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

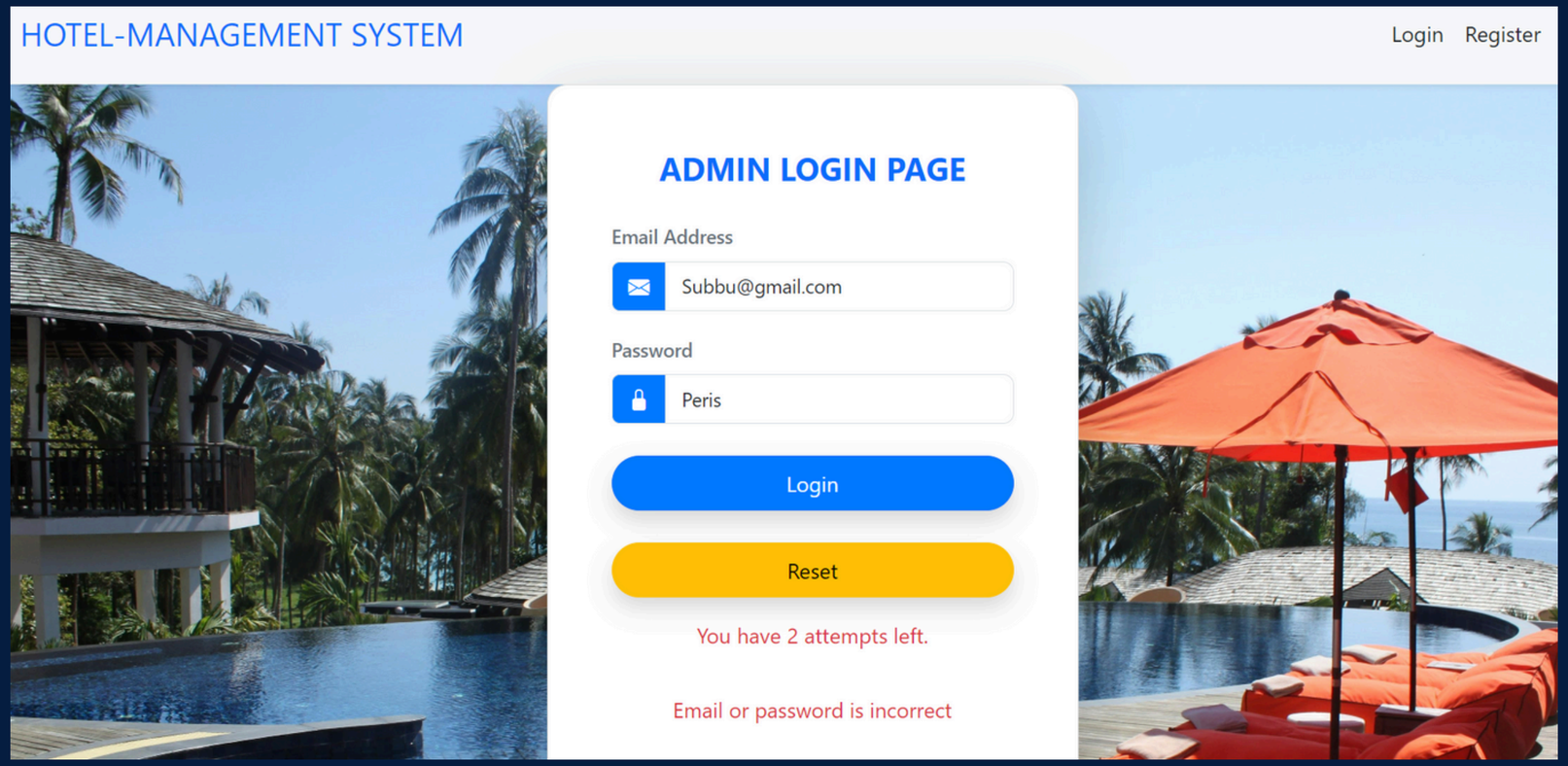
API TESTING -- POSTMAN

DESIGN -- BOOTSTRAP

IDE -- VS CODE & INTELLIJ IDEA

05

