Github link: https://github.com/GGCav/C43-MyBnb.git

**Description:**

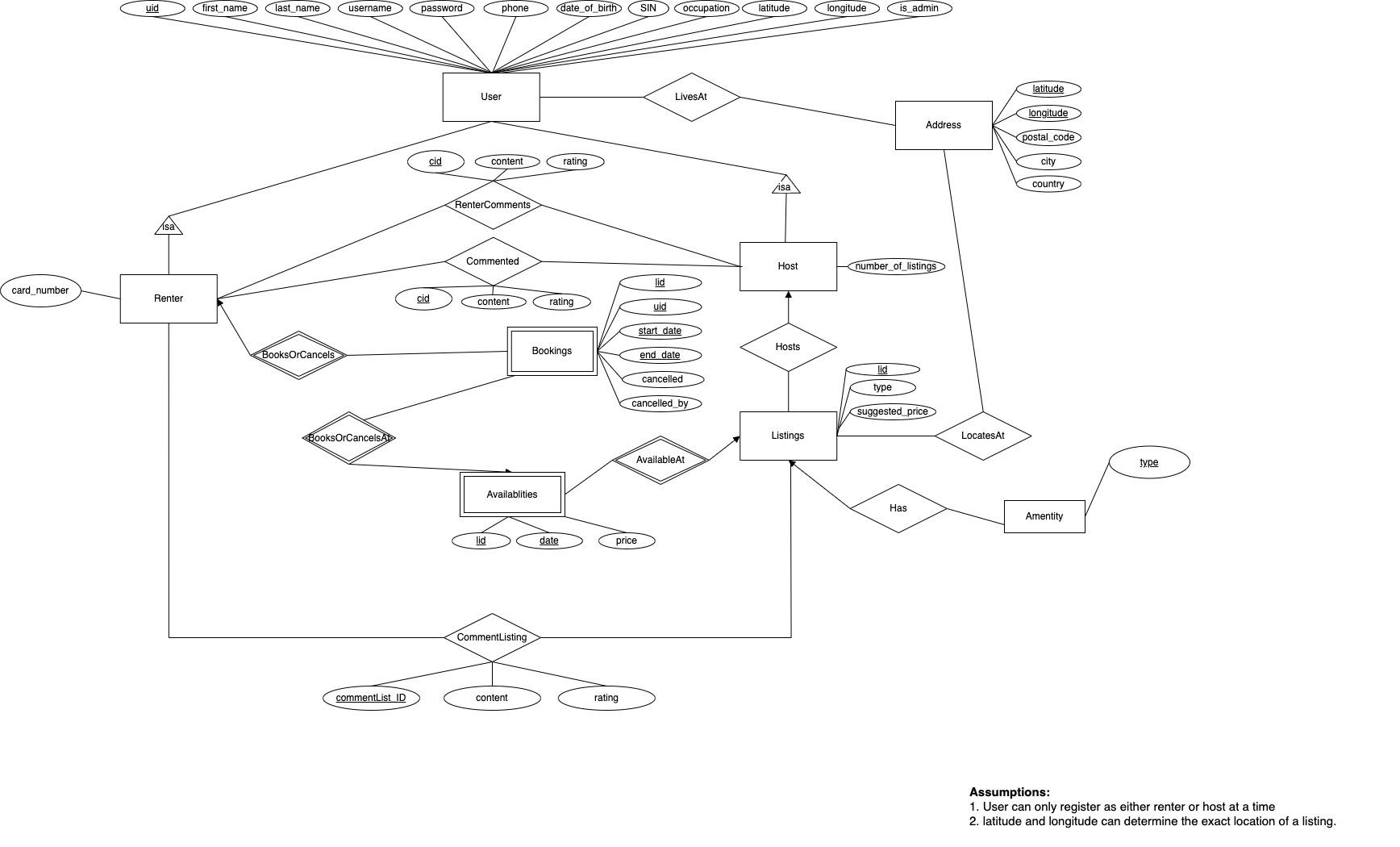
MyBnB is a mock project on Airbnb, a popular online marketplace for short-term and long-term homestays and experiences. The primary goal of this project is to provide quick and efficient solutions for hosts and renters to contact each other. Throughout the project design and implementation, we encountered several conceptual challenges, with one of the most prominent being the complex database schema design.

Initiating the project from scratch and establishing relationships between tables is a major challenge. During the developing process, we have to revise the design multiple times. For example, we initially failed to store the cancelled bookings in our schema, so while we are writing the report to print out the hosts and renters with the largest cancellations , we have to go back and revise the design. We introduced a boolean attribute “cancelled” and an integer value “cancelled\_by” indicating the cancellation status and the user responsible for it. This strategy enhanced our query writing process, and eliminated the need to create an extra table to represent the cancelled bookings.

Optimization of database queries is another challenge during the project’s creation. Our focus consistently revolved around minimizing data redundancy within our design and keeping finding efficient methods for data manipulation. By focusing on this principle, we eventually find efficient queries to retrieve and manipulate data, enhancing the overall responsiveness and performance of the platform.

**Assumptions:**

1. Users can only register as either renter or host at a time.
2. latitude and longitude can determine the exact location of a listing
3. Booking is for the whole day.

