Hotel AI Chatbot Project

Jeremiah Dy

Project duration: 1 month (June 2025)

Project Overview

- Create a microservice-based application that models a hotel website’s basic functionality, such as listing hotel information and allowing users to book stays
  - REST APIs for backend data processing and retrieval
  - React.js for frontend user experience
- Use JWT authentication for stateless security, protecting secure content
- Create an AI-powered chatbot that extends and integrates into the core website’s basic functionality, allowing the user a more human experience using NLP
- Utilize vector embeddings and vector similarity search to extract relevant records from the database, regardless of dimensionality

Project Architecture

reactUI

Languages: Javascript

❑ React.js
❑ React-Bootstrap
❑ webpack
❑ babel

~~~~~~~~~~~~~~~~~~~~~~~~~~~~

This microservice serves the **frontend** of the project application.
It allows users to login/logout, view hotel and room data, and create bookings through HTTP requests to *jwtsecurity*. Upon login, a JWT token is assigned to the user and stored in sessionStorage, which will be used to authenticate all future requests to *jwtsecurity*.

jwtsecurity

Languages: Java

❑ Spring-boot
❑ Maven
❑ Spring Security
❑ Spring Data JPA
❑ PostgreSQL Database
❑ JJWT

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

This microservice serves the **security portion** of the project application.
Acting as a gateway to prevent unauthorized access of database information, this microservice issues and validate JWT tokens using RSA256 signing with public and private keys. All requests made by *reactUI* for database records must pass through *jwtsecurity*.
Upon user logout, the JWT token that was issued is stored in a “revokedList”, ensuring that the same token cannot be used again.

hotelchatbot

Languages: Java

❑ Spring-boot
❑ Maven
❑ Spring Data JPA
❑ Spring AI
❑ Apache HTTP Client 5
❑ PostgreSQL Database
❑ pgvector
❑ Jackson (library)

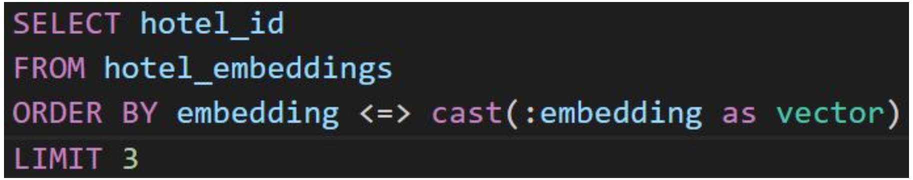
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

This microservice serves the **backend** of the project application.
It accesses the PostgreSQL database to create and view hotel, room, and booking data. It also uses HTTP Client 5 to call OpenAI response and embedding models. Embedding data is stored and used for vector similarity search to find relevant entity data for a given human-language user query.
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Spring Security x JWT Authentication

- *jwtsecurity* issues JWT token on login signed using a random UUID and a private key
- Tokens are valid for one day (development purposes)
