San Jose State University

Fall 2019

Group -14

OpenHome

SUBMITTED BY -

Akshay Bajaj - 012705186

Rajesh Thummala - 013829179

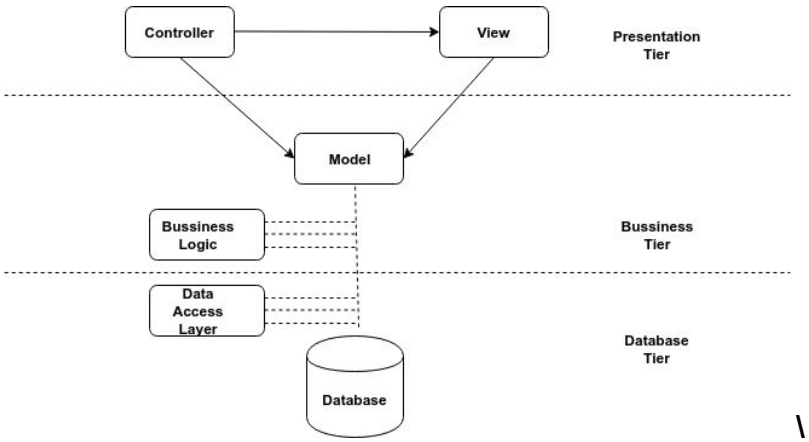
Nithin Krishna Gollanapally - 013820599

Nidhi Tattur Aravinda Kumar -013845494

SUBMITTED TO -

Prof. Charles Zhang

1. **Motivation :** Online hotel booking management is a growing field which is being used by customers widely. You can book hotels online before actually going there, ensuring your stay is planned and not have to move along with your luggage from place to place. On the other hand as a business you can post your place online increasing the domain for the customers from a region to anywhere in the world. This is very much needed application and is used widely now a days and has great scope for good response. As time is moving ahead we have introduced latest features that may override current applications in the market.
2. **Introduction :** OpenHome booking management will be used by hosts for posting the available place and by guests to reserve a place. It will enhance their experience as we have introduced latest features such as google map integration, google/facebook authentication, yearly billing reports, smart and efficient search. Guests have an option to book a place, modify, checkin/checkout, cancel the bookings and add payments. Hosts can modify their property, cancel bookings.
3. **High Level Design :** The high level design of our application is as shown in below diagram



\

Figure: Architecture Diagram

Three Tier Architecture with MVC:

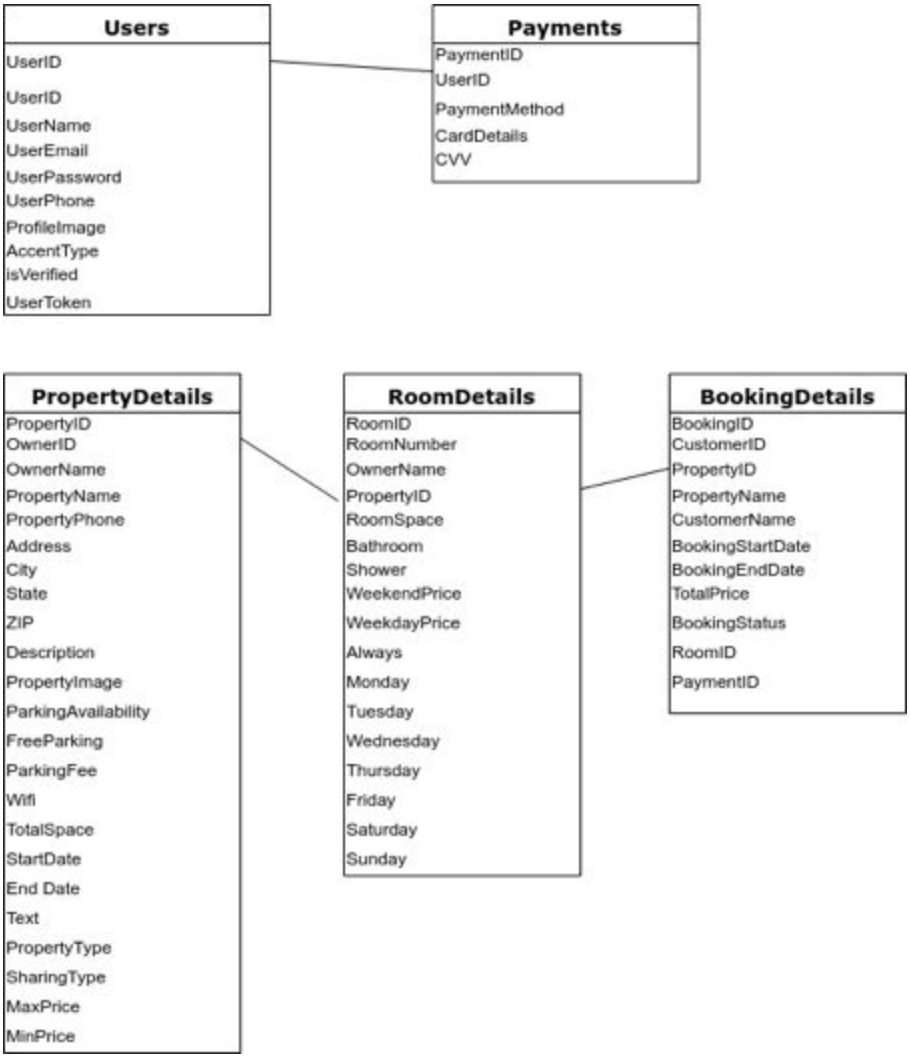
This project will be based on the MVC pattern where M stands for Model, V stand for View and C stands for Controller. The three tiers are :

- Presentation tier : User interaction will done at this layer
- Business tier : It will have all business logic implemented
- Data Tier : It will be storing application related data into database

The presentation layer will consist of view and controller. Model will exist in business tierand will interact with data layer. This three architecture makes the application layers loosely coupled. This will increase the modularity of the code and changes can be made independently from each other. This application is also scalable in design.

