TESTING

Link to the web application:

https://whispering-reaches-99462.herokuapp.com/

The required user roles along with their credentials are as follows:

Type	Username	Password
Admin	admin	admin
Host	alice	alice
Guest	bob	bob

Link to the data model:

Hostname:	cs5200-spring2020-shubhamkurkure.cqj4ratuso1k.us-east-1.rds.amazonaws.com
Username:	shubhamkurkure
Password:	Friends123
Port:	3306

Testing based on given 11 requirements is as follows:

1. **User related to another user**
   URL: /reviews (must be logged in as either host or guest)
   The guest user after making a trip can provide a review for the trip. At the same time, a host can also give a review about a guest that attended a trip at the host's property. These reviews of a given user can be seen under the route "/reviews"

2. **User searches domain objects**
   URL: /search (authentication not required)
   This route provides functionality to search for properties in our application.

3. **User view details of domains**
   URL: /properties (authentication required either as guest or host)
   This route can be visited as a host user (Alice). This route lists all the properties owned and hosted by the host user.

4. **User views all domains objects related to user**
   URL: /properties (for host user Alice)
   /trips (for guest user Bob)
   /properties route provides a list of all properties owned by a host user and /trips route gives a list of all trips booked by the guest user.

5. **A user views all other users related to the user**
   URL: /profile/{profileId}
   Example: /profile/4 (for user bob with profile id = 4)
   This route can be reached in many ways throughout the application where one domain object points to the creator/owner of that domain object.
   This can be seen when you search a property by city, like Boston, now when you click any of the property listed in the search result, you will be taken to the property details page and here you can see who the owner/hostname is. Clicking on the owner's name will take you to the owner's profile page.

6. **A user-related to a domain object**
   URL: /create/trip and /book/property/{propertyId}
   A guest user can click on create a trip button at the top and search for a property suitable for his trip and then click the book now button to book the trip by filling in the details.

7. **Domain object to another domain object**
   URL: /property/{propertyId} (on logging in as host)
   /create/trip (on logging in as guest)
   A property can have multiple trips hosted on it. The list of trips hosted can be seen by logging in as the host and going to the details page of any property. This can be done by clicking on any of the properties listed. Also, you can create a trip under a property by logging in as a guest.

8. **Admin creates a user**
   URL: /userManager
   Log in as admin and go to the “User management” page by clicking the button on the top bar. Here you should be able to create a user.

9. **Admin lists user**
   URL: /userManager
   Log in as admin and go to the “User management” page by clicking the button on the top bar. Here you should be able to search the user by id. We haven’t listed the users, because there is a lot of users injected from an external data source and was causing an overload on the frontend to render and making the site unresponsive for a while.

10. **Admin edits a user**
    URL: /userManager
    Log in as admin and go to the “User management” page by clicking the button on the top bar. Here you should be able to edit a user.

11. **Admin removes a user**
    URL: /userManager
    Log in as admin and go to the “User management” page by clicking the button on the top bar. Here you should be able to remove a user.