**ER diagram:**

**Schema:**

User(uid, first\_name, last\_name, username, password, phone, date\_of\_birth, SIN, occupation, is\_admin)

Address(latitude, longitude, postal\_code, city, country)

LivesAt(uid, latitude, longitude)

Renter(uid, card\_number)

Host(uid, number\_of\_listings)

RenterComments(uid1, uid2, comment\_ID, content, rating)

HostComments(uid1, uid2, comment\_ID, content, rating)

Bookings(lid, uid, start\_date, end\_date, cancelled, cancelled\_by)

BookOrCancels((lid, uid, start\_date, end\_date)

Availabilities(lid, date, price)

BookOrCancelsAt(uid, lid, date)

Listings(lid, type, suggested\_price)

Hosts(uid, lid)

AvailableAt(date, lid)

LocatesAt(lid, latitude, longitude)

CommentListing(uid, lid, content, rating)

Amenity(type, lid)

Has(lid, type)

**DDL statement:**

DROP DATABASE IF EXISTS MyBnB;

CREATE DATABASE MyBnB DEFAULT CHARACTER SET = 'utf8mb4';

USE MyBnB;

DROP TABLE IF EXISTS Addresses;

CREATE TABLE Addresses(

latitude DOUBLE,

longitude DOUBLE,

postal\_code VARCHAR(255),

city VARCHAR(255),

country VARCHAR(255),

PRIMARY KEY (latitude, longitude)

) COMMENT '';

DROP TABLE IF EXISTS Users;

CREATE TABLE Users(

uid int NOT NULL PRIMARY KEY AUTO\_INCREMENT COMMENT 'Primary Key',

first\_name VARCHAR(255),

last\_name VARCHAR(255),

username VARCHAR(255),

password VARCHAR(255),

phone VARCHAR(255),

date\_of\_birth DATE,

SIN VARCHAR(255),

occupation VARCHAR(255),

latitude DOUBLE,

longitude DOUBLE,

Foreign Key (latitude, longitude) REFERENCES Addresses(latitude, longitude) ON DELETE CASCADE,

is\_admin BOOLEAN

) COMMENT '';

DROP TABLE IF EXISTS Renters;

CREATE TABLE Renters(

uid int,

Foreign Key (uid) REFERENCES Users(uid) ON DELETE CASCADE,

card\_number VARCHAR(255),

primary key (uid)

) COMMENT '';

DROP TABLE IF EXISTS Hosts;

CREATE TABLE Hosts(

uid int,

Foreign Key (uid) REFERENCES Users(uid) ON DELETE CASCADE,

number\_of\_listings int,

primary key (uid)

) COMMENT '';

DROP TABLE IF EXISTS Listings;

CREATE TABLE Listings(

lid int NOT NULL PRIMARY KEY AUTO\_INCREMENT COMMENT 'Primary Key',

uid int,

Foreign Key (uid) REFERENCES Hosts(uid) ON DELETE CASCADE,

type VARCHAR(255),

latitude DOUBLE,

longitude DOUBLE,

suggested\_price FLOAT,

Foreign Key (latitude, longitude) REFERENCES Addresses(latitude, longitude) ON DELETE CASCADE

) COMMENT '';

DROP TABLE IF EXISTS Amentity;

CREATE TABLE Amentity(

type VARCHAR(255),

lid int,

Foreign Key (lid) REFERENCES Listings(lid) ON DELETE CASCADE,

PRIMARY KEY (type, lid)

) COMMENT '';

DROP TABLE IF EXISTS Availabilities;

CREATE TABLE Availabilities(

lid int,

Foreign Key (lid) REFERENCES Listings(lid) ON DELETE CASCADE,

Date DATE,

price FLOAT,

PRIMARY KEY (lid, DATE)

) COMMENT '';

DROP TABLE IF EXISTS Bookings;

CREATE TABLE Bookings(

lid int,

Foreign Key (lid) REFERENCES Listings(lid) ON DELETE CASCADE,

uid int,

Foreign Key (uid) REFERENCES Renters(uid) ON DELETE CASCADE,

start\_date DATE,

end\_date DATE,

cancelled BOOLEAN,

cancelled\_by int,

PRIMARY KEY (lid, uid, start\_date, end\_date)

) COMMENT '';

DROP TABLE IF EXISTS RenterComments;

CREATE TABLE RenterComments(

cid int NOT NULL PRIMARY KEY AUTO\_INCREMENT COMMENT 'Primary Key',

uid1 int,

Foreign Key (uid1) REFERENCES Renters(uid) ON DELETE CASCADE,

uid2 int,

Foreign Key (uid2) REFERENCES Hosts(uid) ON DELETE CASCADE,

content VARCHAR(255),

rating int

) COMMENT '';

DROP TABLE IF EXISTS HostComments;

CREATE TABLE HostComments(

cid int NOT NULL PRIMARY KEY AUTO\_INCREMENT COMMENT 'Primary Key',

uid1 int,

Foreign Key (uid1) REFERENCES Hosts(uid) ON DELETE CASCADE,

uid2 int,

Foreign Key (uid2) REFERENCES Renters(uid) ON DELETE CASCADE,

content VARCHAR(255),

rating int

) COMMENT '';

DROP TABLE IF EXISTS ListingComments;