- Frontend behavior
  - All content other than the Login/Register pages is secured behind a valid JWT token
  - Token is stored in sessionStorage, persists across page refresh, deleted on tab close
  - Secured content is wrapped in a React component that must receive a “successful authorization” from *jwtsecurity*, otherwise routes the user back to Login
- Backend behavior
  - Default Spring-Security **UsernamePasswordAuthenticationFilter** is bypassed using custom **JwtAuthenticationFilter**

Vector Similarity Search



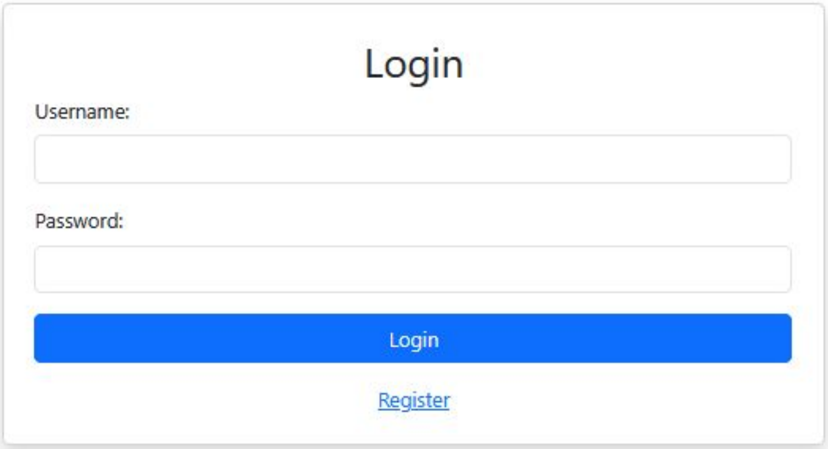
```
SELECT hotel_id
FROM hotel_embeddings
ORDER BY embedding <=> cast(:embedding as vector)
LIMIT 3
```

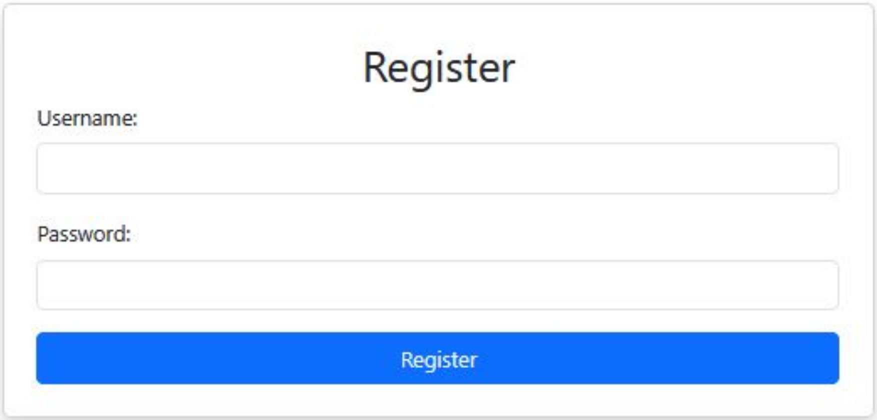
- PostgreSQL database is configured with the pgvector extension
- Upon saving hotel or hotel room data, a call is made to an OpenAI text-embedding model
- The embedding data is stored in separate tables to avoid Hibernate mapping errors
- Embedding model used: OpenAI’s *text-embedding-3-small*
- When a user asks a question in human language, the query is formatted and embedded to compare cosine distance against the existing embeddings for that entity.
  - User query: “What hotels are near the beach?”
  - Formatted user query: “hotels near beach”
  - Embeddings table to search: Hotel Embeddings