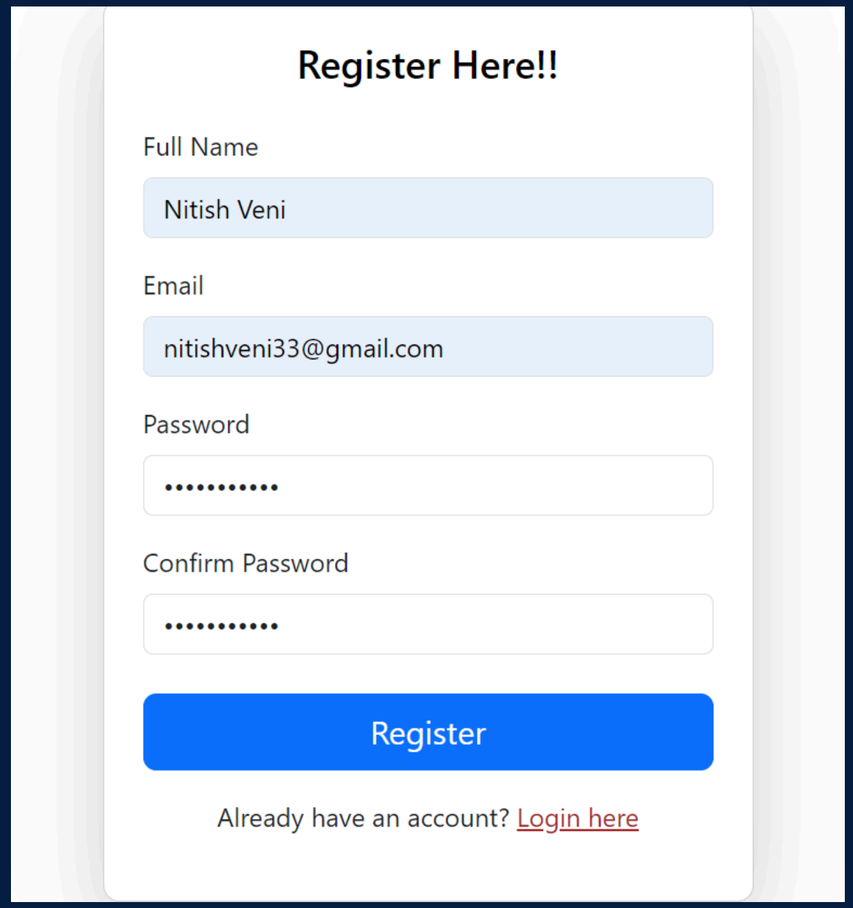
PROJECT ARCHITECTURE



Login Page with Validation

05

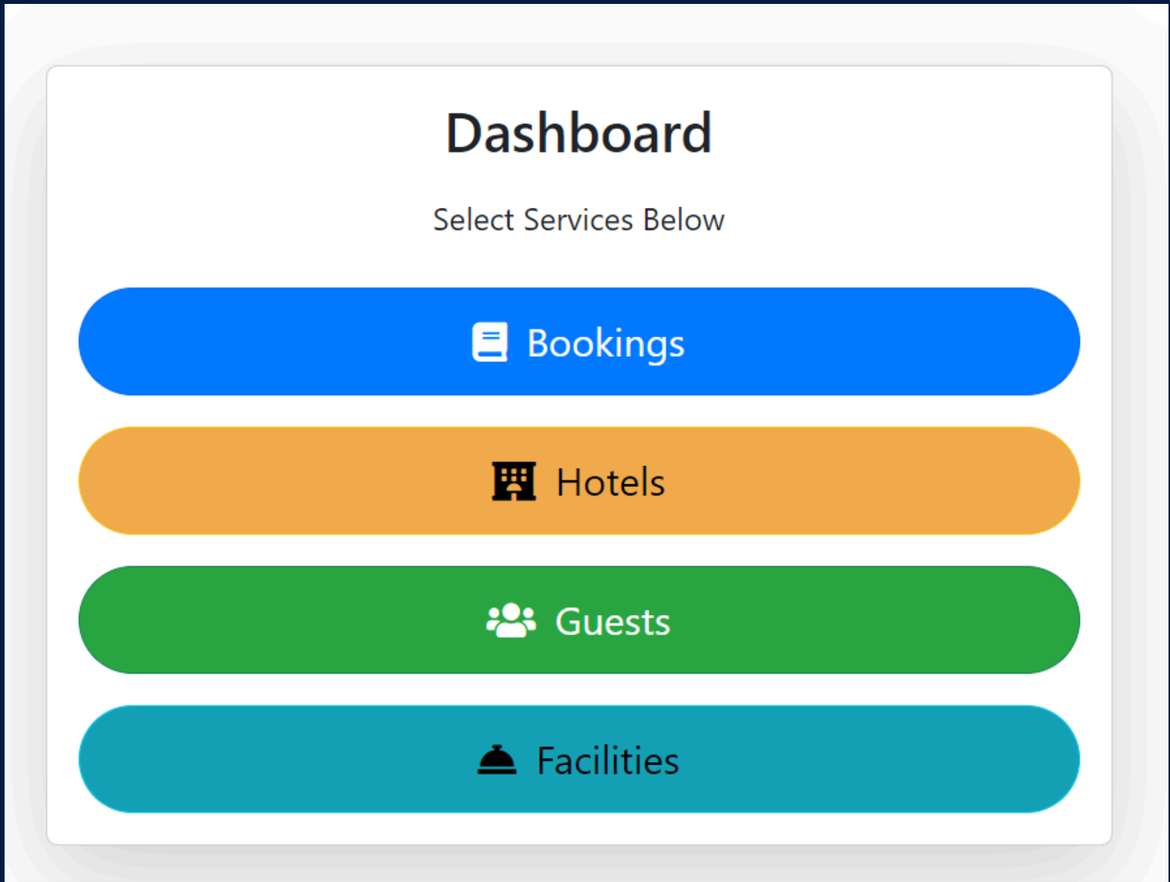
PROJECT ARCHITECTURE



New Admin Registration