CREATE TABLE ListingComments(

cid int NOT NULL PRIMARY KEY AUTO\_INCREMENT COMMENT 'Primary Key',

uid int,

Foreign Key (uid) REFERENCES Renters(uid) ON DELETE CASCADE,

lid int,

Foreign Key (lid) REFERENCES Listings(lid) ON DELETE CASCADE,

content VARCHAR(255),

rating int

) COMMENT '';

# USER MANUAL

## Start Page:

#### Enter 1 to enter MyBnB, enter 2 to EXIT.

* + 
  + 
  + 

#### Sign Up (enter 2 to sign up).

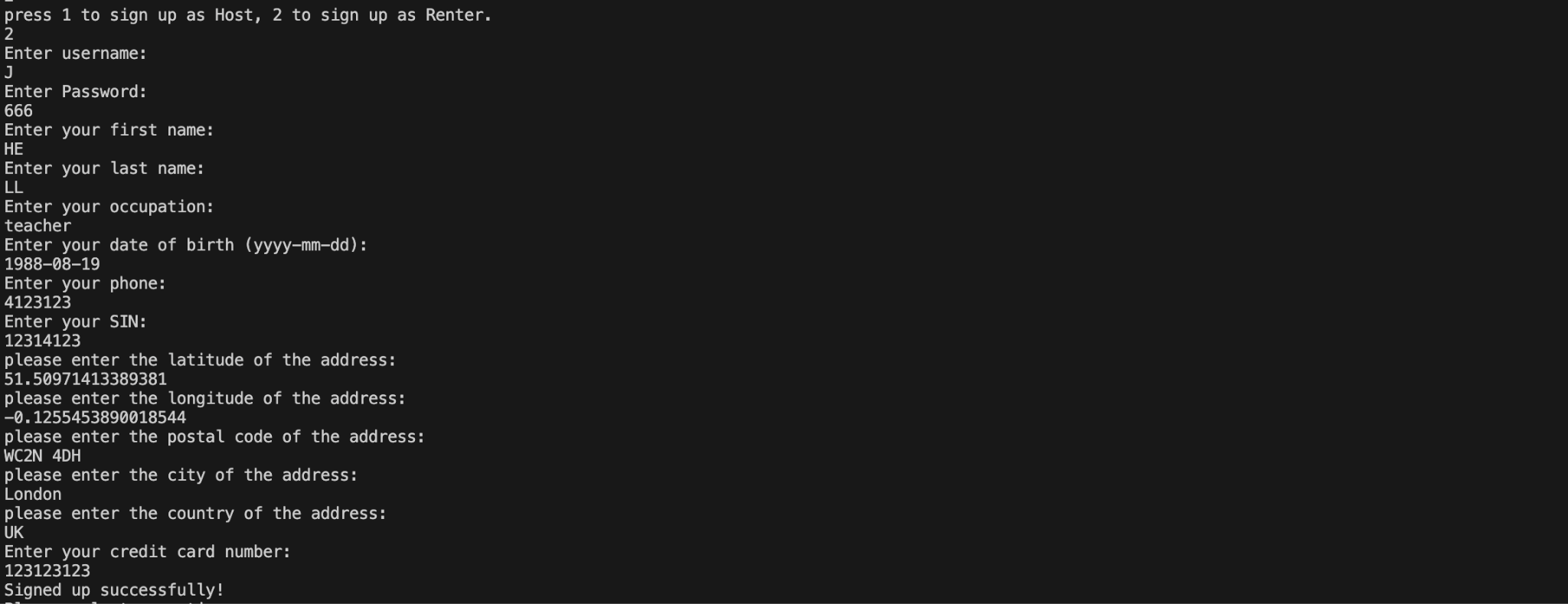
##### Host sign up

* + - According to the instructions on the screen, enter username, password, first name, last name, occupation, date of birth(in yyyy-mm-dd format), phone number, SIN number, latitude of address, longitude of address, postal code, city and country.



##### Renter sign up

* + - According to the instructions on the screen, enter username, password, first name, last name, occupation, date of birth(in yyyy-mm-dd format), phone number, SIN number, latitude of address, longitude of address, postal code, city, country, and credit card number.



#### Host Page

* + Host can choose from the following options:



* + Post a listing (press 1 and enter):
    - Post a house (press 1 and enter):
      * Then enter the latitude, longitude, postal code, city, country amenities of listing accordingly.
      * 
      * 
    - Post an Apartment (press 2 and enter):
      * Then enter the latitude, longitude, postal code, city, country amenities of listing accordingly.
      * 
      * 
    - Post a Guest house (press 3 and enter):
      * Then enter the latitude, longitude, postal code, city, country amenities of listing accordingly.
      * 
      * 
    - Post a Hotel (press 4 and enter):
      * Then enter the latitude, longitude, postal code, city, country amenities of listing accordingly.
      * 
      * 
  + View the listings the current host user own (press 2 and enter):
    - provide all the information of the host's listings, including booking, availability and comments of the listing.
    - Host can then manage the listing by the following options:
      * Update listing (press 1 and enter):
        + 
      * Update availabilities (press 2 and enter):
        + 
      * Update bookings (press 3 and enter):
        + 
      * Delete listing (press 4 and enter):
        + 
      * Back to previous page (press 0 and enter):
        + 
  + Comment on renters (press 3 and enter):
    - Hosts can comment on renters from previous bookings by entering the associated listing id and renter id, then give their comments and ratings for the renter.
    - 
  + View user account information (press 4 and enter):
    - 
  + Update user account (press 5 and enter):
    - users can update their account information by re-entering all the informations
    - 
  + Delete user account (press 6 and enter):
    - After the account is successfully deleted, a message will appear on screen.
    - 
  + Logout (press 7 and enter):
    - 