4. **Component Level Design:**

The entity relationship diagram of our application is as followed:



User, Payments, PropertyDetails, BookingDetails and RoomDetails are the entities that we have considered for our project. User entity has user related attributes such as username , email, password. Payments will store all the payment related information of

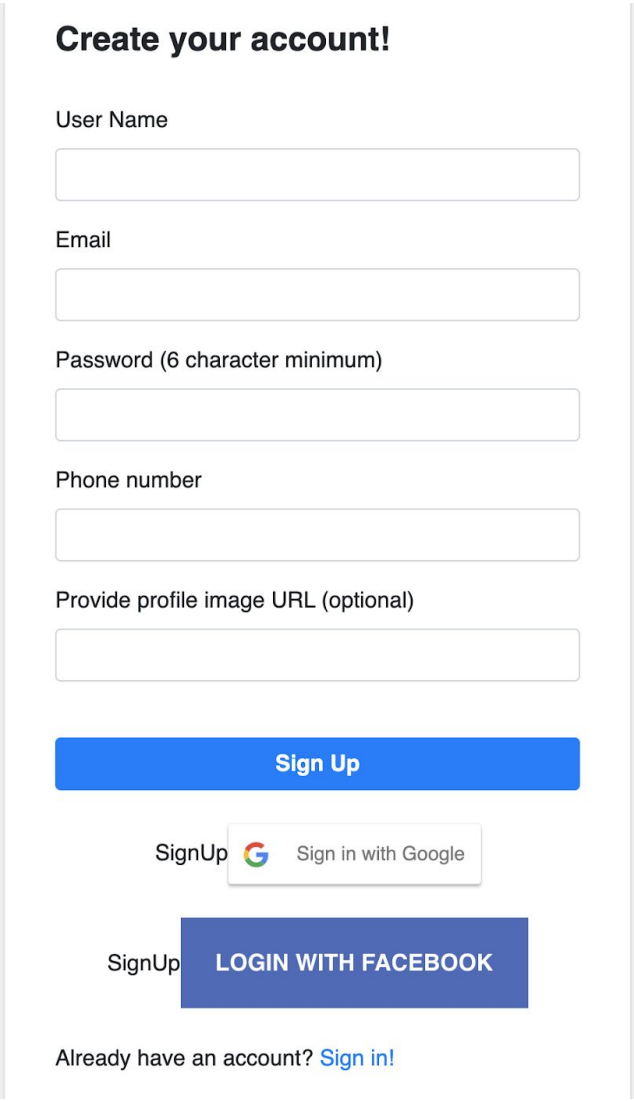
the user. Property Details will have all the property related data such as property name, address, location etc. Room Details has room related data such as type of room and various amenities related to it. Booking Details will have all the booking related data that users have booked such duration of booking, total price, room etc.

