

SOFTWARE ENGINEERING DEPARTMENT

Fall 2016



SAN JOSÉ STATE UNIVERSITY

**CMPE 273 – Enterprise Distributed Systems – Team Project
Airbnb
Team Number 4**

Team Member	SJSU ID
Aniruddha Pratap Singh	011432317
Kunal Ahuja	010731227
Mayank Sharma	011435320
Prateek Sharma	011475620

Member Contribution towards the Group Project

- **Aniruddha Pratap Singh:**
Design and development of Host Module
Pagination and Integration with Rabbit MQ
- **Kunal Ahuja :**
Design and development of trip module- backend
Google Map API Integration
- **Mayank Tanwar:**
Admin Module
User Authentication
- **Prateek Sharma :**
Design and development of trip bidding
Trip Module-front End.

❖ Introduction

Goal

The application is the replica of Airbnb. In this project we are building an application where the user can either host other people by renting his property for a short period of

time or the user can be a guest if he books property hosted by other user. The user can make payment, view the listings on map and provide reviews for other properties and people.

Purpose of System

We learn web development using MEAN stack and RabbitMQ a message broker which uses AMQP protocol for scalability. We have also used Redis for cache management.

❖ System Design

- Front End/GUI Technologies
 - HTML
 - CSS/Bootstrap
 - JavaScript
 - Angular JavaScript
- Back End/ Server Technologies
 - Node Java Script
 - Rabbit MQ (Message Broker)
- Databases
 - MongoDB
 - MySQL
 - Redis (for caching)
- Framework
 - Express

❖ Object Management Policy

- Requirement Analysis:

Planning and requirement analysis phase is the first phase of our application, as per the requirements the application is divided into three layers:

1. Presentation Layer: The presentation layer consists of the graphical user

interface of our application. The user interacts with the user interface to send requests to the server.

2. Business Layer: The business layer consists of the implementation of business logic. In this layer we translate the functional requirements into working code, that is, this layer is responsible for processing of the requests received from the user and to return valid response. The modules have been implemented using the message broker Rabbit MQ which implements the Advance Messaging and Queuing Protocol. The various modules are also integrated in this layer.
3. Data Layer: The data layer comprises of the database to store the data for various modules. In this layer we define the relationship between different tables. We have used both MongoDB and MySQL to store the data.

- Modules:

We have implemented various modules as per the requirements namely, Host, Trips, Billing and Admin. The host model stores the data about the host and the property he has listed. The trips module stores the data about the trip of the user. The billing module stores the bills for various trips and admin model has the data for all the users, their trips, lists and bills. Each of these module has its own characteristics and schema.

- Implementation:

We have implemented the user interface and various functionalities for each of the modules mentioned above. We have implemented key features like address validator to validate the address of the property entered by the user, maps using google map API which the properties in a particular area on google map. We have used Rabbit MQ as message broker between the client and server which makes the application reliable and scalable. We have also used Redis for caching sql data.

- Testing:

We have tested our application using mocha and jmeter. The various graphs shown below discusses the result of jmeter testing. We also tested application using Mocha to check if the functionalities are as expected.

❖ How we handled “Heavy Weight” resources.

The heavy weight resources in our application are the profiles images and profile videos of the user, the images of the property that the user saved while listing his property for

rent and the images uploaded by the user while reviewing a property. We have followed industry best practices and used Amazon S3 to store these images, retrieving the images from cloud is less expensive than fetching the images from database.

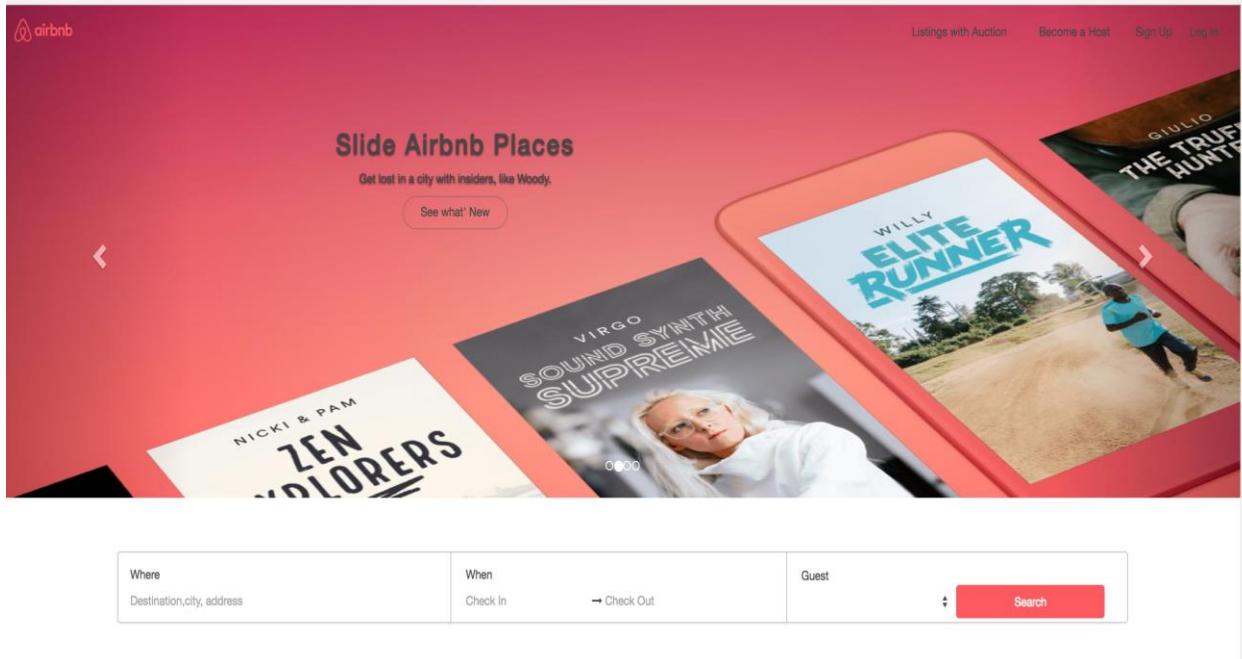
❖ **The policy you used to decide when to write data into the database**

We are storing the data into the database optimally. For Example, while posting a property on rent the user has to enter the information about the property in multiple pages, we are not storing the data into the database at each page rather we store the data in the database only at the last step, this way if the user wishes to cancel the booking in between we don't have to rollback all the data and also we don't have to store the data into the database at each step. Similarly, in trips we are fetching the listings of a particular city only once from the database and then we store this data in local storage, we use this locally saved data in the next booking screens and save the data about the booking at the last step.

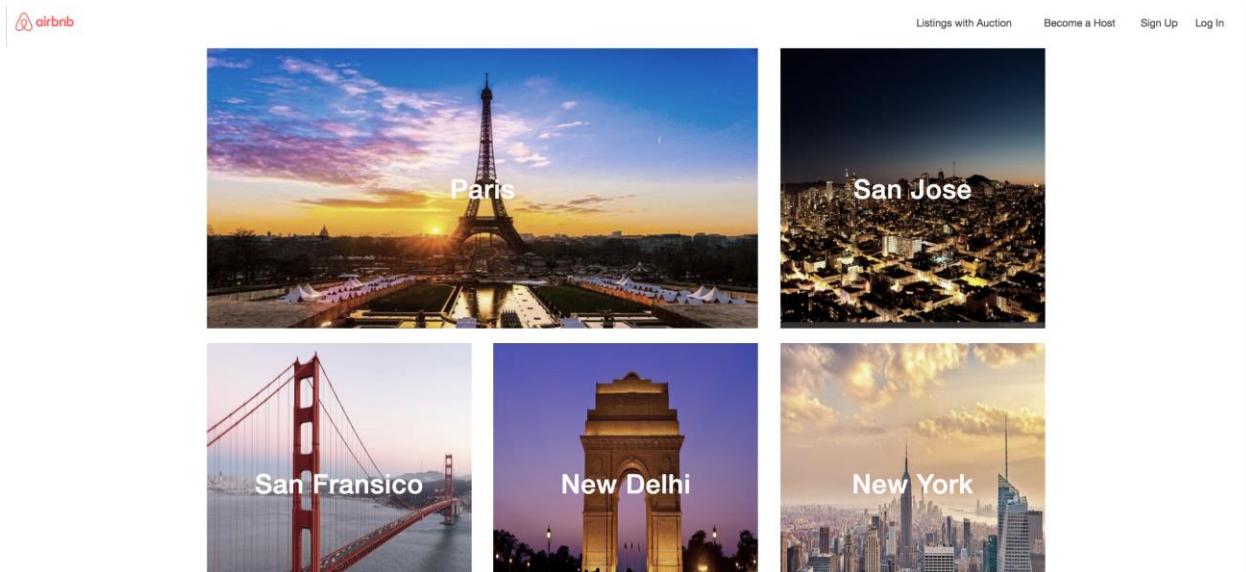
Walkthrough:

1) HomePage:

This page provides user with options like SignUp and SignIn and Search Places.