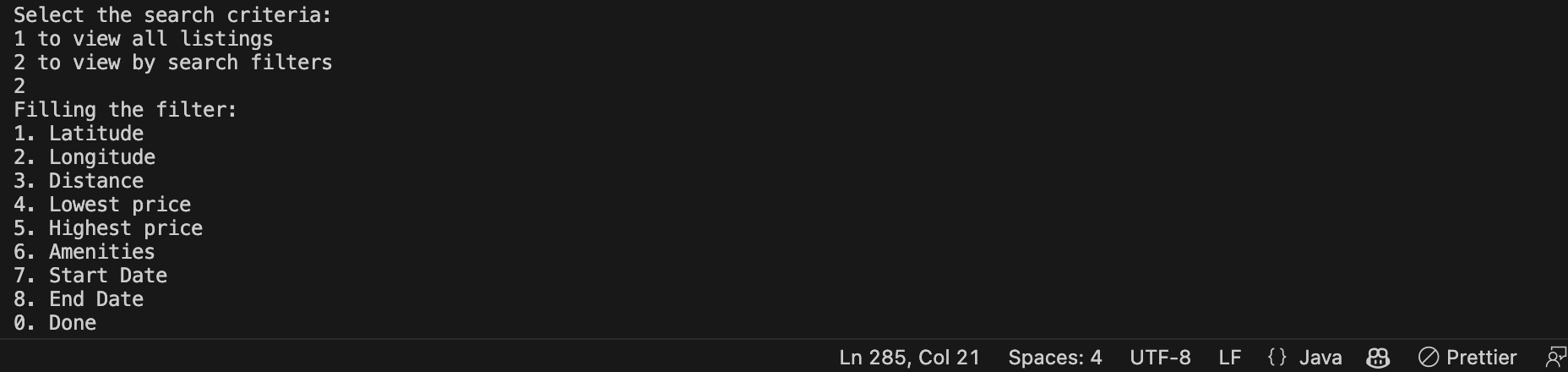
#### Renter Page

* + Renter can choose from the following options:
    - 
  + To search listings (press and enter 1). Select the search criteria:
    - * 1 to view all listings

Ranked by price:

* + - * + 1. Ascending. 2. Descending. 3. No ranking
      * 2 to view by search filters

Filling the filter:



* + To book listings (press and enter 2):
    - Renters can enter the listing id:

Enter the start date in yyyy-mm-dd:

Enter the end date in yyyy-mm-dd:



* + To cancel a booking (press and enter 3):
    - Enter the listing id:

Enter the start date in yyyy-mm-dd:

Enter the end date in yyyy-mm-dd:



* + To comment (press and enter 4):
    - By entering the listing id, renters can comment on the listing and corresponding host they have rented from.



* + To view bookings in the past (press and enter 5):



* + To view renter account information (press and enter 6):



* + To update renter account information (press and enter 7):
    - 
  + To delete the current renter account(press and enter 8):



* + To log out the current renter account(press and enter 9):



#### Admin Page

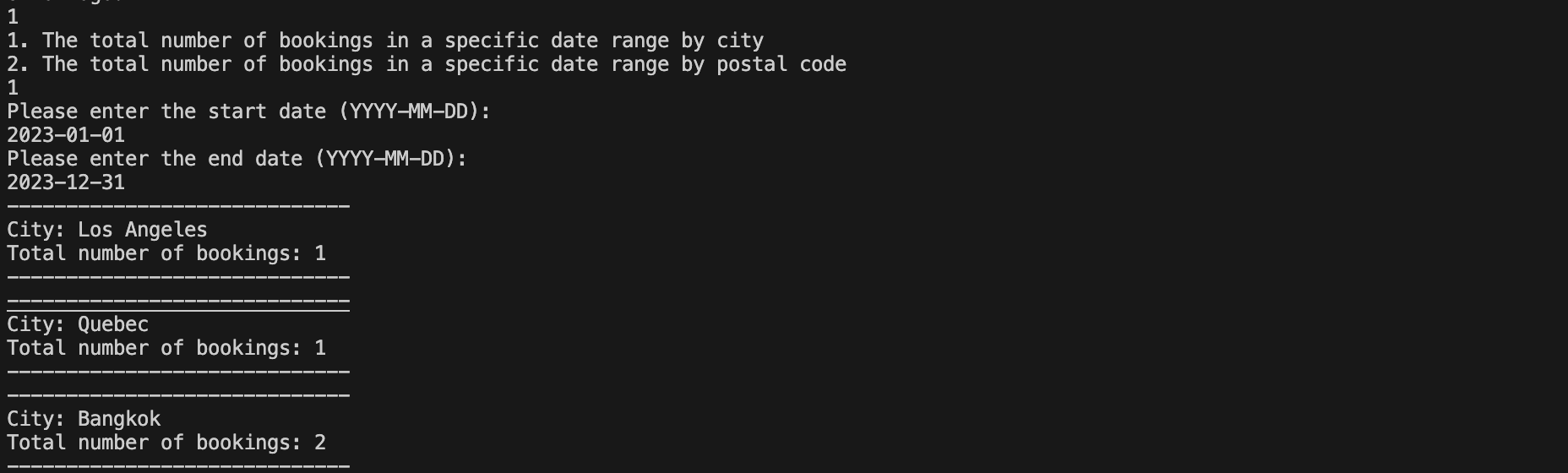
Please select an option:

1 to view total number of bookings

1. The total number of bookings in a specific date range by city

Please enter the start date (YYYY-MM-DD):

Please enter the end date (YYYY-MM-DD):



2. The total number of bookings in a specific date range by postal code

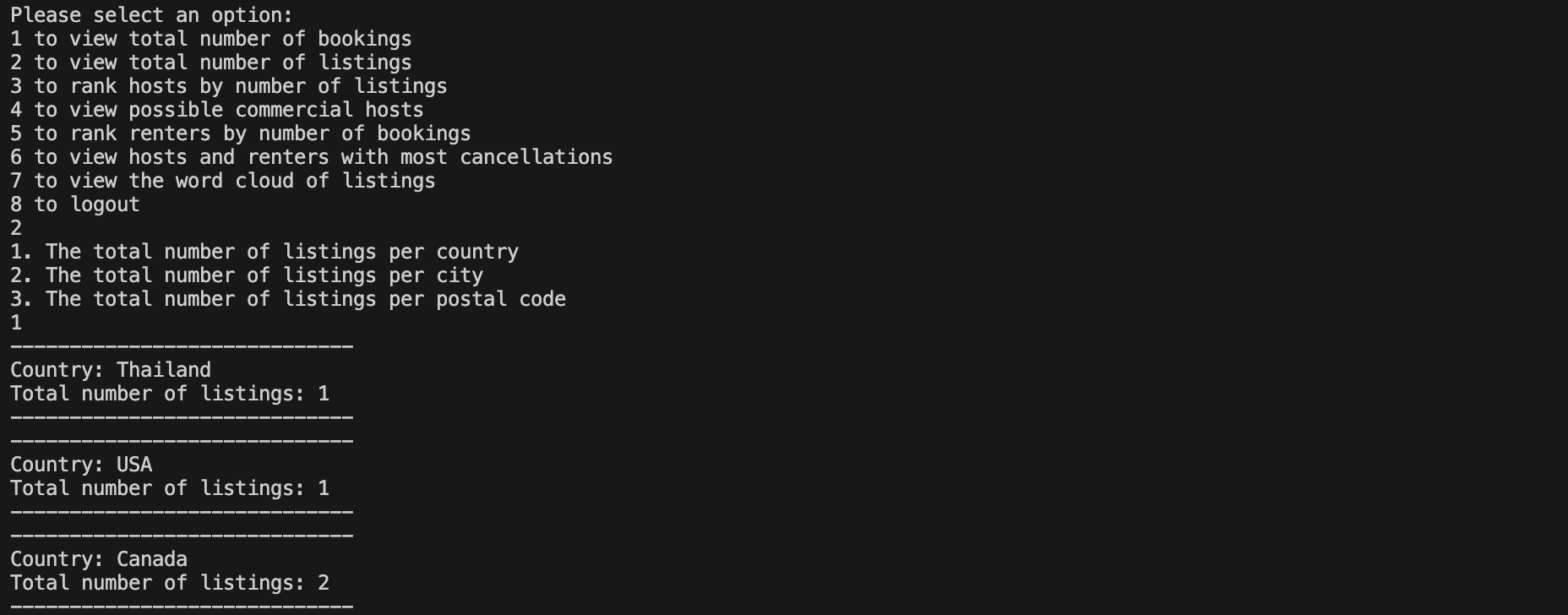
Please enter the start date (YYYY-MM-DD):

Please enter the end date (YYYY-MM-DD):