05

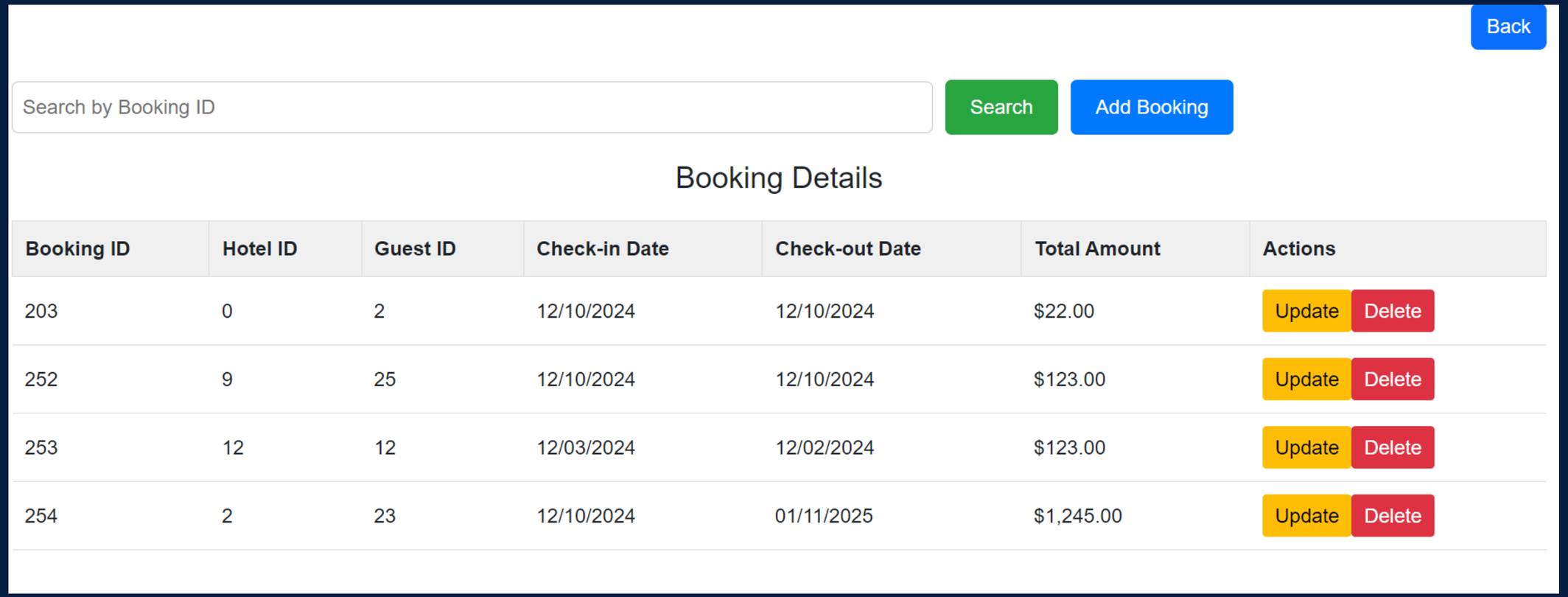
PROJECT ARCHITECTURE



Dashboard For Admin

05

PROJECT ARCHITECTURE



Bookings Page with CRUD Operations & Search Functionality

