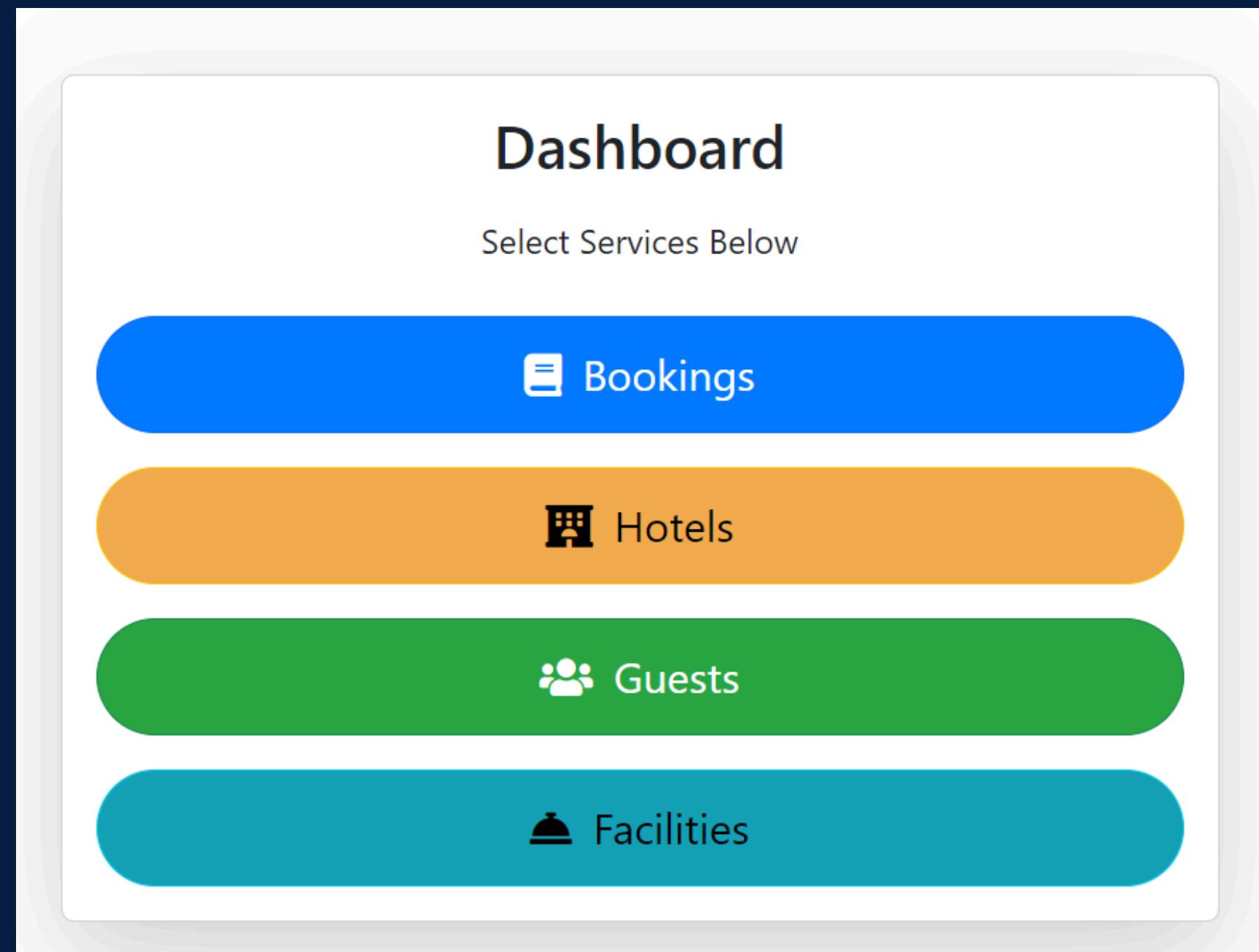
Register

Already have an account? [Login here](#)

New Admin
Registration

05

PROJECT ARCHITECTURE



Dashboard For
Admin

05

PROJECT ARCHITECTURE

[Back](#)

Search Add Booking

Booking Details

Booking ID	Hotel ID	Guest ID	Check-in Date	Check-out Date	Total Amount	Actions
203	0	2	12/10/2024	12/10/2024	\$22.00	<button>Update</button> <button>Delete</button>
252	9	25	12/10/2024	12/10/2024	\$123.00	<button>Update</button> <button>Delete</button>
253	12	12	12/03/2024	12/02/2024	\$123.00	<button>Update</button> <button>Delete</button>
254	2	23	12/10/2024	01/11/2025	\$1,245.00	<button>Update</button> <button>Delete</button>