5. Platforms and Technologies Used

Frontend : REACTJS
Cloud : AWS
Backend : SPRING BOOT
Database : MYSQL

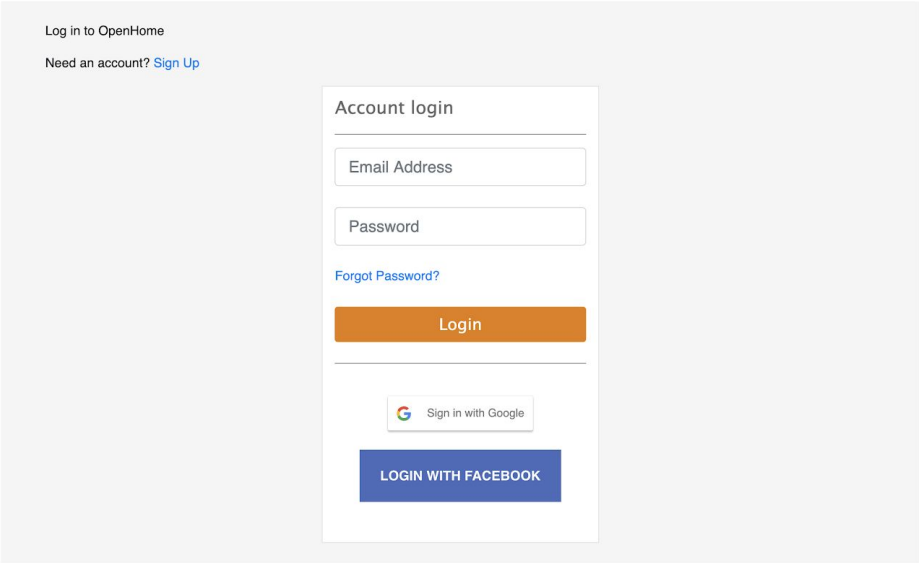
6. Features and Screenshots

SignUp