2) All the cities at home page. When customer clicks on any of the cities he is redirected to next page which shows the property under that city or user can search the city from above search bar.



3) Sign Up:

Kunal

Ahuja

kunal318@gmail.com

.....|

I'd like to receive coupons, promotions, surveys, and updates via email about Airbnb and its partners.

By signing up, I agree to Airbnb's Terms of Service, Nondiscrimination Policy, Payments Terms of Service, Privacy Policy, Guest Refund Policy, and Host Guarantee Terms.

Sign Up

Already have an Airbnb account? [Log in](#)

	When Check In → Check Out	Guest
--	------------------------------	-------

Validations:

If password is less than 8 characters

The screenshot shows the Airbnb sign-up process. The user has entered "Kunal" for their first name, "Ahuja" for their last name, and "kunal318@gmail.com" for their email address. In the password field, the user has entered "*****". A red validation message at the bottom of the form states: "Your password must be at least 8 characters. Please try again." To the right of the message is a small red square icon containing a white 'X'. Below the password field, there is a checkbox for opting into newsletters and terms of service links. At the bottom of the form is a large red "Sign Up" button. At the very bottom, there is a link for users who already have an account.

Kunal

Ahuja

kunal318@gmail.com

Your password must be at least 8 characters. Please try again.

I'd like to receive coupons, promotions, surveys, and updates via email about Airbnb and its partners.

By signing up, I agree to Airbnb's Terms of Service, Nondiscrimination Policy, Payments Terms of Service, Privacy Policy, Guest Refund Policy, and Host Guarantee Terms.

Sign Up

Already have an Airbnb account?

Log in

If Email Id is not formatted correctly(missing '@' from emailed)

Kunal

Ahuja

kunal318@gmail.com

Please enter a valid email include @

I'd like to receive coupons, promotions, surveys, and updates via email about Airbnb and its partners.

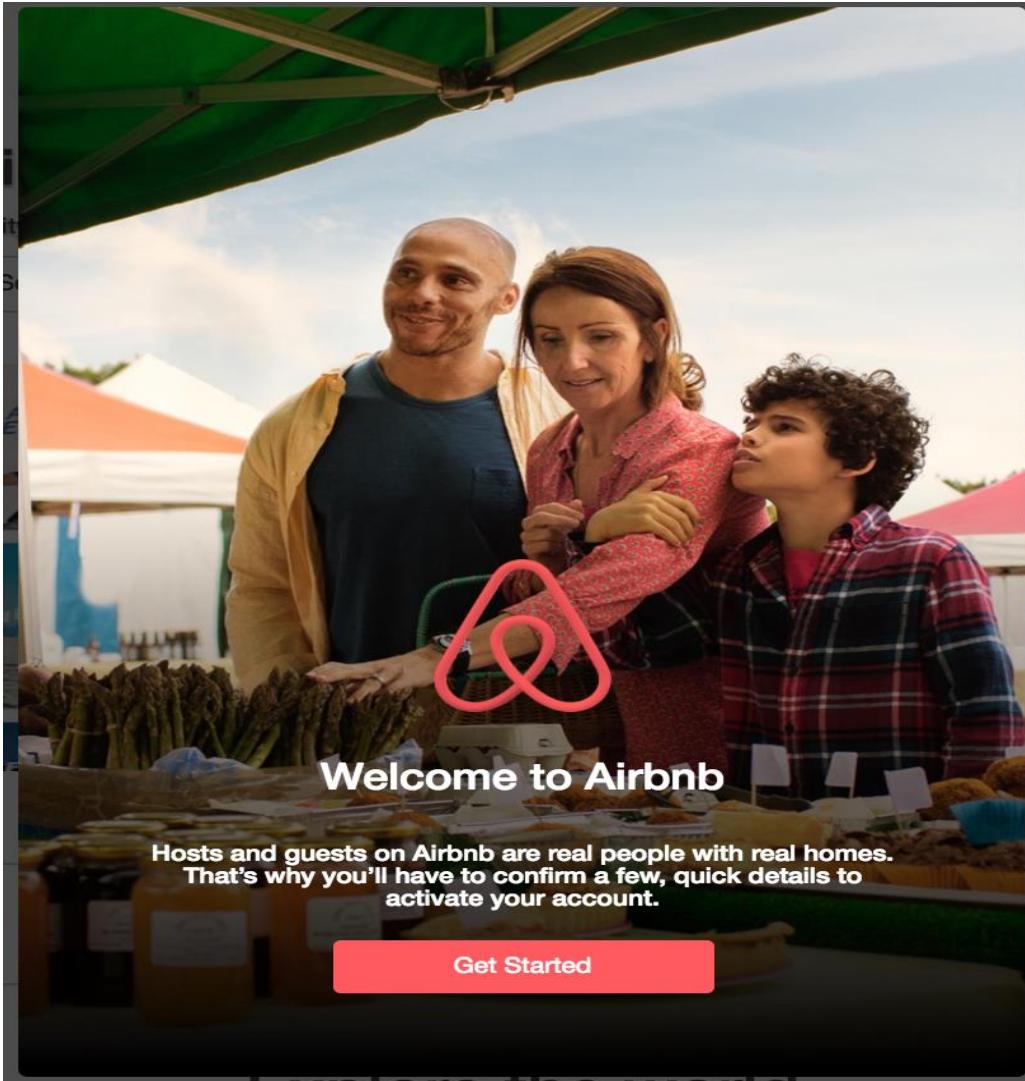
By signing up, I agree to Airbnb's Terms of Service, Nondiscrimination Policy, Payments Terms of Service, Privacy Policy, Guest Refund Policy, and Host Guarantee Terms.

Sign Up

Already have an Airbnb account?

Log in

After successfully sign up user is asked to upload profile Pic:



Add your profile photo

Put a face to your name! We'll add this to your profile, and share it with future hosts and guests.



Upload photo

Add your profile photo

Put a face to your name! We'll add this to your profile, and share it with future hosts and guests.

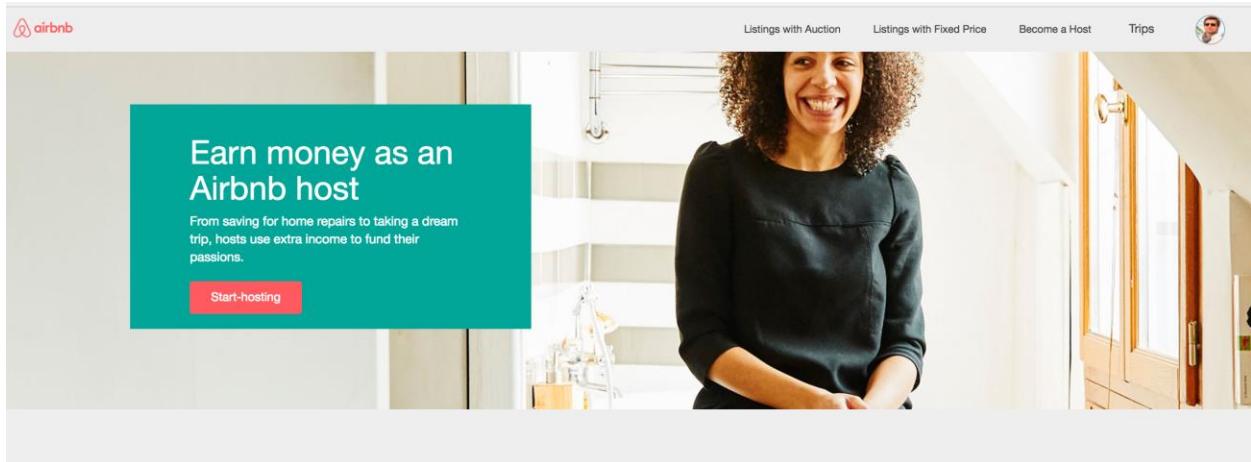


Change Photo

Upload Now

Host Module:

When user wants to host his property:



User Enters the type of place:

Hi, Kunal! Let's get you ready to become a host.

STEP 1

What kind of place do you have?

Private room

for 5 guest

San Jose

How many bathrooms ?

2 bathrooms

-

+

Continue

Address Validation:

Since this Application's main business is renting places we have validated addresses using API known as **Address Validator**. If the address is incorrect we suggest the host right addresses as shown in below screenshot. User entered incorrect address we validated from Map and gave him the correct suggestions.

Did you mean!