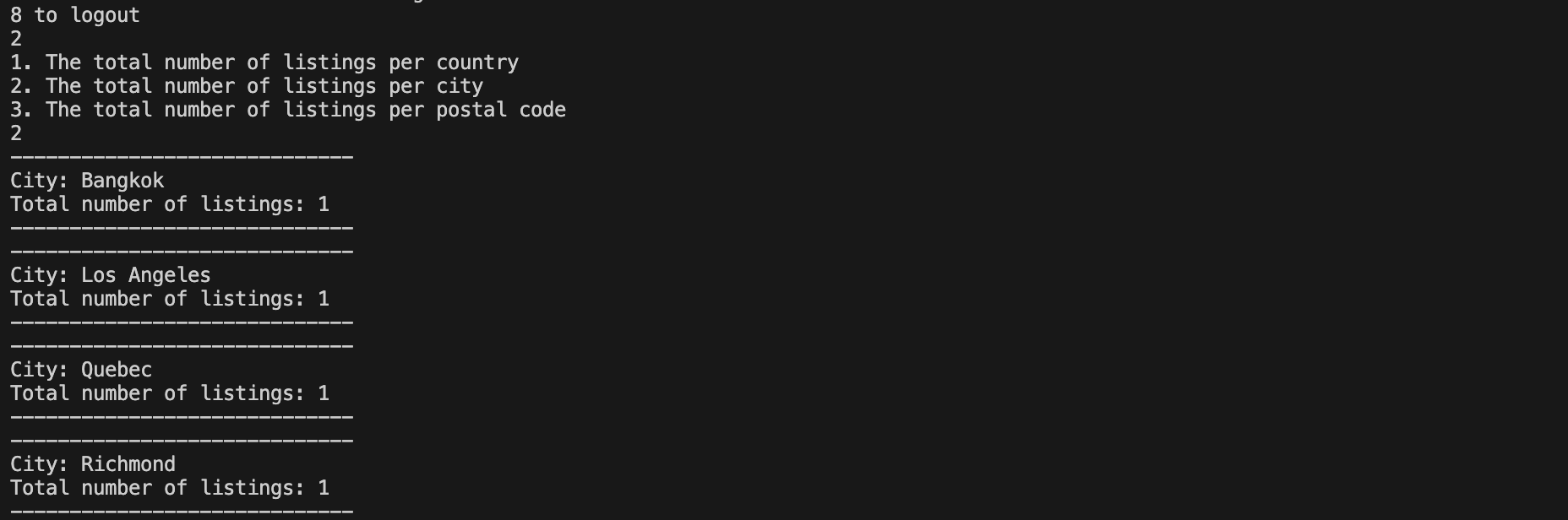


2 to view total number of listings

​​1. The total number of listings per country



2. The total number of listings per city

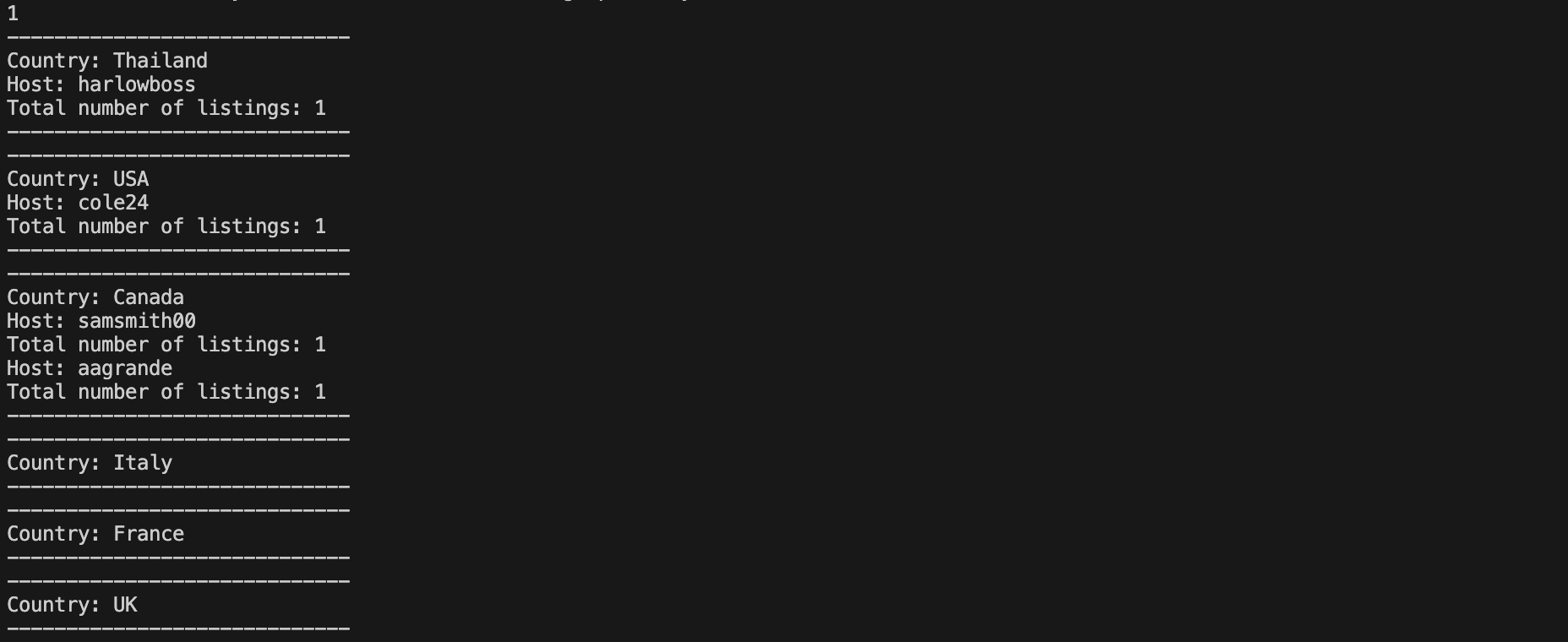


3. The total number of listings per postal code



3 to rank hosts by number of listings

1. Rank the hosts by the total number of listings per country

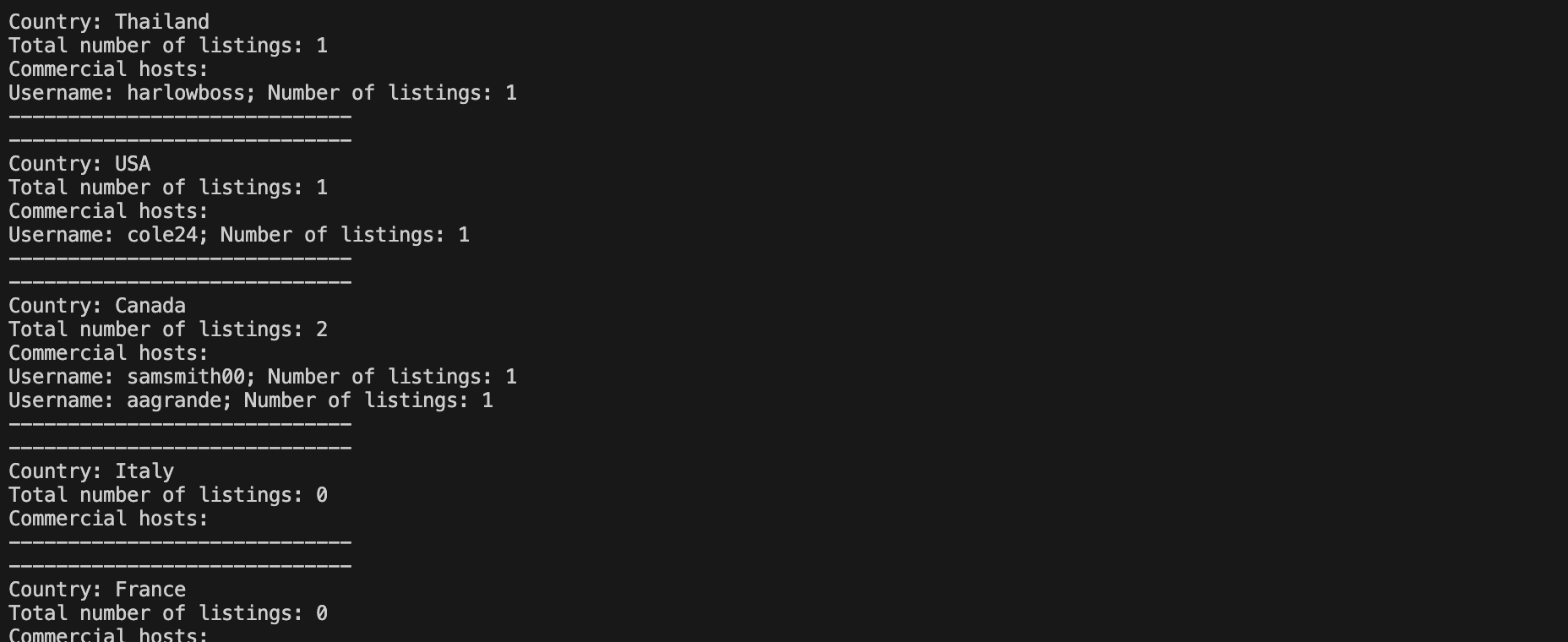


2. Rank the hosts by the total number of listings per city



4 to view possible commercial hosts

1. View the commercial hosts by the total number of listings per country