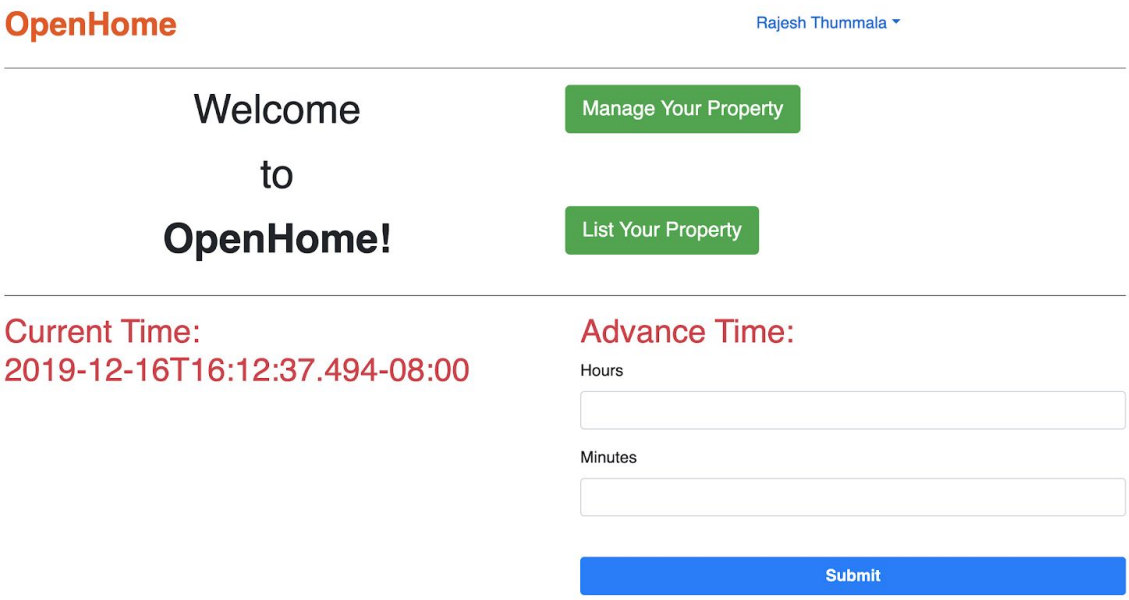


Login

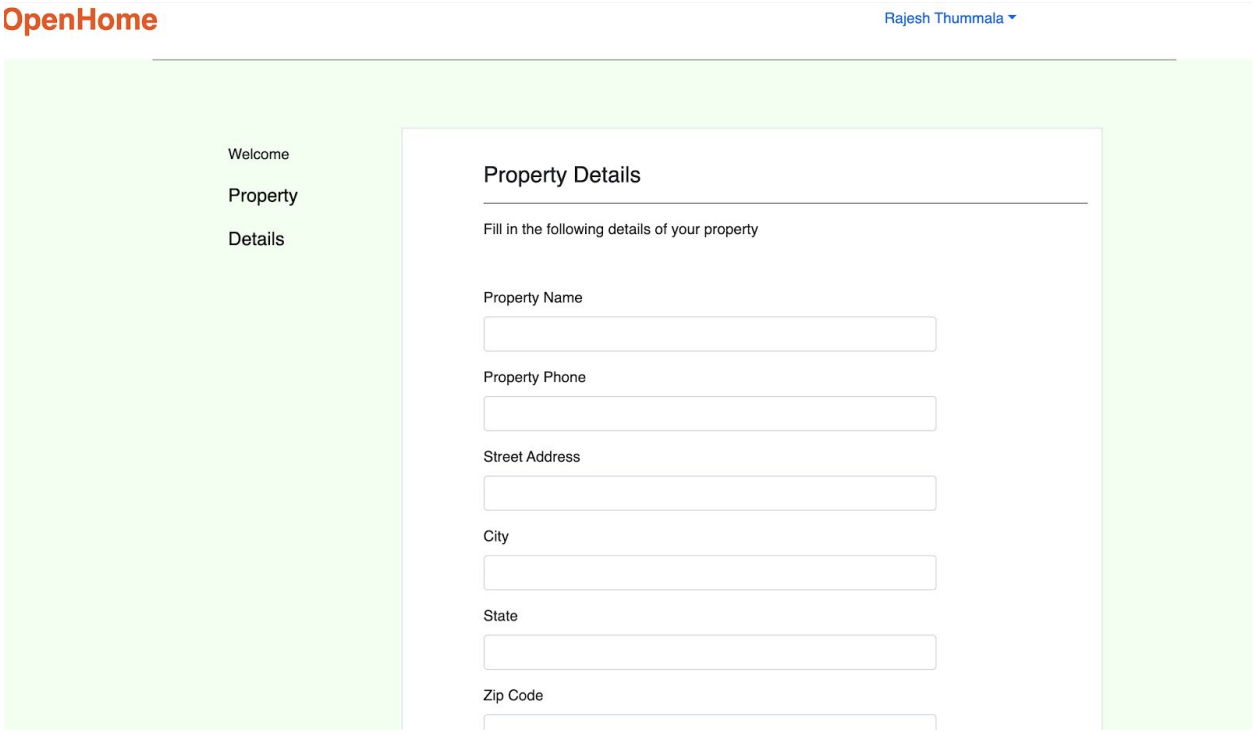
OpenHome



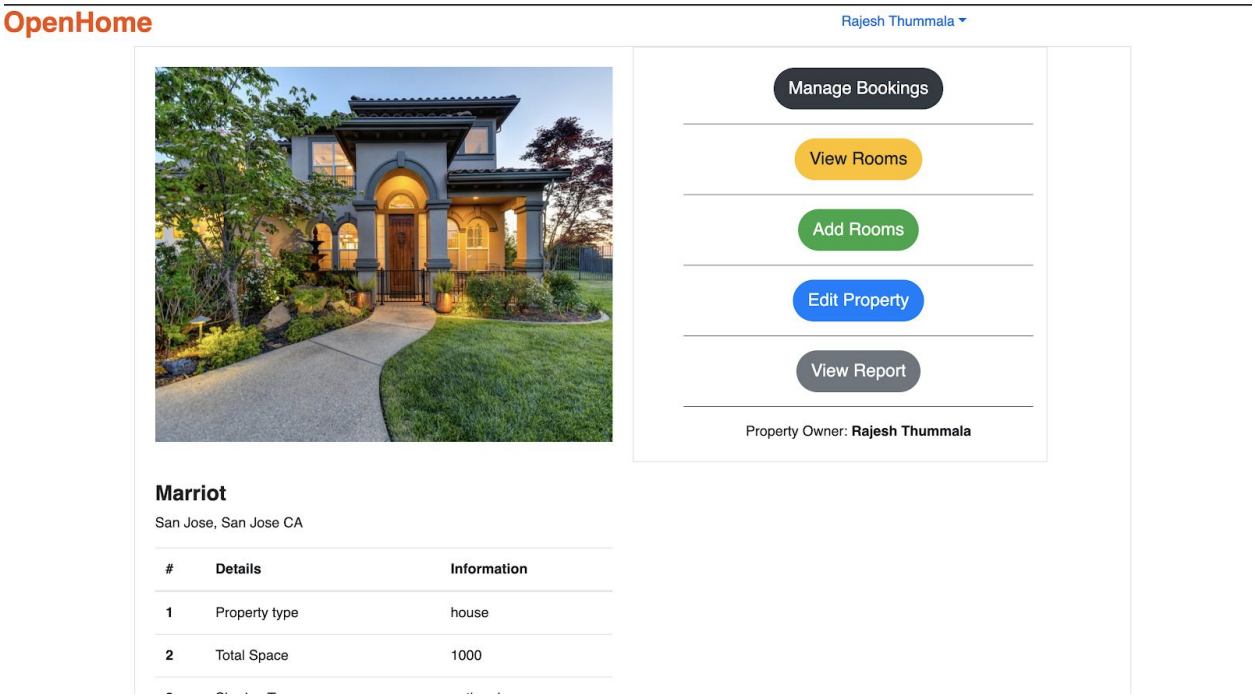
Host Home Page



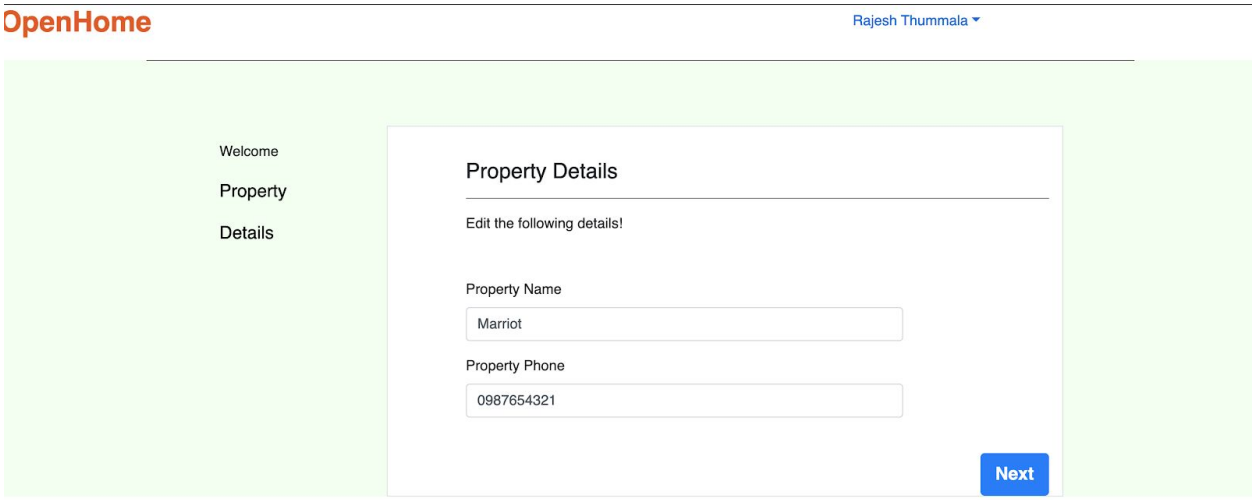