329 North 1st Street, San Jose, CA, US

Where's your place located?

Country
United States

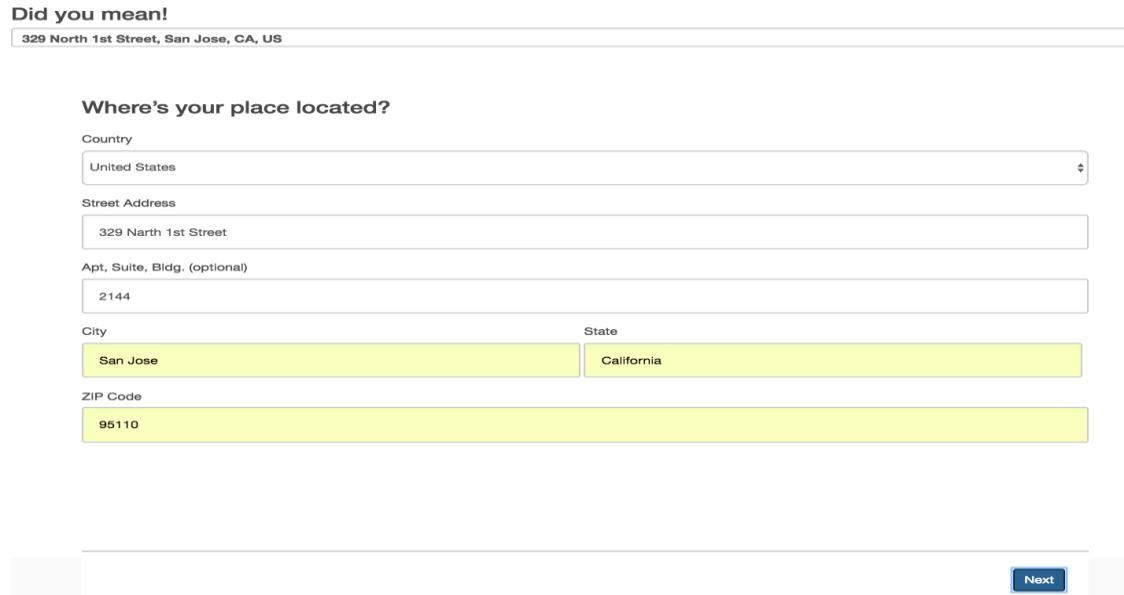
Street Address
329 North 1st Street

Apt, Suite, Bldg. (optional)
2144

City San Jose State California

ZIP Code 95110

Next



Code:

```
module.exports.addressValidator =function(req, res) {
  var json_responses = '';
  address = req.body.address + ", " + req.body.city + ", " + req.body.state + ", " + req.body.country;
  console.log("Address " + address);
  var unknownAddresses = [];
  addressValidator.validate(address, addressValidator.match.unknown, function (err, exact, inexact) {
    if (err) {
      console.log('in error');
      json_responses = {'statusCode': '401', 'error': 'Please enter valid data'};
    } else {
      if (exact != "" && exact != null) {
        console.log('in match found');
        json_responses = {'statusCode': '200', 'addressCoordinate': exact[0].location};
      } else {
        unknownAddresses = _.map(inexact, function (a) {
          return a.toString();
        });
        console.log('in no match found');
        json_responses = {'statusCode': '400', 'suggestedAddress': unknownAddresses};
      }
    }
  });
}
```

User can select amenities available in his property:

The Airbnb header includes the logo, navigation links for different listing types, and a user profile icon.

What amenities do you offer?

- Essentials Towels, bed sheets, soap, and toilet paper
- Wifi
- Shampoo
- Closet/drawers
- TV
- Heat
- Air Conditioning
- Iron
- Safety Amenities
- Smoke Detector
- First Aid and Kit

Next

Name of property:

The Airbnb header includes the logo, navigation links for different listing types, and a user profile icon.

Name your place

Gordon House

Enter a description

A beautiful Place with Swimming Pool

User can select if property is available for fixedPrice or Auction:

The Airbnb header includes the logo, navigation links for different listing types, and a user profile icon.

Auction

Fixed Price

Per day rent of the property and available dates

The screenshot shows the Airbnb listing interface. At the top, there are two buttons: "Auction" (red) and "Fixed Price" (grey). Below them, a section titled "Fixed Price for your place" contains a text input field with the value "433". Underneath, a section titled "Set your Calendar" shows date inputs. The "Available From Date" is set to "12/04/2016" and the "Available Till Date" is set to "12/21/2020". The Till Date input has a dropdown arrow icon.

If user selects available from date less than current date message is thrown:

The screenshot shows the same Airbnb listing interface as above, but with an error message overlay. The message box says: "localhost:3000 says: Available from date should be greater than today's date". There is an "OK" button at the bottom right of the message box. The rest of the interface remains the same, with the "Available From Date" set to "11/10/2016" and the "Available Till Date" set to "12/21/2020".

User can upload images of his property. We have restricted upload image limit to 24 following Airbnb.

Show travelers what your space looks like



[Remove and select other from computer](#)

[Upload Now](#)

Now after this request of this property is sent to Admin for approval. Once admin approves then only property will be visible to other users for booking.

Trip Module:

User can click on any of the cities or search for the properties in the city:

The screenshot shows the Airbnb website's trip search interface. At the top, the Airbnb logo is on the left, and navigation links for "Listings with Auction", "Host", "Trips", and a user icon are on the right. Below the header is a banner with the word "EXPLORE" and a colorful geometric pattern. The main search form is centered, divided into three sections: "Where" (containing the input "San Jose" with a teal underline), "When" (containing "Check In" and "Check Out" fields), and "Guest" (with a dropdown arrow and a "Search" button). Below the search form is a section titled "Explore the world" with the sub-instruction "See where people are traveling, all around the world". This section features two images: a vibrant sunset over a tall tower, and a dark, solid black square.

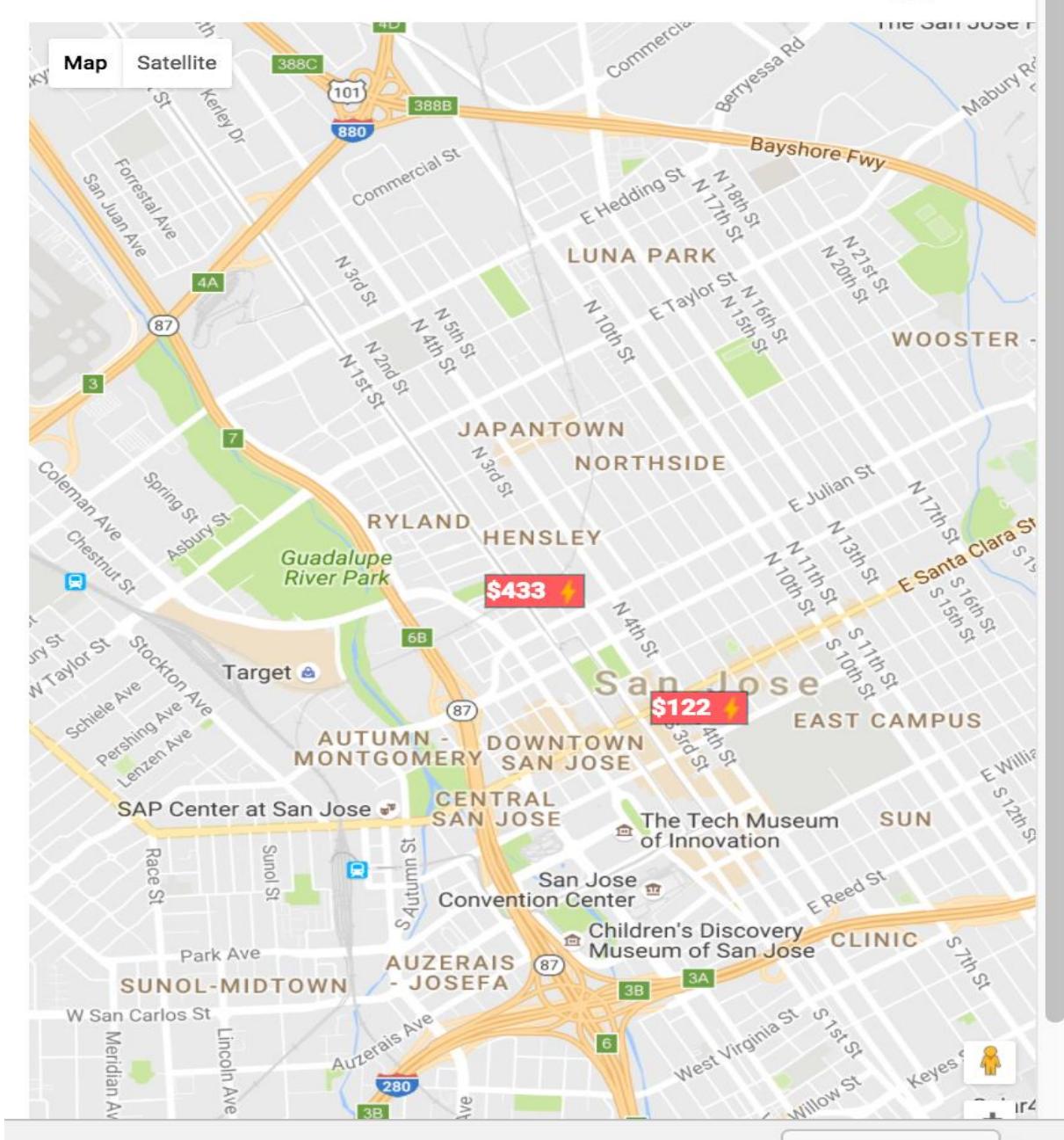
User can apply for the filters from the below screen. User sees all the properties. All the filters are function and are applied on click.