Bookings Page with CRUD Operations & Search Functionality

PROJECT ARCHITECTURE

```
@Table(name = "BOOKINGS")
public class Booking {

    @Id 3 usages
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(name = "BOOKING_ID")
    private int bookingId; // Primary key field

    @Column(name = "GUEST_ID", nullable = false) 3 usages
    private int guestId;

    @Column(name = "HOTEL_ID", nullable = false) 3 usages
    private int hotelId;

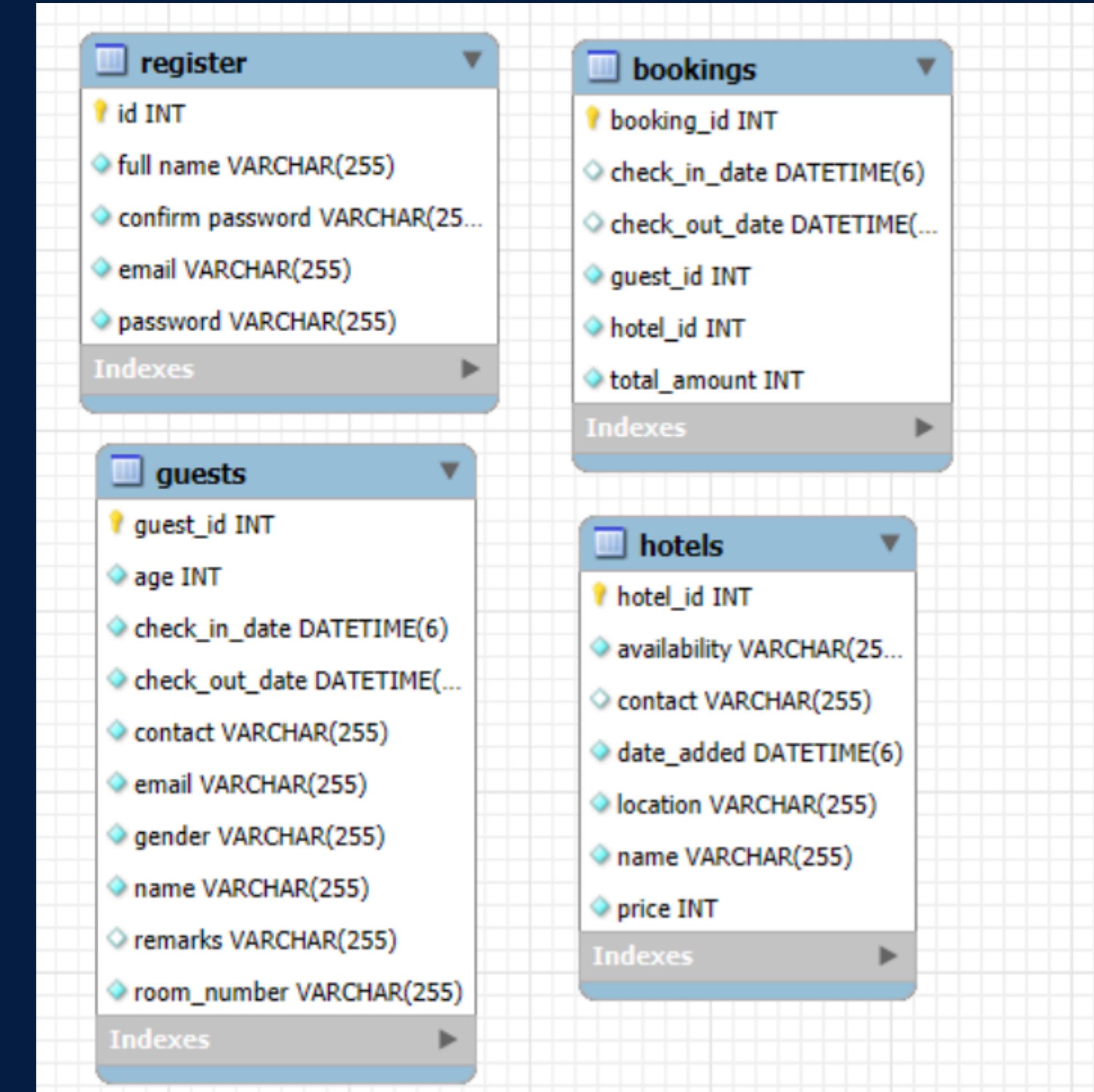
    @Column(name = "CHECK_IN_DATE") 3 usages
    private Date checkInDate;

    @Column(name = "CHECK_OUT_DATE") 3 usages
    private Date checkOutDate;

    @Column(name = "TOTAL_AMOUNT", nullable = false) 3 usages
    private int totalAmount;

    public Booking() { no usages }
```

PROJECT ARCHITECTURE



06

RESTAPI TESTING(POSTMAN)

The screenshot shows the Postman application interface. On the left, a sidebar lists three tests: "GET Register" (selected), "GET Booking", and "GET Hotel". The main area displays a test for the "Register" endpoint. The request method is "GET" and the URL is "http://localhost:8080/api/v6/register". Below the request details, there are tabs for "Params", "Authorization", "Headers (7)", "Body", "Scripts", "Settings", and "Cookies". The "Headers (7)" tab is selected. Under "Query Params", there is a table with columns: Key, Value, Description, and Bulk Edit. The table is currently empty. At the bottom of the request section, there are tabs for "Body", "Cookies", "Headers (8)", and "Test Results". The "Test Results" tab is selected, showing a green status bar with "200 OK", "23 ms", "565 B", and other metrics. Below the status bar, there are buttons for "Pretty", "Raw", "Preview", "Visualize", and "JSON". The "Pretty" button is selected. The "Body" section displays the JSON response from the API, which is a single object with an id of 1 and a name of Nitish.

```
1 [  
2 {  
3   "id": 1,  
4   "email": "nitish@asc.com",  
5   "password": "Nitish@2002",  
6   "confirmPassword": "Nitish@2002",  
7   "name": "Nitish"  
8 },
```

07 CHALLENGES & LEARNINGS



- ▶ MAPPING FRONTEND TO BACK END
- ▶ AUTOGENERATING TABLES IN SPRINGBOOT
- ▶ UPDATE FUNCTIONALITY IN CONTROLLER FILE
- ▶ NAMING CONVENTIONS IN SQL AND ANGULAR

FUTURE ENHANCEMENT

PRIMARY KEY & FOREIGN KEY
SUPER ADMIN

THANK YOU