Add Property



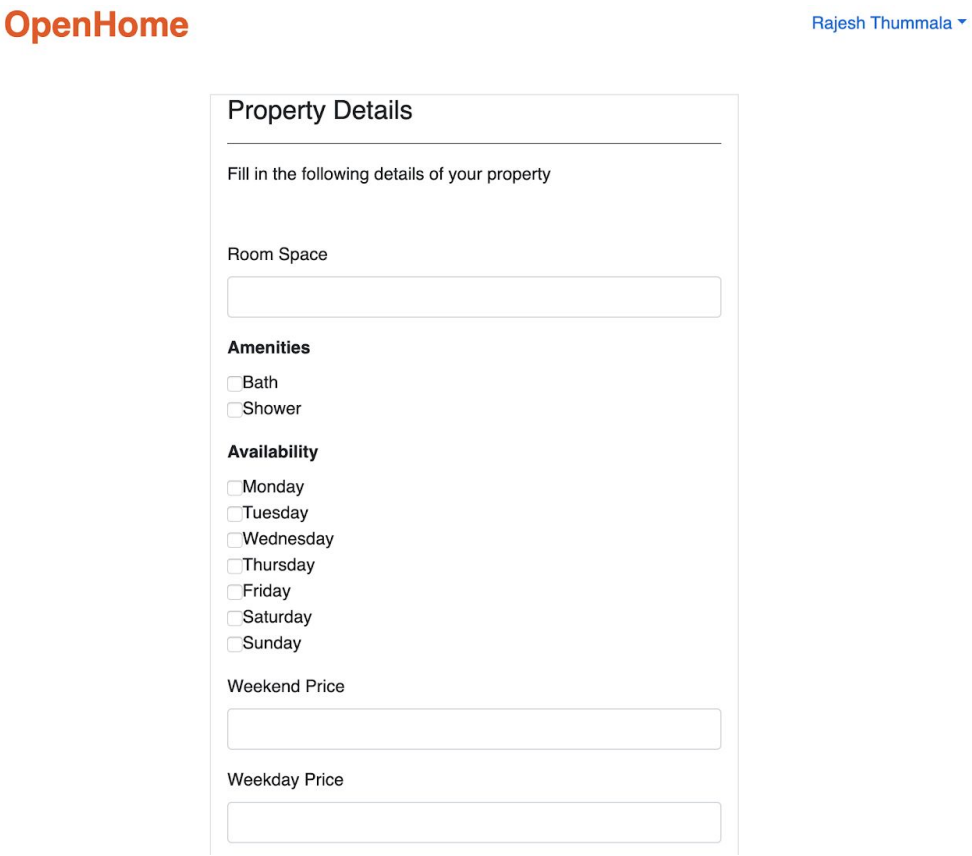
View Property Details



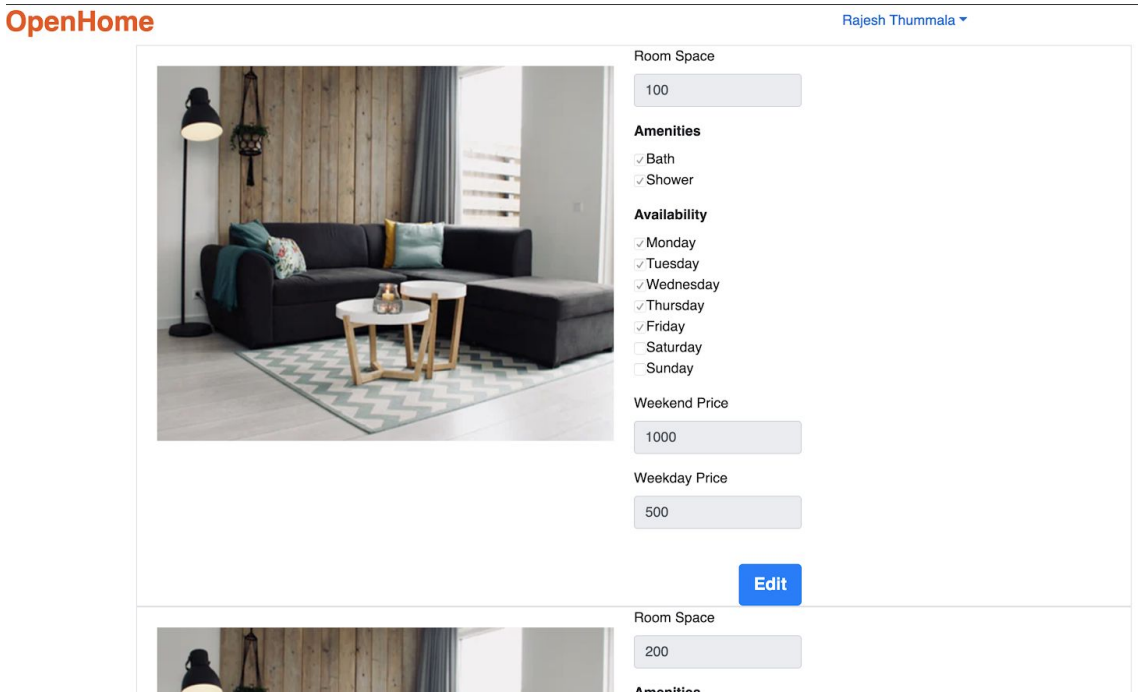
Edit Property Details



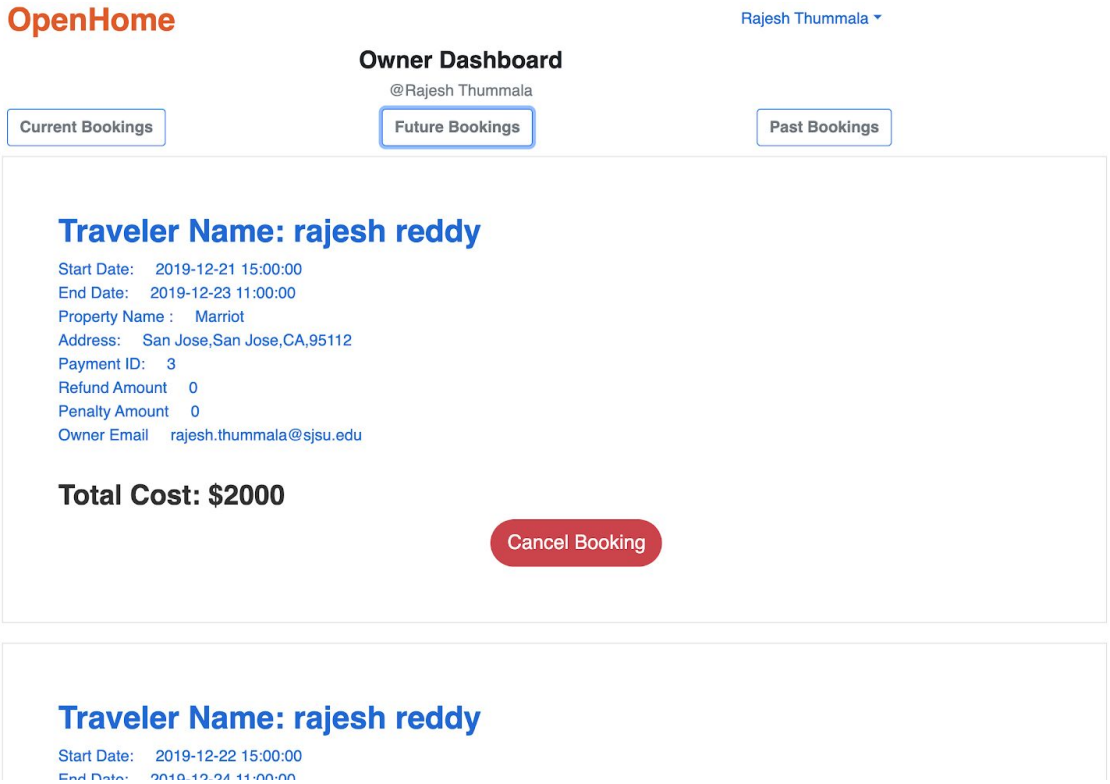