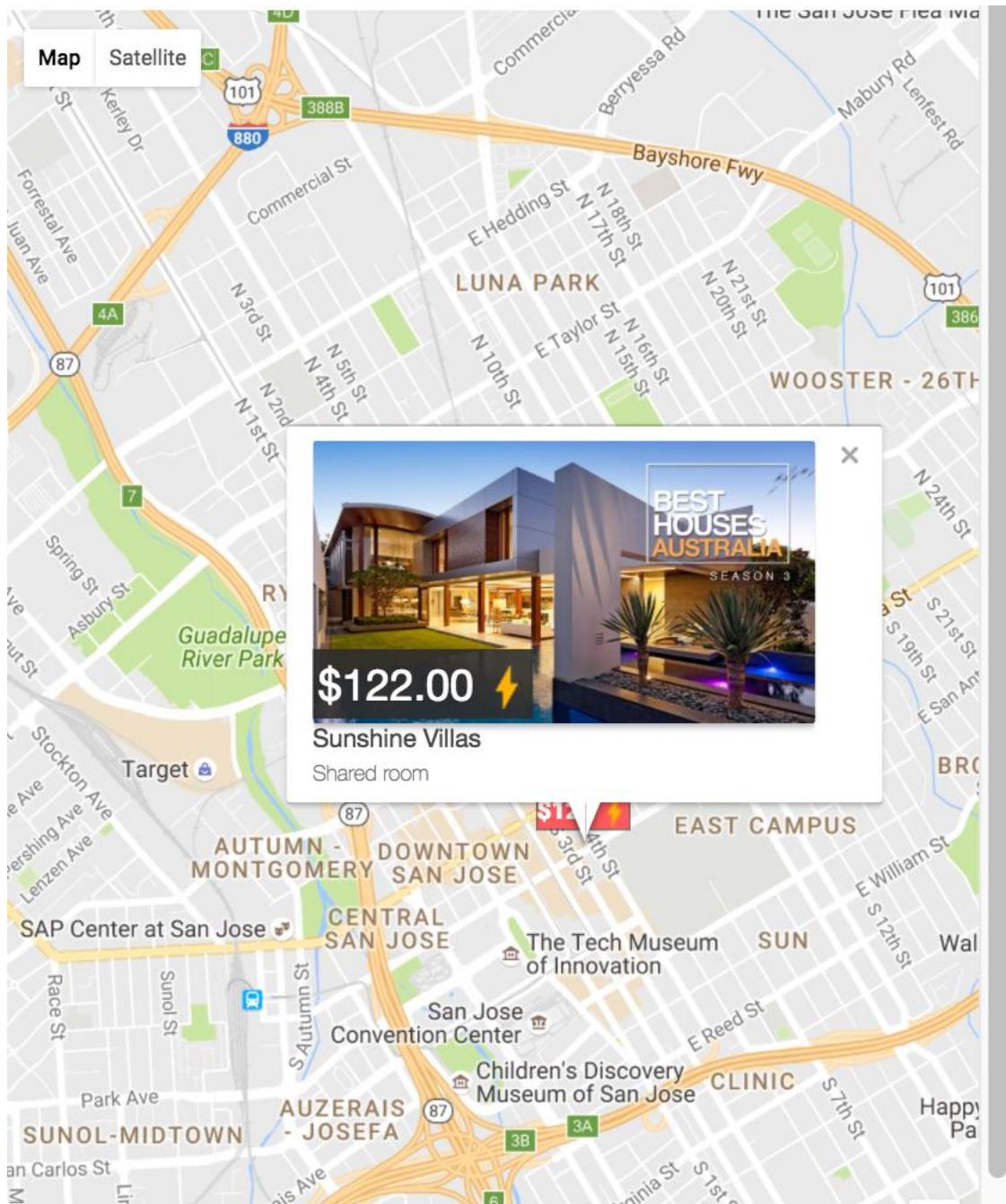
The screenshot shows the Airbnb search interface. At the top, there are date input fields ('mm/dd/yyyy') for check-in and check-out, and a dropdown for guest count. Below these are room type filters ('Entire Home', 'Private Room', 'Shared Room'). A price range slider is set between \$20 and \$2000. To the right is a map of San Jose, California, with several property locations marked by pins. One pin for 'Gordon House' is highlighted, showing a thumbnail image of a colorful building at night, the price '\$433.00', and a rating of 5 stars. Another pin for 'Sunshine Villas' is also shown. The map includes labels for various neighborhoods like Downtown San Jose, SAP Center, and The Tech Museum of Innovation.

user can also see the ratings for each and every property.

Google Map:

User can see prices of each property on Map and those objects are clickable providing images of that property





Once user clicks on any property he can view all the details of the property. He can select the checkIn and checkout dates.

The screenshot shows an Airbnb listing for 'Gordon House' in San Jose, California. The main image is a night photograph of a colorful town with a church tower reflected in a lake, surrounded by snow-capped mountains. The listing price is \$433 per night. A booking form allows users to enter check-in and check-out dates and the number of guests. Below the main image, there's a 'View Photos' button and a section for the host, Kunal. It also displays room details: a private room for 5 guests, 2 bedrooms, and 2 beds. A '100% refundable' guarantee is mentioned, along with a note about canceling up to 30 days before the trip. A 'Book' button is present, and a note states 'You would not be charged yet.'

User can view all images of the properties. We have used carousel to display images.



User can see detailed review of the properties by other users along with the real images of the properties posted by the users.

The screenshot shows a Airbnb listing page for a property. At the top, there's a navigation bar with links for 'Listings with Auction', 'Listings with Fixed Price', 'Host', 'Trips', and a user profile icon. Below the navigation, there's a section for 'CHECK-IN CHECK-OUT' with notes about check-in times and flexibility. Under 'Availability', it says '1 night minimum stay'. There are '3 Reviews' with a 4-star rating. A search bar for reviews is present. Below the reviews, there are breakdowns for Accuracy, Communication, Cleanliness, Location, Check In, and Value, each with a 4-star rating. A review by a user named 'Denis' is shown, with a small profile picture and a positive comment about the apartment's location and amenities. The overall layout is clean and organized, typical of a travel booking platform.

We have applied all possible validations for selecting dates:

The image displays two side-by-side screenshots of a booking interface, likely from a mobile application or website. Both screenshots show a summary bar at the top with a price of '\$433 ⚡ Per Night'. Below this, there are fields for 'Check In' (set to '12/06/2016') and 'Check Out' (set to '12/01/2016'), and a 'Guests' selector set to '2'. A large red 'Book' button is at the bottom of each panel. Validation messages are displayed below the input fields:
Left panel: 'Checkin date should be before CheckOut date'
Right panel: 'Checkin date and CheckOut date cannot be same'
Both panels also include a note at the bottom stating 'You would not be charged yet'.

\$433 ⚡ Per Night

Check In Check Out Guests

11/03/2016 12/01/2016 2

Book

Please select a date after current date

You would not be charged yet

Since user has selected while listing his property that property is available only after 2016-12-06 and same goes for checkout validation as well.

\$433 ⚡ Per Night

Check In Check Out Guests

12/05/2016 02/16/2017 2

Book

CheckIn date should be after 2016-12-06

You would not be charged yet

Once user selects the correct dates, he is redirected to final page where he can leave comments for the host and once he sends the request, request goes to approval for host and user is redirected to 'view trips' page.