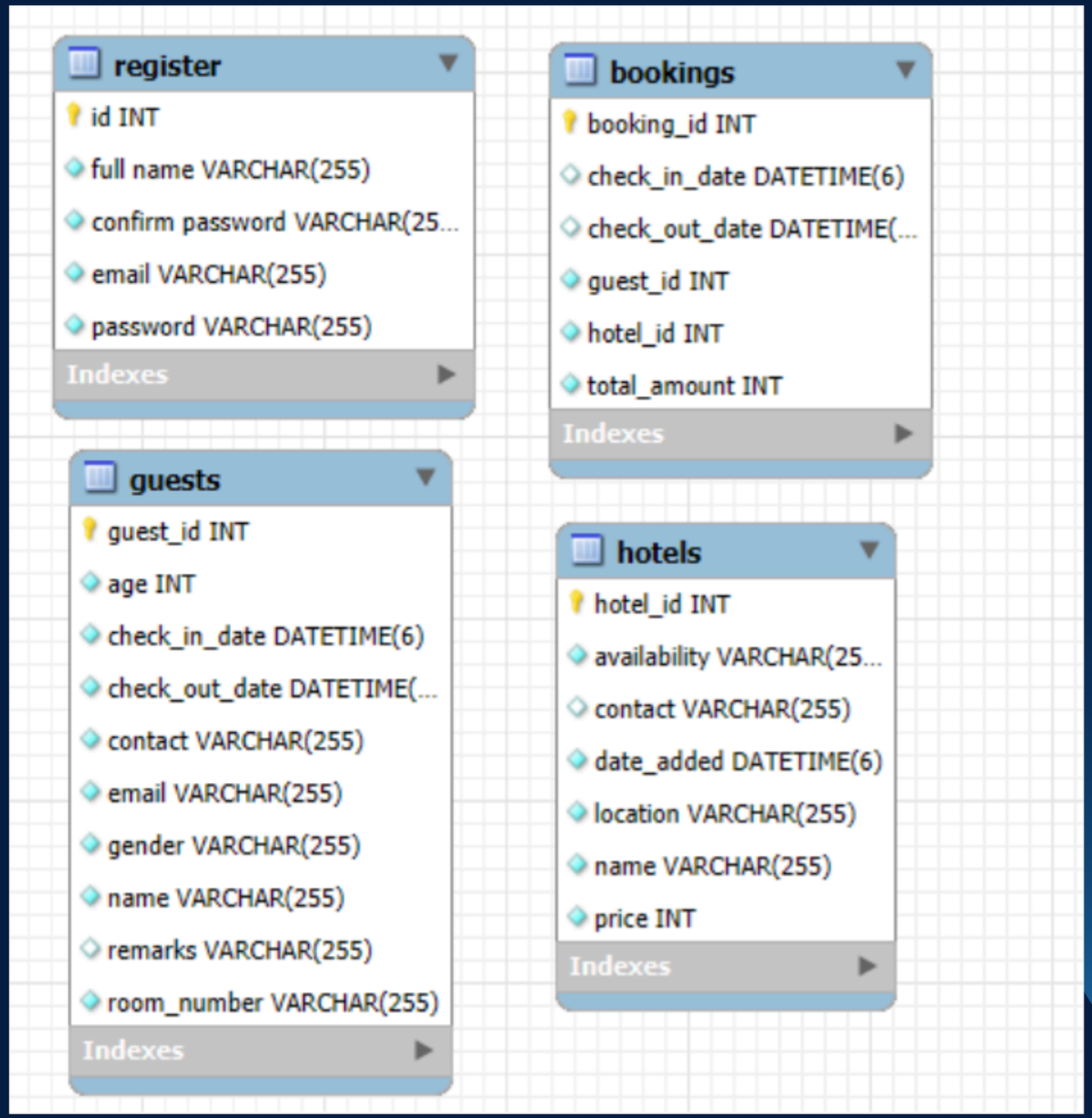
05

PROJECT ARCHITECTURE

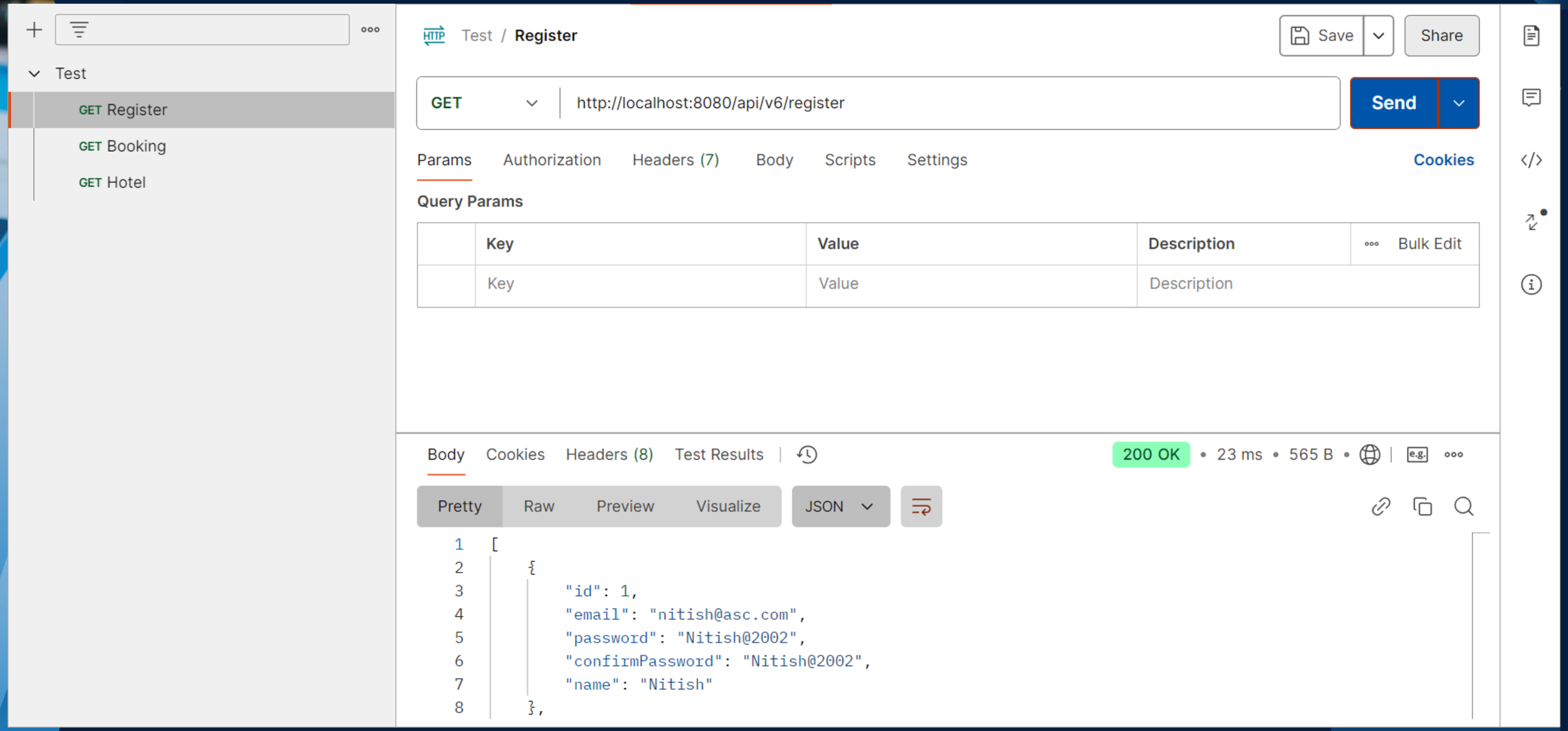
```java
@Table(name = "BOOKINGS")
public class Booking {

    @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
```

05

PROJECT ARCHITECTURE



06 RESTAPI TESTING(POSTMAN)



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