Add Room in a Property



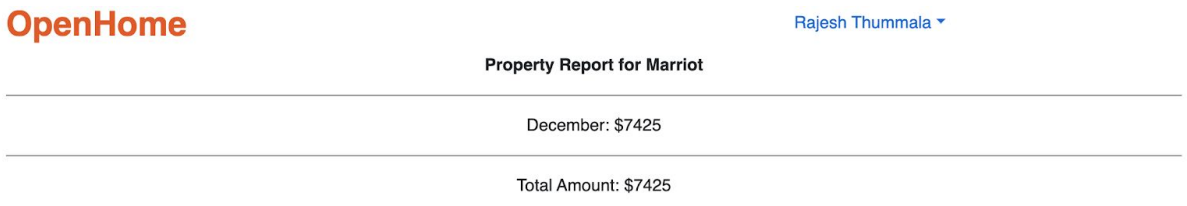
View and Edit room details



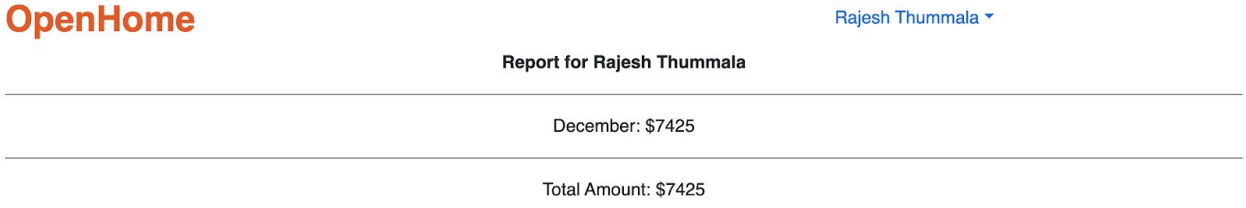
Manage Current, Past and Future Bookings