The screenshot shows the Airbnb booking confirmation page. At the top left is the Airbnb logo. The main content area is titled "About Your Trip". It features a photo of the host, Kunal, and a listing for "Gordon House A beautiful Place with Swimming Pool, San Jose". Below this, there's a section for "Who's coming?" with a dropdown set to "2" and a note about maximum guests allowed. A text input field contains the message: "Hey, Please reserve this for me.". Under "House Rules", it lists "No Smoking" and "No parties". To the right, there's a sidebar with a photo of the listing, the host's name, and details like room type, address, and dates. The total cost is listed as \$3,464.00.

Booking Successful.

This screenshot shows the same Airbnb booking confirmation page as above, but with a modal dialog box overlaid. The dialog box contains the message "localhost:3000 says: Booking Successful" and a checkbox option "Prevent this page from creating additional dialogs." An "OK" button is visible at the bottom right of the dialog. The rest of the page content is visible through the dialog window.

View Trips

Unless hosts approves it user does not gets to pay and user can edit and cancel trip. Similarly user can see his previous trips and trips which he bid for.

The screenshot shows the Airbnb dashboard with the 'Your Trips' tab selected. On the left, there are navigation links for 'Upcoming Trips', 'Previous Trips', and 'Auction Trips'. The main content area displays 'Your Upcoming Trips' with two entries:

Status	Location	Host	Dates	Payment Status	Options
PENDINGHOSTAPPROVAL	101 East San Fernando Street San Jose-95112	Brad	Check In: Dec 21, 2016 Checkout: Jan 27, 2017	PENDING	<button>Pay</button> <button>Edit Trip</button> <button>Cancel Trip</button>
PENDINGHOSTAPPROVAL	329 North 1st Street San Jose-95110	Kunal	Check In: Dec 20, 2016 Checkout: Dec 28, 2016	PENDING	<button>Pay</button> <button>Edit Trip</button> <button>Cancel Trip</button>

If admin approves the request, Pay button will be enabled and user can pay the amount. Once user clicks on Pay he gets an option to pay the amount. If user has already added the credit card in his profile the fields are already populated:

We have also applied the checks, credit card number should be 16 digits, cvv should be 3 digits and expiry should be greater than today's date.

The screenshot shows the same Airbnb dashboard and 'Your Trips' page as above, but with a modal window open over the first trip entry. The modal title is 'Please fill the card details' and contains the following message: 'Card Number should be of 16 digits'. The modal form includes fields for Card Number (11223445), Expiry (Dec 21, 2016 - Jan 27, 2017), Security Code, Country, First Name, and Last Name. At the bottom of the modal, it says 'Total: \$4,514.00' and has 'Cancel' and 'Make Payment' buttons. The 'Make Payment' button is highlighted with a red border. The background of the page shows the second trip entry with its own set of 'Pay', 'Edit Trip', and 'Cancel Trip' buttons.

User can view the bill once he pays the amount. View Bill Button is made visible after paying.

The screenshot shows the Airbnb 'Your Trips' section. At the top, there's a navigation bar with the Airbnb logo, 'airbnb', and links for 'Listings with Auction', 'Listings with Fixed Price', 'Host', 'Trips', 'Dashboard', 'Your Listing', 'Profile', and 'Account'. On the left, there are three categories: 'Upcoming Trips' (selected), 'Previous Trips', and 'Auction Trips'. The main content area is titled 'Your Upcoming Trips' and lists two trips:

Status	Location	Host	Dates	Payment Status	Options
APPROVEDBYHOST	101 East San Fernando Street San Jose-95112	Brad	Check In: Dec 21, 2016 Checkout: Jan 27, 2017	PAID	Pay View Bill
APPROVEDBYHOST	329 North 1st Street San Jose-95110	Kunal	Check In: Dec 20, 2016 Checkout: Dec 28, 2016	PENDING	Edit Trip Cancel Trip

Billing

Our Bill has covered all the fields like property name, Receipt number(ssn format), dates, amount, host, guests etc..

Customer Receipt

Confirmation Code: 590
Wed March 7, 2012
Receipt # 590-31-7835

Airbnb
+1 (669) 263 7801
WWW.AIRBNB.COM/CONTACT

 Kunal	 San Jose
NAME	TRAVEL DESTINATION
 101 East San Fernando Street	 Kingdom Palace
ACCOMODATION ADDRESS	PROPERTY
 brad@gmail.com	 2
HOST	GUESTS
 2016-12-21 01:00PM	 2017-01-27 12:00PM
CHECKIN	CHECKOUT
 37	 \$0
NIGHTS	SECURITY DEPOSIT

Payment Details

Accomodations	\$4,514.00(\$122.00 per night)
Service Fee	\$0
Total	\$4,514.00

Payment Received: Sun, May 01, 2016 **\$4,514.00**
Card : XXXXXXXXXXXXXXX7788


Thanks for travelling with us

User Profile:

A screenshot of the Airbnb profile edit page. The page has a header with the Airbnb logo and navigation links: Listings with Auction, Listings with Fixed Price, Host, Trips, and a user icon. The main content area is a form for editing user details. The form fields and their values are:

First Name	Kunal
Last Name	Smith
I Am	Male
Birth Date	12/13/2016
Email Address	kunal312@gmail.com
Phone Number	6692637801
Preferred Language	English
Preferred Currency	USD
Street Address	329 N 1st Street
City	San Jose
State	California
Country	United States
Zip Code	95110
Describe Yourself	I am a good host

User can view his profile.

A screenshot of the Airbnb profile view page. The page has a header with the Airbnb logo and navigation links: Dashboard, Your Listing, Your Trips, Profile (which is selected), Account, Listings with Auction, Listings with Fixed Price, Host, Trips, and a user icon. The main content area shows a summary of the user's profile information under the heading "User Profile". On the left, there is a placeholder for a user photo with the name "Kunal" and links to View Profile and Edit Profile. On the right, the "User Details" section displays the following information:

First Name:	Kunal
Last Name:	Smith
Gender:	male
Birth Date:	2016-12-13
Email Address:	kunal312@gmail.com
Phone Number:	6692637801
Preferred Language:	English
Preferred Currency:	USD
Street Address:	329 N 1st Street
City:	San Jose
State:	California
Country:	United States
Zip Code:	95110

User can upload his profile pic and video.

The screenshot shows the 'Profile Photo' section of the Airbnb profile. It features a large placeholder image of a person's head and shoulders in a blue gradient. Below the placeholder are two buttons: 'Take a photo with your Webcam' and 'Upload a file from your computer'. A note at the top right says 'Images tell a lot about oneself, please upload a clear picture which will help people to learn about you!'. A small orange 'required' asterisk is located next to the 'Upload a file from your computer' button.

Profile Photo

Images tell a lot about oneself, please upload a clear picture which will help people to learn about you!

Take a photo with your Webcam

Upload a file from your computer ***required**

Profile Video

You haven't recorded a video yet!

Video Checklist!

- We suggest that you sit at arm's length.
- You have got 30 seconds, so do your best to be concise.
- Try to keep the light in front of you.
- Look into the camera to create that personal connection.

Upload a file from your computer

***required**

The screenshot shows the 'Profile Photo' and 'Profile Video' sections of the Airbnb profile after uploads. In the 'Profile Photo' section, a new circular placeholder image shows a scenic view of a city skyline with flowers in the foreground. Below the image are the same 'Webcam' and 'Upload' buttons, and a red 'Save Profile Pic' button is visible. In the 'Profile Video' section, a thumbnail of a video showing a person's face is displayed, with playback controls at the bottom. Below the thumbnail is a red 'Upload Video' button. The 'Video Checklist!' and its bullet points remain the same as in the first screenshot.

Profile Photo

Images tell a lot about oneself, please upload a clear picture which will help people to learn about you!

Take a photo with your Webcam

Upload a file from your computer

Save Profile Pic

Profile Video

You haven't recorded a video yet!