Future Improvements

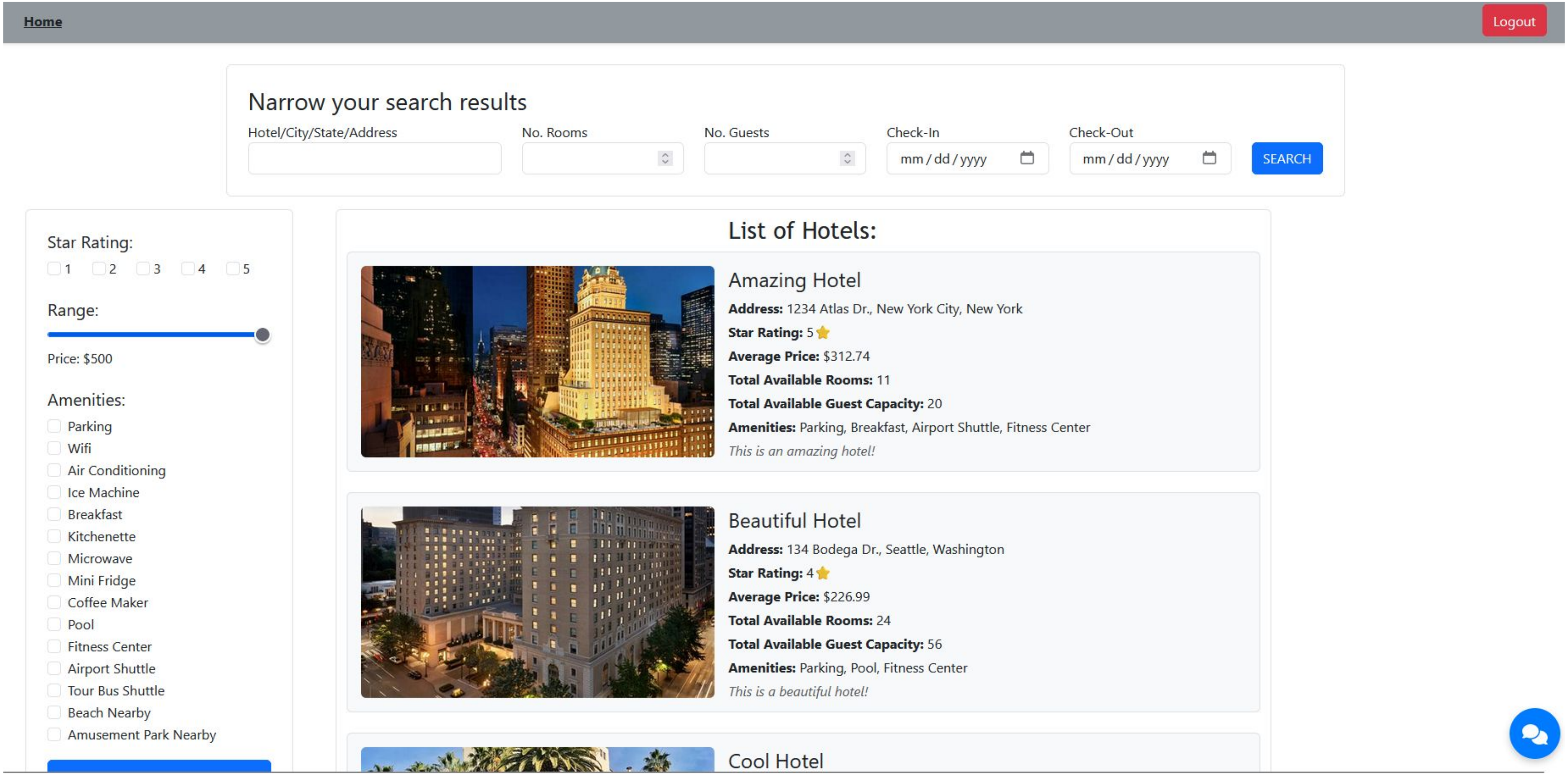
- Conversation storage per user, allow user to view past chats (also clear stale chats)
- CI/CD project flow, Dockerize microservices
- More robust multi-lingual support (currently, user must prompt for a certain language first)
- Greater hotel and hotel room data variety to test the effectiveness of vector similarity search
- More robust data validation when submitting a booking request

Login/Register





Homepage



Narrow your search results

Hotel/City/State/Address

No. Rooms

No. Guests

Check-In: mm / dd / yyyy

Check-Out: mm / dd / yyyy

Star Rating:

□ 1 □ 2 □ 3 □ 4 □ 5

Range:

Price: $500

Amenities:

- □ Parking
- □ Wifi
- □ Air Conditioning
- □ Ice Machine
- □ Breakfast
- □ Kitchenette
- □ Microwave
- □ Mini Fridge
- □ Coffee Maker
- □ Pool
- □ Fitness Center
- □ Airport Shuttle
- □ Tour Bus Shuttle
- □ Beach Nearby
- □ Amusement Park Nearby

List of Hotels:

Amazing Hotel

Address: 1234 Atlas Dr., New York City, New York

Star Rating: 5 ⭐

Average Price: $312.74

Total Available Rooms: 11

Total Available Guest Capacity: 20

Amenities: Parking, Breakfast, Airport Shuttle, Fitness Center

This is an amazing hotel!

Beautiful Hotel

Address: 134 Bodega Dr., Seattle, Washington

Star Rating: 4 ⭐

Average Price: $226.99

Total Available Rooms: 24