2. View the commercial hosts by the total number of listings per city

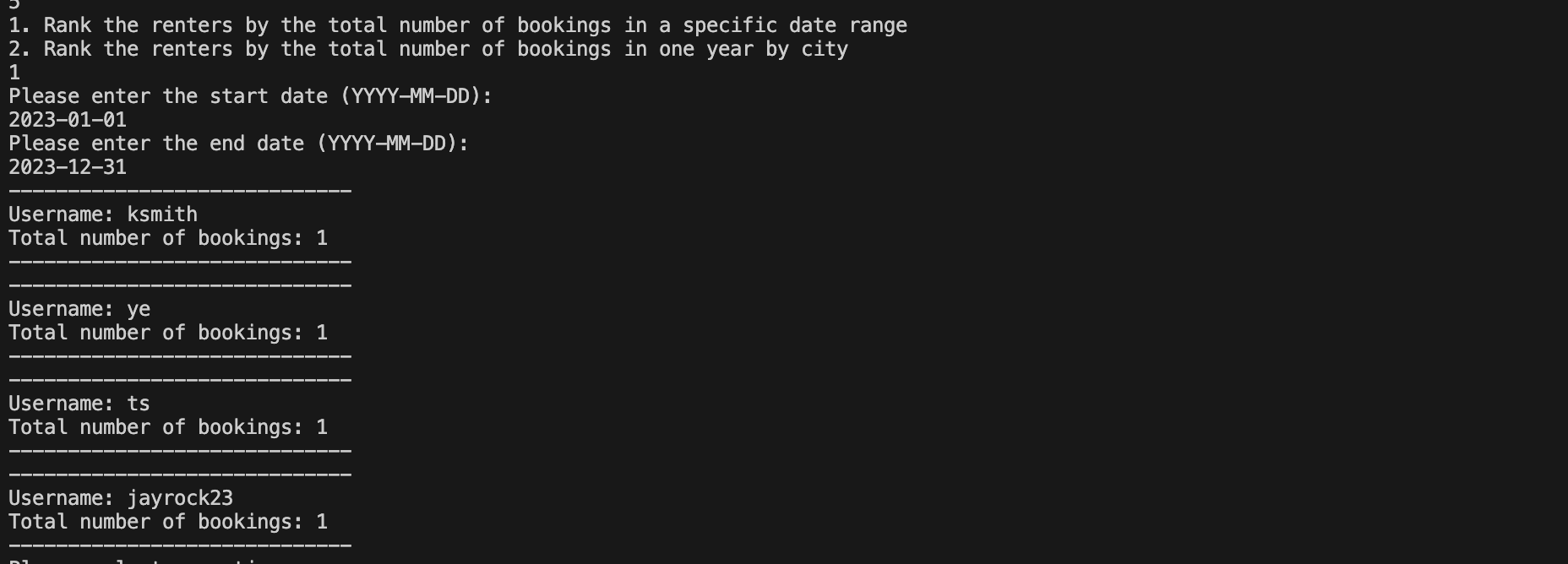


5 to rank renters by number of bookings

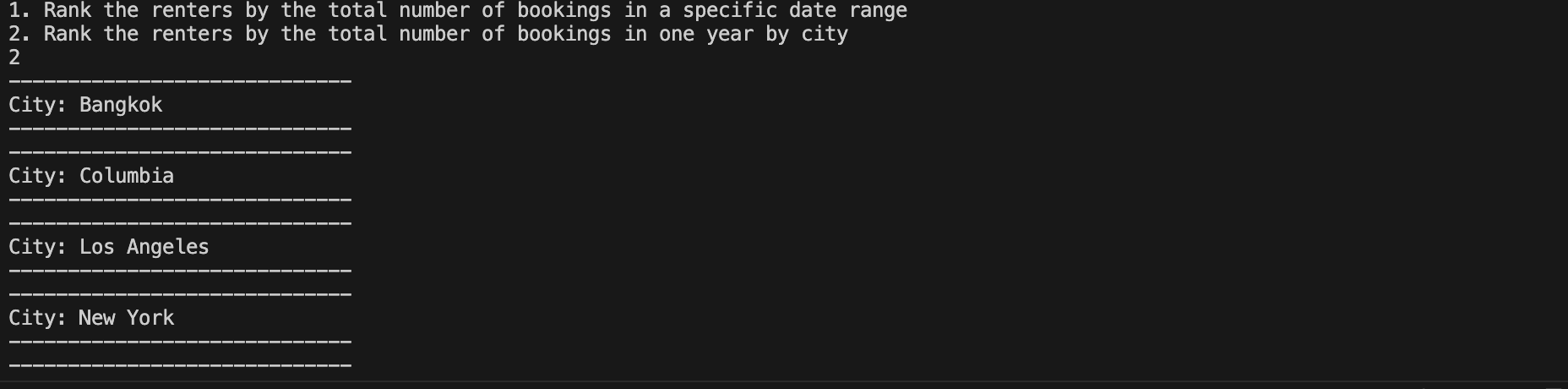
1. Rank the renters by the total number of bookings in a specific date range

Please enter the start date (YYYY-MM-DD)：

Please enter the end date (YYYY-MM-DD):



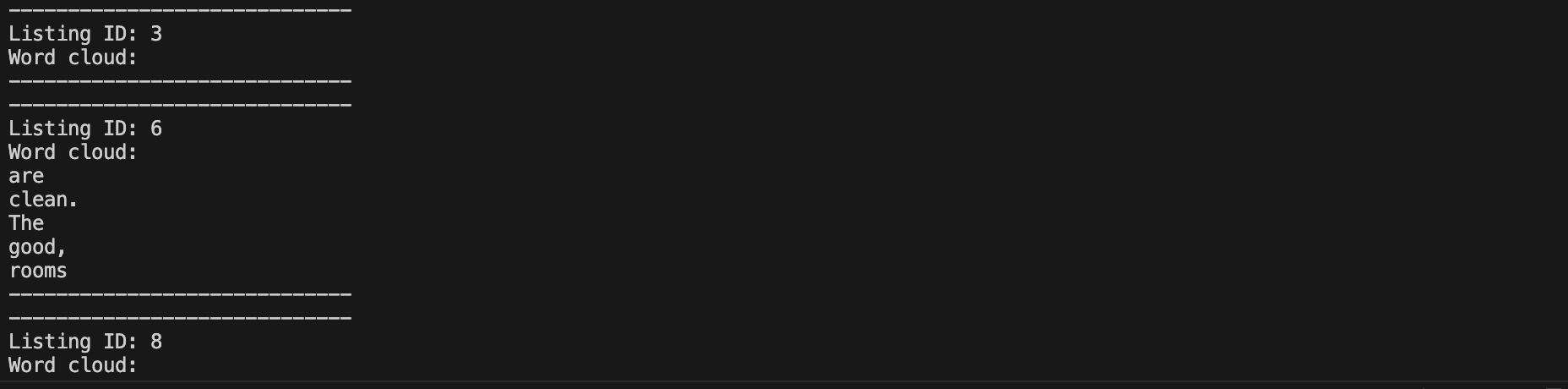
2. Rank the renters by the total number of bookings in one year by city



6 to view hosts and renters with most cancellations



7 to view the word cloud of listings



8 to logout

**System limitations:**

Although the program runs all necessary queries, there are still some limitations that need to be fixed for future reference. One limitation is that the current system stores availability of a listing by date, and once the data gets large, the run time will certainly slow down. So if there is more time, I believe we can store data in a better and proper way. Another limitation would be the lack of good user interface and lack of pictures for each listing, because of the time limit, users can only use command prompt to type their input and view output, they cannot view the pictures of listings. The project will be improved a lot with a decent user interface.