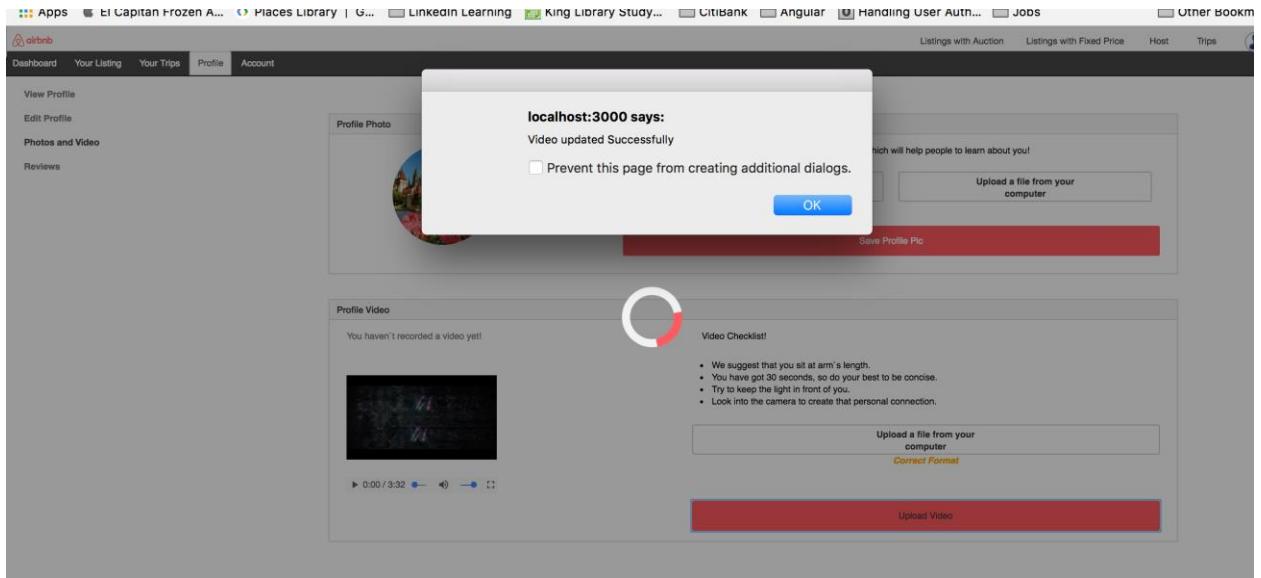
Video Checklist!

- We suggest that you sit at arm's length.
- You have got 30 seconds, so do your best to be concise.
- Try to keep the light in front of you.
- Look into the camera to create that personal connection.

Upload a file from your computer

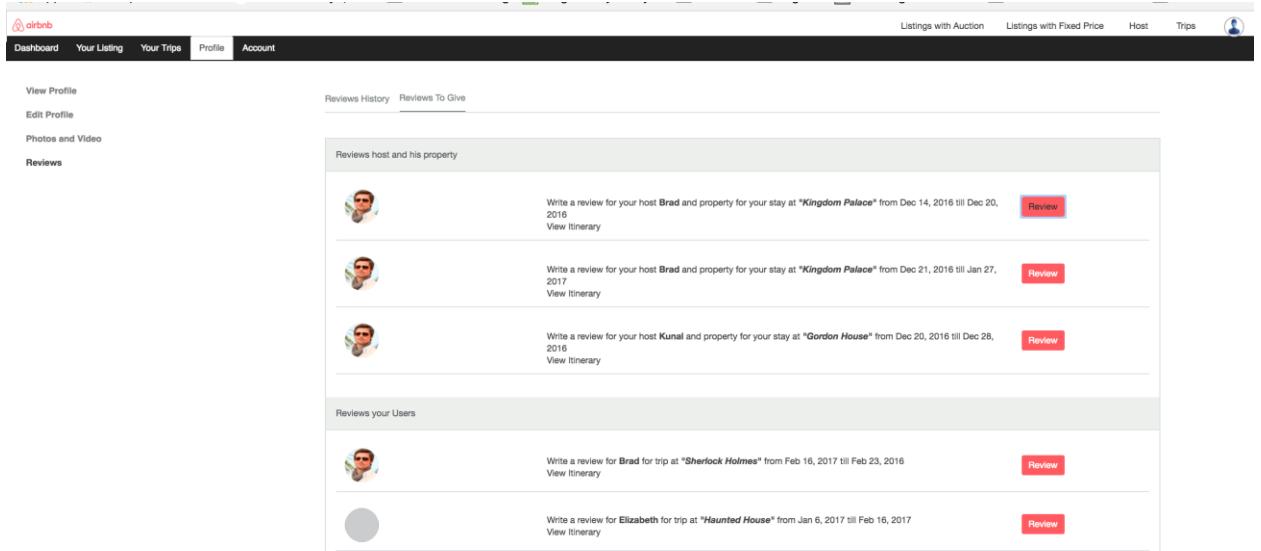
Correct Format

Upload Video



Reviews:

User can review the properties and host once the trip gets completed.





Review your host Brad for
stay at property "**Kingdom
Palace**"



x

**What you loved best about your
stay**

Wonderful..

Cleanliness Feedback



Was very clean

Accuracy Feedback



Accuracy of the place

Location Feedback



Location of the place

Communication Feedback



Communication of the host

CheckIn Feedback



Checkin time of the place

Value Feedback



what you loved about the place

Upload images for your review

Upload Images

Submit Review

User can also upload real images:

Was very clean

Accuracy Feedback ★★★★☆

Accuracy of the place

Location Feedback ★★★★★

Location of the place

Communication Feedback ★★★★☆

Communication of the host

CheckIn Feedback ★★★★★

CheckIn time of the place

Value Feedback ★★★★☆

what you loved about the place

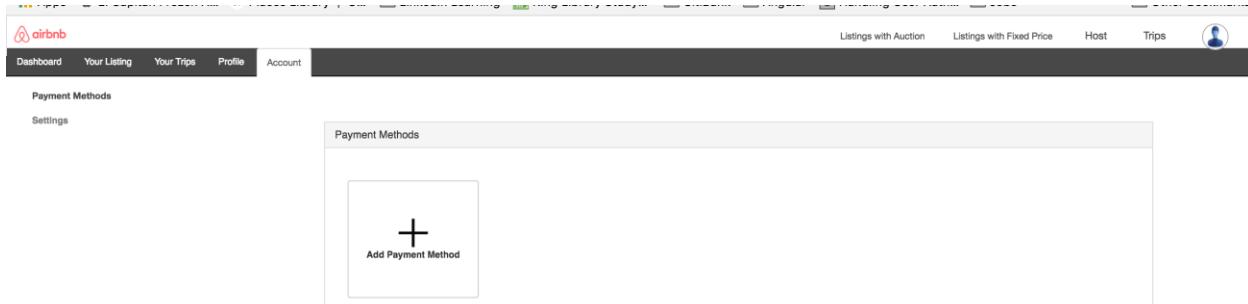
Upload images for your review



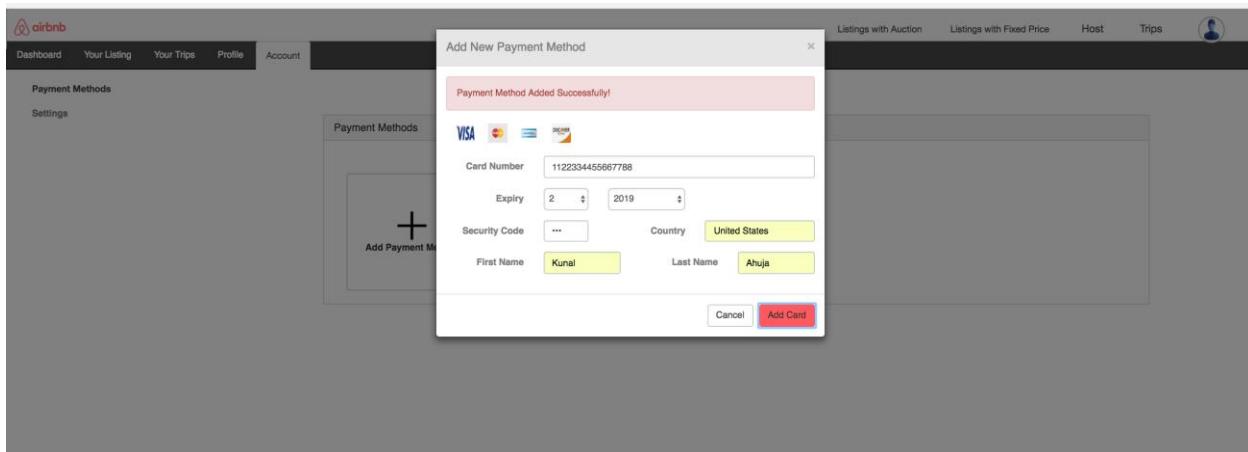
Upload Images

Submit Review

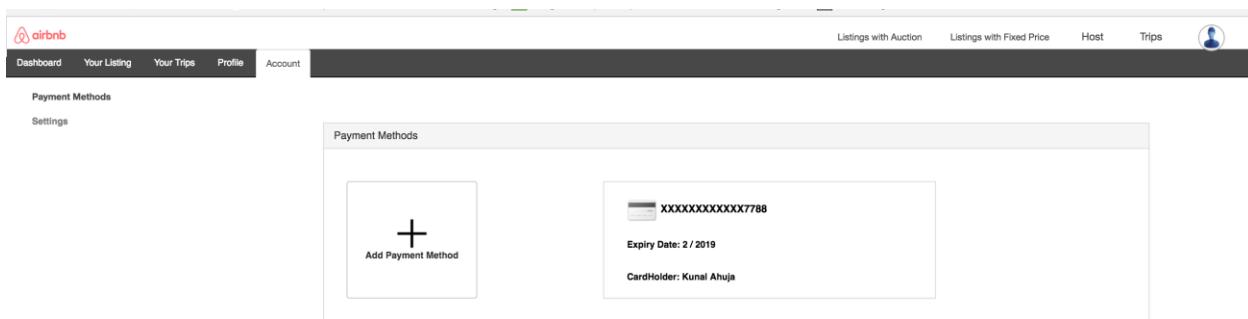
User can add his credit card details:



The screenshot shows the Airbnb account settings interface. The top navigation bar includes links for Dashboard, Your Listing, Your Trips, Profile, and Account. Under the Account section, there are links for Payment Methods and Settings. A large central box is titled "Payment Methods" and contains a button labeled "Add Payment Method" with a plus sign icon.