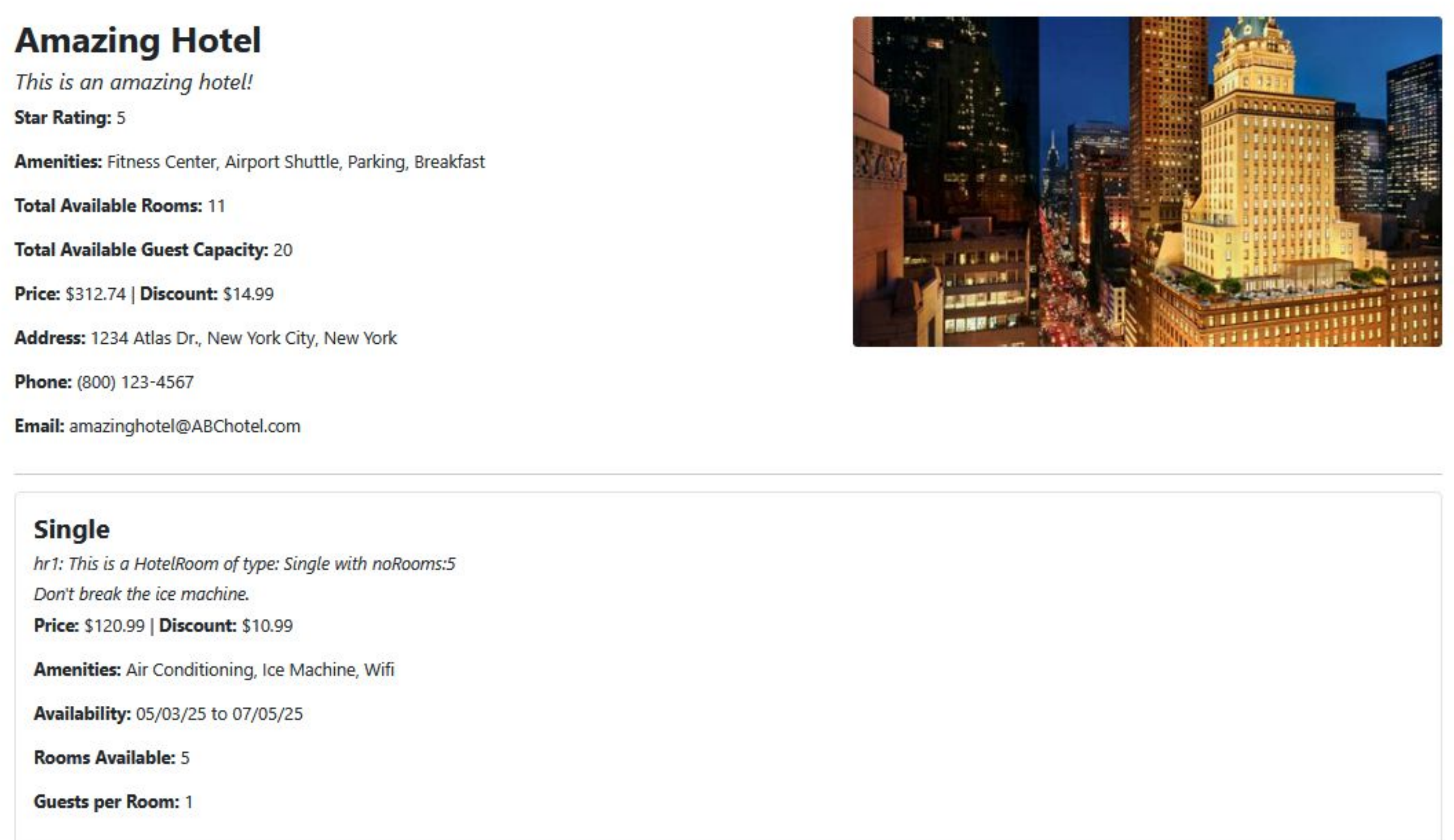
Total Available Guest Capacity: 56

Amenities: Parking, Pool, Fitness Center

This is a beautiful hotel!

Cool Hotel

Hotel Details



Amazing Hotel

This is an amazing hotel!

Star Rating: 5

Amenities: Fitness Center, Airport Shuttle, Parking, Breakfast

Total Available Rooms: 11

Total Available Guest Capacity: 20

Price: $312.74 | **Discount:** $14.99

Address: 1234 Atlas Dr., New York City, New York

Phone: (800) 123-4567

Email: amazinghotel@ABChotel.com

Single

hr1: This is a HotelRoom of type: Single with noRooms:5

Don't break the ice machine.

Price: $120.99 | **Discount:** $10.99

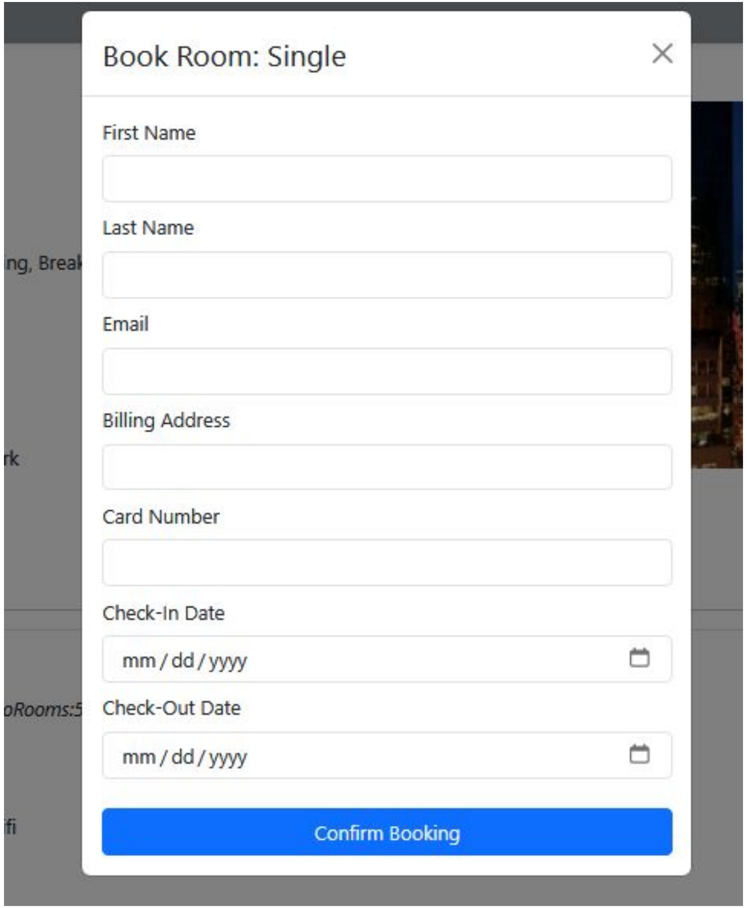
Amenities: Air Conditioning, Ice Machine, Wifi

Availability: 05/03/25 to 07/05/25

Rooms Available: 5

Guests per Room: 1

Booking popup



Book Room: Single

First Name

Last Name

Email

Billing Address

Card Number

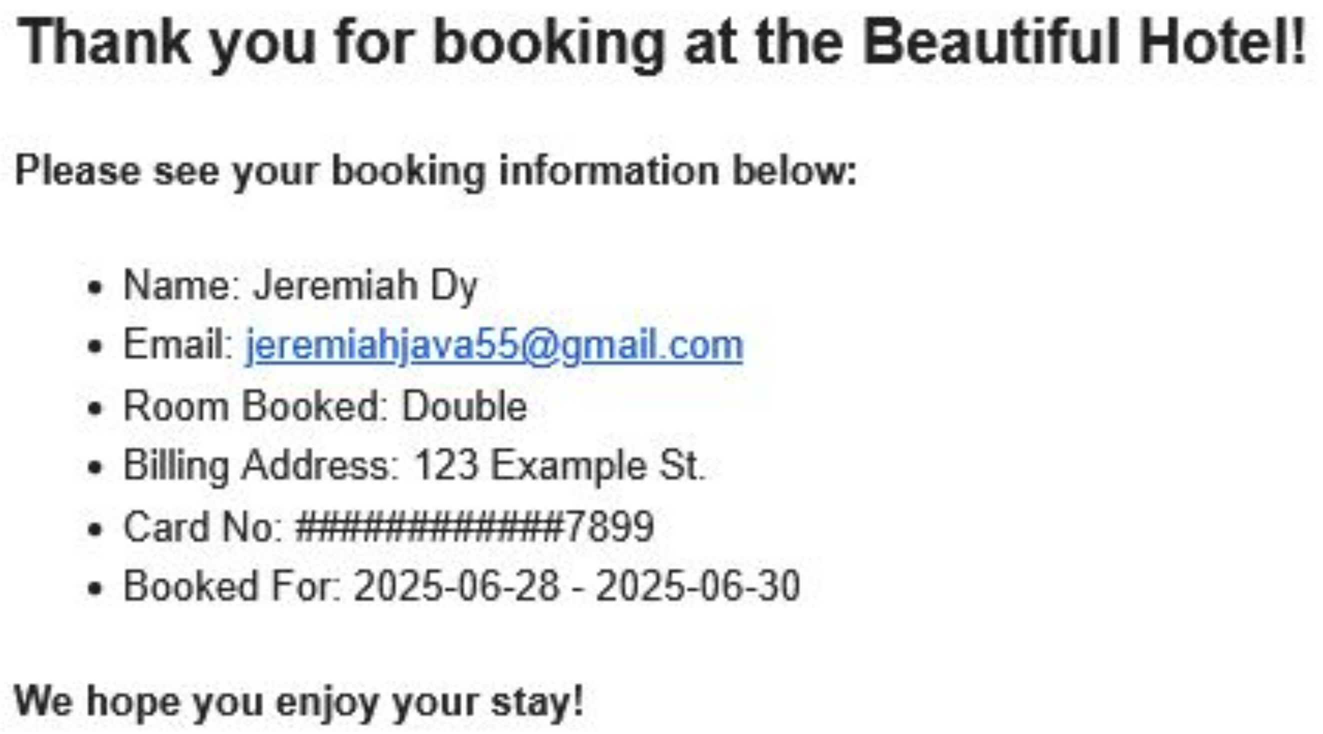
Check-In Date

mm / dd / yyyy

Check-Out Date

mm / dd / yyyy

Booking Email Confirmation



Thank you for booking at the Beautiful Hotel!

Please see your booking information below:

- Name: Jeremiah Dy
- Email: jeremiahjava55@gmail.com
- Room Booked: Double
- Billing Address: 123 Example St.
- Card No: ############7899
- Booked For: 2025-06-28 - 2025-06-30

We hope you enjoy your stay!