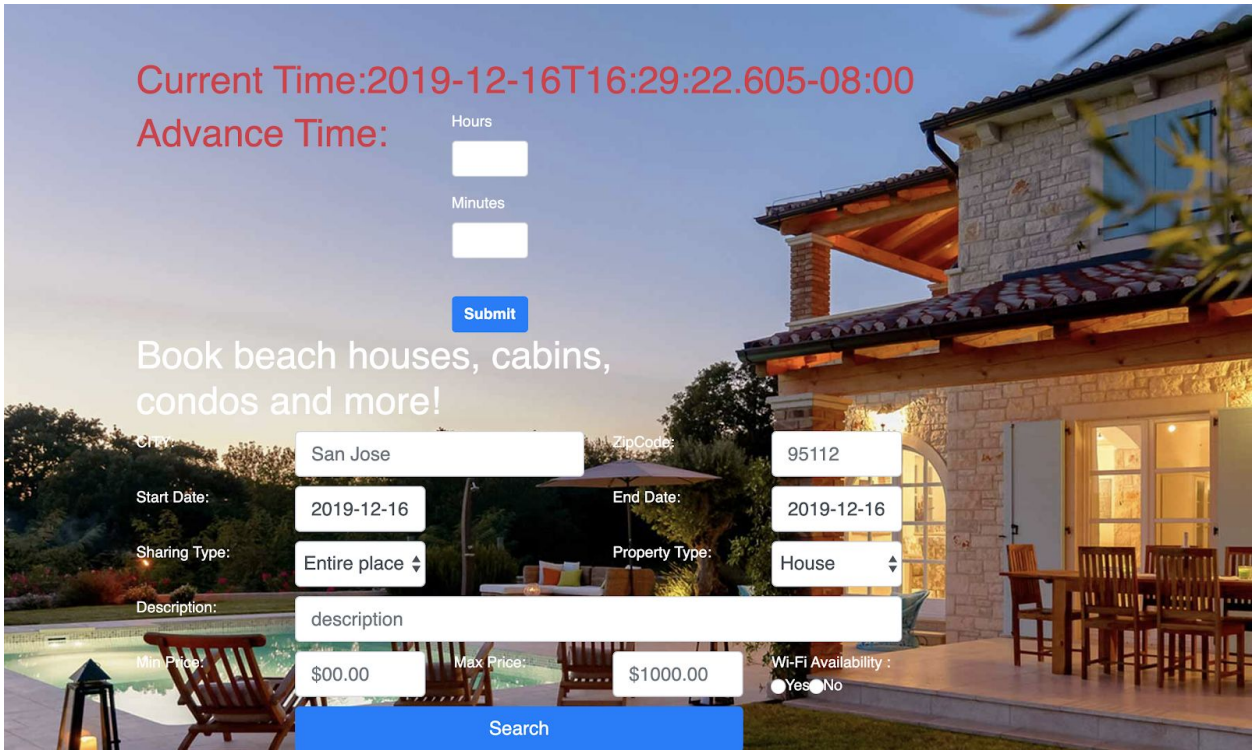
View Property Monthly wise Report



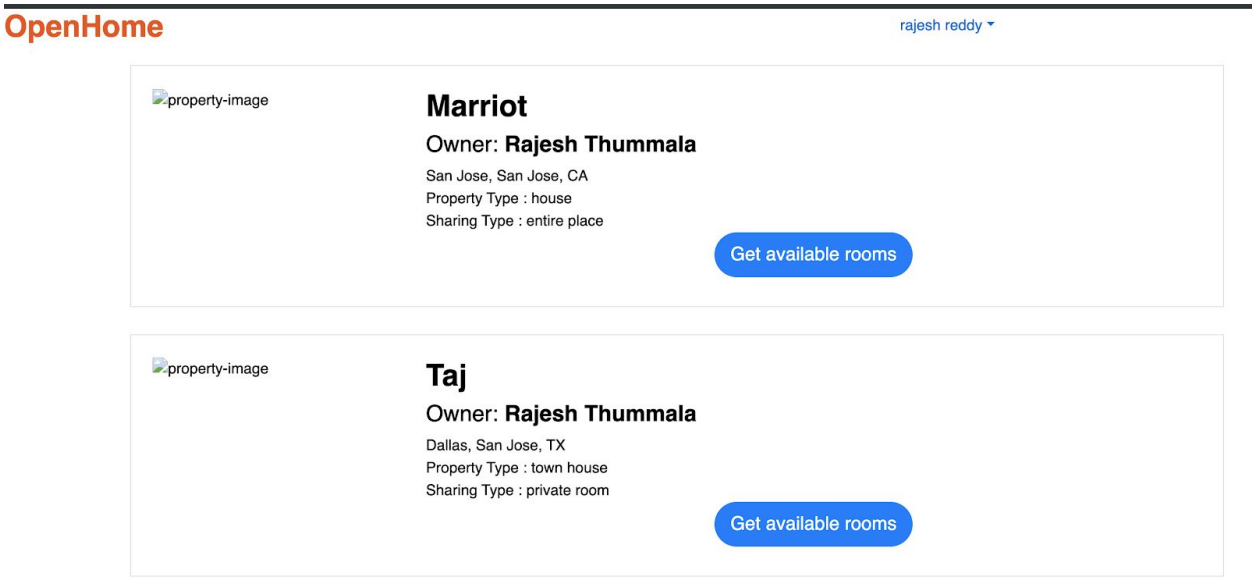
View Host Complete Report



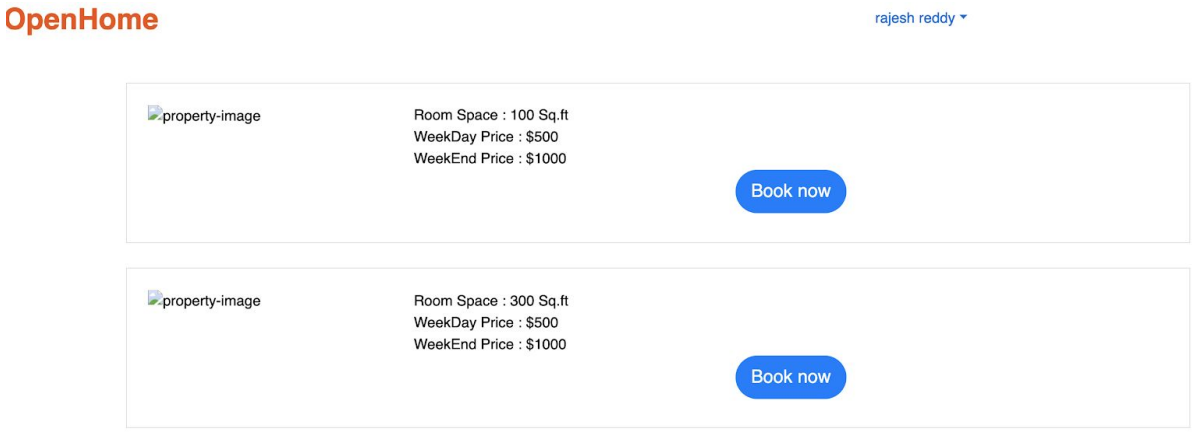
Guest Search Page



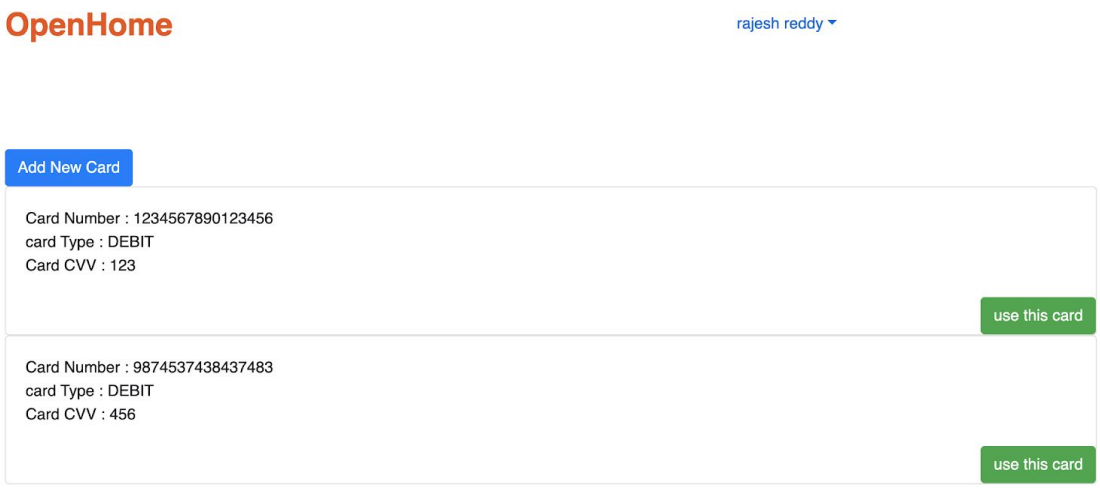
Guest Property Search Results



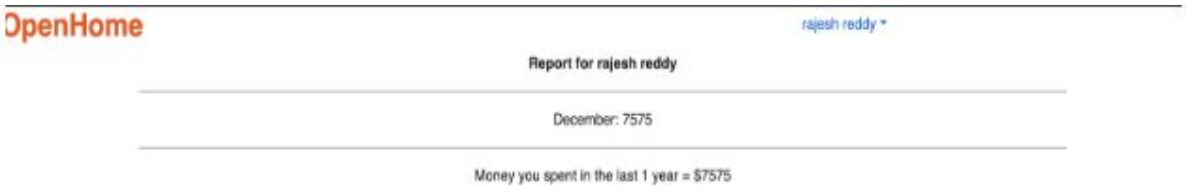
Available Room Details



Add new card / payment details



Guest Monthly wise report



7. Test Results

Following are the test cases which were used while functional testing. All the validation are tested which are shown in section 6.

Note : Test cases and Test plan are added as separate files and submitted over canvas

Test Case ID	Signup
Test Case Description	User can create account using several details using User Name, Email, Password, Phone Number, Image
Steps	Enter valid username, email, password, phone number and image(optional)
Expected Output	User will Sign up successfully
Actual Output	User Signed up successfully

Test Case ID	Login
Test Case Description	Login functionality
Steps	Enter valid email address in the text and valid password, User will be able to login successfully
Expected Output	User will Log in successfully
Actual Output	User Logged in successfully

Test Case ID	Login via facebook and google
Test Case Description	User can login using their facebook and google ids
Steps	Click on facebook/login button to log using the facebook/google account
Expected Output	User will Log in successfully
Actual Output	User Logged in successfully

Test Case ID	Login via facebook and google
Test Case Description	User can login using their facebook and google ids
Steps	Click on facebook/login button to log using the facebook/google account
Expected Output	User will Log in successfully
Actual Output	User Logged in successfully

Test Case ID	Advance Time
Test Case Description	This will advance time used used for testing purpose
Steps	Enters hours and minutes you want to add to the current time and click on submit

Expected Output	Time should advance by mentioned arguments
Actual Output	Time advances by mentioned hours and minutes

Test Case ID	Automatic booking cancellation after time advancement and cron job execution
Test Case Description	The Booking will be cancelled after execution of time advancement and cron jobs
Steps	Enters hours and minutes you want to add to the current time and click on submit.
Expected Output	Time should advance by mentioned arguments and User should see the current booking in past booking.
Actual Output	Time advances by mentioned hours and minutes and users views the current booking in past booking after cancellation

Test Case ID	Manage Property for Host
Test Case Description	Host can manage property by updating property name,

	phone number, address, city, state and zip code
Steps	Enter the updated data in the field which you want to update and information will be updated successfully
Expected Output	Information should be updated successfully.
Actual Output	Information will be updated successfully.

Test Case ID	Add room to property by host
Test Case Description	User can add room to the property.
Steps	Add room space, amenities, availability, weekday price and weekend price
Expected Output	Host should be able to add room to the property
Actual Output	Host added room to the property successfully

Test Case ID	View and Edit room details by host
Test Case Description	Host can edit and view the room details

Steps	After login click on edit property and select property. Click on room to view/edit the room
Expected Output	Host should be able to edit/view the room
Actual Output	Host was able to edit/view successfully.

Test Case ID	Manage bookings
Test Case Description	Host can manage past/future/current bookings
Steps	Click on Manage Bookings then click on current bookings/future bookings to update those bookings and enter the updated data
Expected Output	Host will be able to edit current booking/future booking.
Actual Output	Host updated current bookings/future bookings successfully.

Test Case ID	View Monthly report
Test Case Description	Host can view monthly report of the booking of the selected property

Steps	Click on property and then click on view monthly report
Expected Output	Host should be able to view monthly report successfully
Actual Output	Host was able to view monthly report successfully

Test Case ID	View the rooms by guest
Test Case Description	Guests can view the available rooms by certain filters such as city name, zip code, start date, end date, property type, sharing type price, WIFI, parking etc
Steps	Guest has to login first and then enter the required search related info
Expected Output	Guest should be able to view rooms on basis of search criteria
Actual Output	Guest was able to view rooms on basis of search criteria successfully.

Test Case ID	Select available rooms for guest
Test Case Description	Guests can select available rooms as displayed on screen
Steps	Click on property and select the room available for that property
Expected Output	Guest should be able to add room to their selection
Actual Output	Guest was able to select room successfully

Test Case ID	Pay for the current booking for guest
Test Case Description	Guests can pay for their selected room using new/previous card
Steps	Select the room that you want to book, then use your saved card or enter the details for the new card
Expected Output	Guests should be able to pay using saved/new card
Actual Output	Guest was able to pay using saved/new card successfully

Test Case ID	Guest can view monthly report
Test Case Description	Guest can view monthly report for their bookings
Steps	After login click on view monthly report
Expected Output	Guest should be able to view monthly report
Actual Output	Guest is able to view monthly report successfully

8. **Lessons Learned**

This project gave insights to many technologies and concepts that we have not used before such as Spring AOP, Spring Boot and Hibernate. It also gave us end to end developer experience in which we worked on all stacks that is Frontend, Backend and Database and deployment. We have implemented AOP before and after user logs in or sign up. It also gave us a learning experience on how to work in MVC framework. We also learned how to work as a team and complete our task accordingly.

9. **Future Scope**

● We can add feature of face detection login where user can login using face detection.
● We can implement remember me feature in which if user logins once from a particular machine he does not have to login again.
● We can implement machine learning for prediction on the basis of previous searches and bookings.
● Host can have an option of graphs related to their Annual Bookings reports that can help him analyse his yearly income place/room wise.