A modal window titled "Add New Payment Method" is displayed over the payment methods list. It shows a success message: "Payment Method Added Successfully!". The form fields include: Card Number (1122334455667788), Expiry (2 / 2019), Security Code (redacted), Country (United States), First Name (Kunal), and Last Name (Ahuja). At the bottom are "Cancel" and "Add Card" buttons, with "Add Card" being red.



The screenshot shows the "Payment Methods" page again. The newly added card is listed with the following details: a small card icon, the number "XXXXXXXXXXXX7788", the expiry date "Expiry Date: 2 / 2019", and the cardholder name "CardHolder: Kunal Ahuja".

User can manage his listings, check who have booked his properties.

The screenshot shows the Airbnb dashboard under the 'Host' tab. On the left, there's a sidebar with links: 'Your Listings', 'Your Reservations', and 'Manage Listing'. The main area is titled 'Reservations' and lists two entries:

Status	Dates and Location	Guest	Details	Payment-Details
TRIPCOMPLETED	16.02.2017 --> 23.02.2016 Sherlock Holmes London, 43567	Brad	Bill	Amount: \$854 XXXX-XXXX-XXXX-7788
APPROVEDBYHOST	06.01.2017 --> 16.02.2017 Haunted House San Jose, 95110	Elizabeth	Bill	Amount: \$14145 XXXX-XXXX-XXXX-7788

Host Dashboard:

The screenshot shows the Host Dashboard. On the left, there's a profile section for 'Kunal' with options to 'View Profile' or 'Edit Profile'. To the right, there's a 'Notifications' panel with three sections: 'Pending Requests', 'Inbox', and a bell icon.

User can click on any pending requests to approve the requests:

The screenshot shows the 'Pending Request' section of the Host Dashboard. It lists two pending requests:

Status	Dates and Location	Guest	Details	Actions
pendingHostApproval	16.02.2017 --> 23.02.2016 Sherlock Holmes London, 43567	Brad	\$854 total Message History Print Confirmation	Accept Cancel
pendingHostApproval	06.01.2017 --> 16.02.2017 Haunted House San Jose, 95110	John Elizabeth	\$14145 total Message History Print Confirmation	Accept Cancel

Bidding:

User can place a bid selecting listing auctions:

The screenshot shows a listing for a house in San Jose, CA, United States. The listing title is "G house Bidding avail". It features a photo of a person's face, a rating of 4 stars from 10 reviews, and a host named Kunal. The price is \$112 per night. The listing includes options for Entire Place, 5 Guests, 2 Bedroom, and 2 Beds. A "100% refundable" badge is present. The booking period is from 12/17/21 to 02/03/21 for 2 guests. The total cost is \$5376. A minimum bid amount of \$5376 is specified. A bid amount of \$6000 is entered in the "Bid Amount" field, and a "Place Bid" button is visible.

Admin Dashboard:

The Admin Dashboard shows the following metrics:

- Revenue: 25.141k
- Bookings: 7
- Properties: 10283
- Users: 10058

Pending Approval Listings for admin:

Screenshot of a web application interface showing a pending listing for approval.

The sidebar on the left shows navigation links: Dashboard, Listings, and review.

The main area is titled "Pending Listings".

A listing card is displayed:

- Image:** A night photograph of a modern house with a large swimming pool.
- Host:** Kunal
- Type:** House
- Title:** agag - Shared room
- Address:** ...
- Listing Details:**
 - Listing Id:** 522-29-9133
 - Facility:** Laundry – washer,
 - Basic Amenities:** Shampoo,
 - Number of Bathrooms:** 2
 - From Date:** Dec 7, 2016 12:00:00 AM - **To Date:** Sep 29, 2017 12:00:00 AM
 - Price:** 144 USD
- Status:** newListing
- Action Buttons:** Accept (green) and Reject (red)

Approved Listing:

Screenshot of a web application interface showing an approved listing.

The sidebar on the left shows navigation links: Dashboard, Listings, and review.

The main area is titled "Approved Listing".

A listing card is displayed:

- Image:** A night photograph of a modern house with a large swimming pool.
- Host:** Kunal
- Type:** Haunted House
- Title:** Ghost house - Entire Place
- Address:** ...
- Listing Details:**
 - Listing Id:** 004-24-2784
 - Facility:** Laundry – dryer,
 - Basic Amenities:** Air Conditioning, Iron,
 - Number of Bathrooms:** 2
 - From Date:** Dec 31, 2016 12:00:00 AM - **To Date:** Dec 31, 2018 12:00:00 AM

Admin can view the users:

The screenshot shows a web-based administrative interface. On the left, there is a sidebar with navigation links: 'Dashboard', 'Listings', and 'review'. The main content area has a red header bar with the text 'Review Users'. Below the header, there is a search bar with a placeholder 'Enter City Name' and two buttons: 'Search' (green) and 'Reset grid' (red). The main area is titled 'User Details' and contains a table with the following data:

First Name	Last Name	User Id	Email	Contact	Host	City	Created on	Profile
Kunal	Smith	503-24-0005	kunal312@gmail.com	6692637801	true	San Jose	12/3/16 12:00 AM	<button>View</button>
Elizabeth	Smith	047-46-4164	ebryantgr@cloudflare.com		true		12/3/16 12:00 AM	<button>View</button>
Frances	Smith	069-84-9108	fcole8u@yandex.ru		true		12/3/16 12:00 AM	<button>View</button>
Heather	Smith	503-56-4970	hlittle1n@nbcnews.com		true		12/3/16 12:00 AM	<button>View</button>
Richard	Smith	652-07-5959	rpowell@themeforest.net		true		12/3/16 12:00 AM	<button>View</button>
Bobby	Smith	646-56-8309	bhernandez96@usnews.com		true		12/3/16 12:00 AM	<button>View</button>
Jacqueline	Smith	496-52-2617	jdeanhf@php.net		true		12/3/16 12:00 AM	<button>View</button>
Craig	Smith	379-64-3372	chamiltonhg@bizjournals.com		true		12/3/16 12:00 AM	<button>View</button>

The screenshot shows a web-based administrative interface. On the left, there is a sidebar with navigation links: 'Dashboard', 'Listings', and 'review'. The main content area has a red header bar with the text 'Review hosts'. Below the header, there is a search bar with a placeholder 'Enter City Name' and two buttons: 'Search' (green) and 'Reset grid' (red). The main area is titled 'Host Details' and contains a table with the following data:

First Name	Last Name	User Id	Email	Contact	City	Host	Created on	Profile
Kunal	Smith	503-24-0005	kunal312@gmail.com	6692637801	San Jose	true	12/3/16 12:00 AM	<button>View</button>
Elizabeth	Smith	047-46-4164	ebryantgr@cloudflare.com			true	12/3/16 12:00 AM	<button>View</button>
Frances	Smith	069-84-9108	fcole8u@yandex.ru			true	12/3/16 12:00 AM	<button>View</button>
Heather	Smith	503-56-4970	hlittle1n@nbcnews.com			true	12/3/16 12:00 AM	<button>View</button>
Richard	Smith	652-07-5959	rpowell@themeforest.net			true	12/3/16 12:00 AM	<button>View</button>
Bobby	Smith	646-56-8309	bhernandez96@usnews.com			true	12/3/16 12:00 AM	<button>View</button>
Jacqueline	Smith	496-52-2617	jdeanhf@php.net			true	12/3/16 12:00 AM	<button>View</button>
Craig	Smith	379-64-3372	chamiltonhg@bizjournals.com			true	12/3/16 12:00 AM	<button>View</button>

Admin can view the bills:

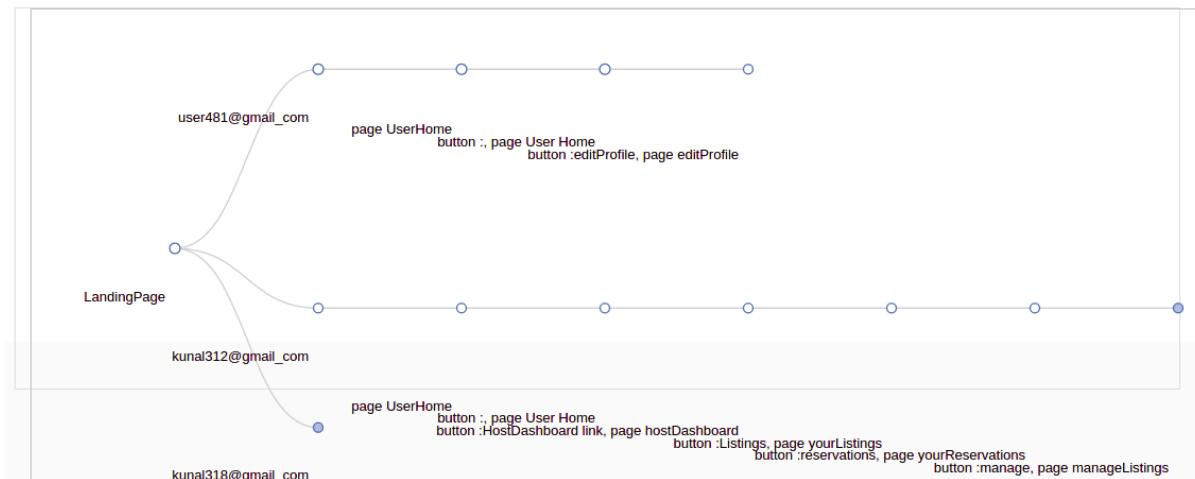
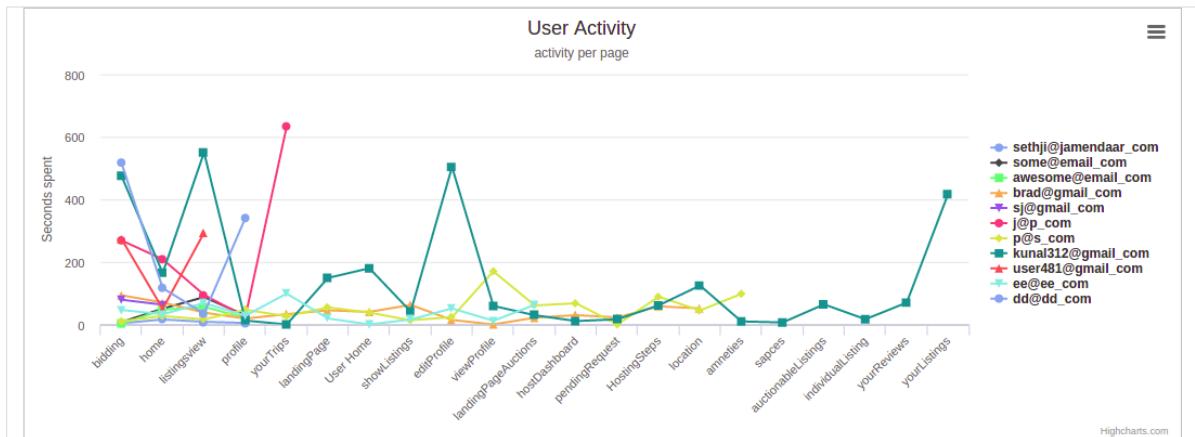
The screenshot shows a web-based application interface for managing bills. On the left, there is a sidebar with navigation links: 'Dashboard' (selected), 'Listings', and 'review'. The main content area has a red header bar with the text 'Review hosts'. Below this, the title 'Enter Bill date' is displayed, followed by a search input field 'mm/dd/yyyy' and a green 'Search by date' button. A section titled 'Search by Month and Year' contains two dropdown menus for selecting months and years, with a green 'Search by month' button next to them. A red 'Reset grid' button is located above a table. The table has a header row with columns: Bill Id, Bill Date, Host Name, Listing City, Payment Status, Trip Status, User Email, check In Date, check Out Date, and View Bill. There are five data rows in the table, each with a 'View' button in the last column.

Bill Id	Bill Date	Host Name	Listing City	Payment Status	Trip Status	User Email	check In Date	check Out Date	View Bill
497-44-1823	12/2/16 12:00 AM	Kunal	London	paid	tripCompleted	brad@gmail.com	2/16/17 12:00 AM	2/23/16 12:00 AM	<button>View</button>
445-25-2380		Brad	San Jose	cancelled	pendingHostApproval	kunal312@gmail.com	12/14/16 12:00 AM	12/20/16 12:00 AM	<button>View</button>
590-31-7835	12/4/16 12:00 AM	Brad	San Jose	paid	pendinghostApproval	kunal312@gmail.com	12/21/16 12:00 AM	1/27/17 12:00 AM	<button>View</button>
516-60-2431	12/4/16 12:00 AM	Kunal	San Jose	paid	pendingHostApproval	ebryantgr@cloudflare.com	1/6/17 12:00 AM	2/16/17 12:00 AM	<button>View</button>

Admin Analytics graphs:

❖ Admin Dashboard





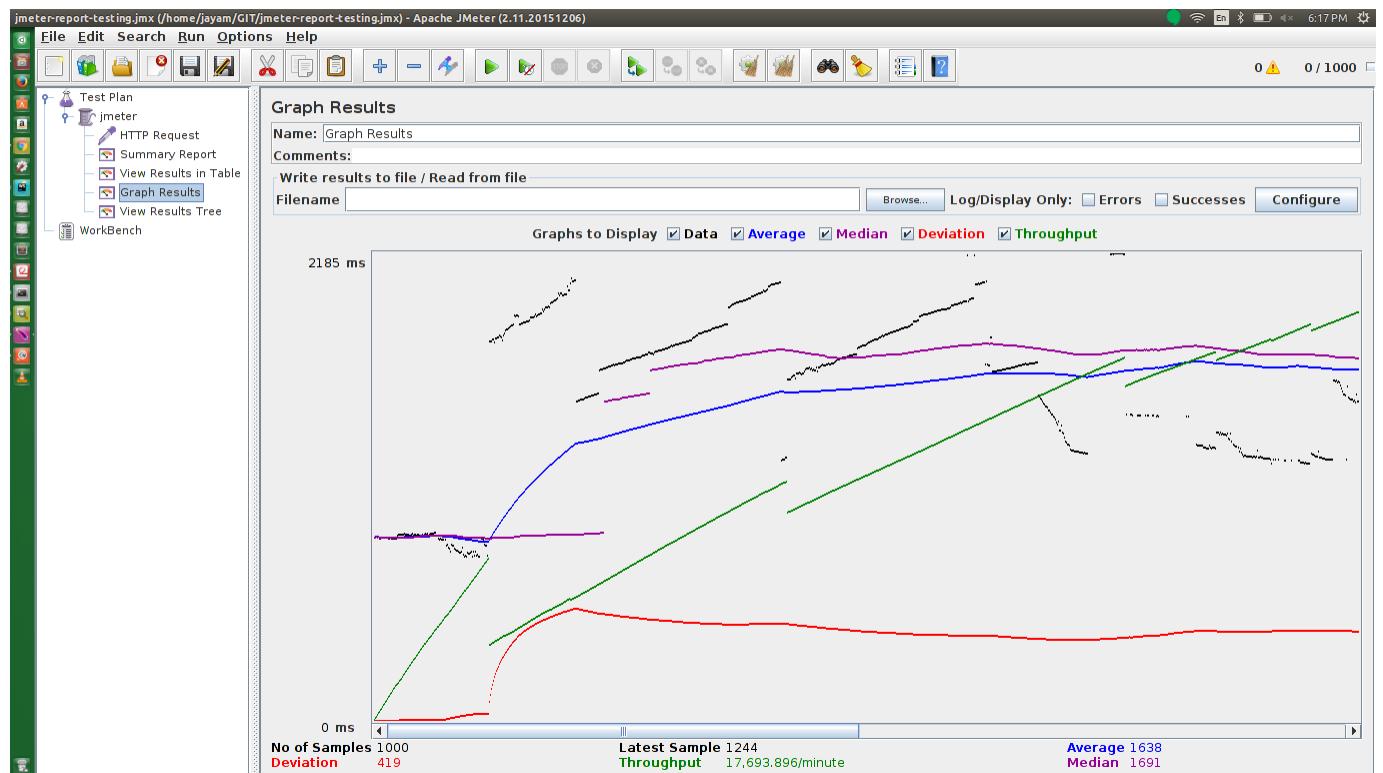
Testing:

❖ **Jmeter Graphs**

- **1000 Requests - Base (Simple Database Connection)**

Throughput / minute - 17693.896

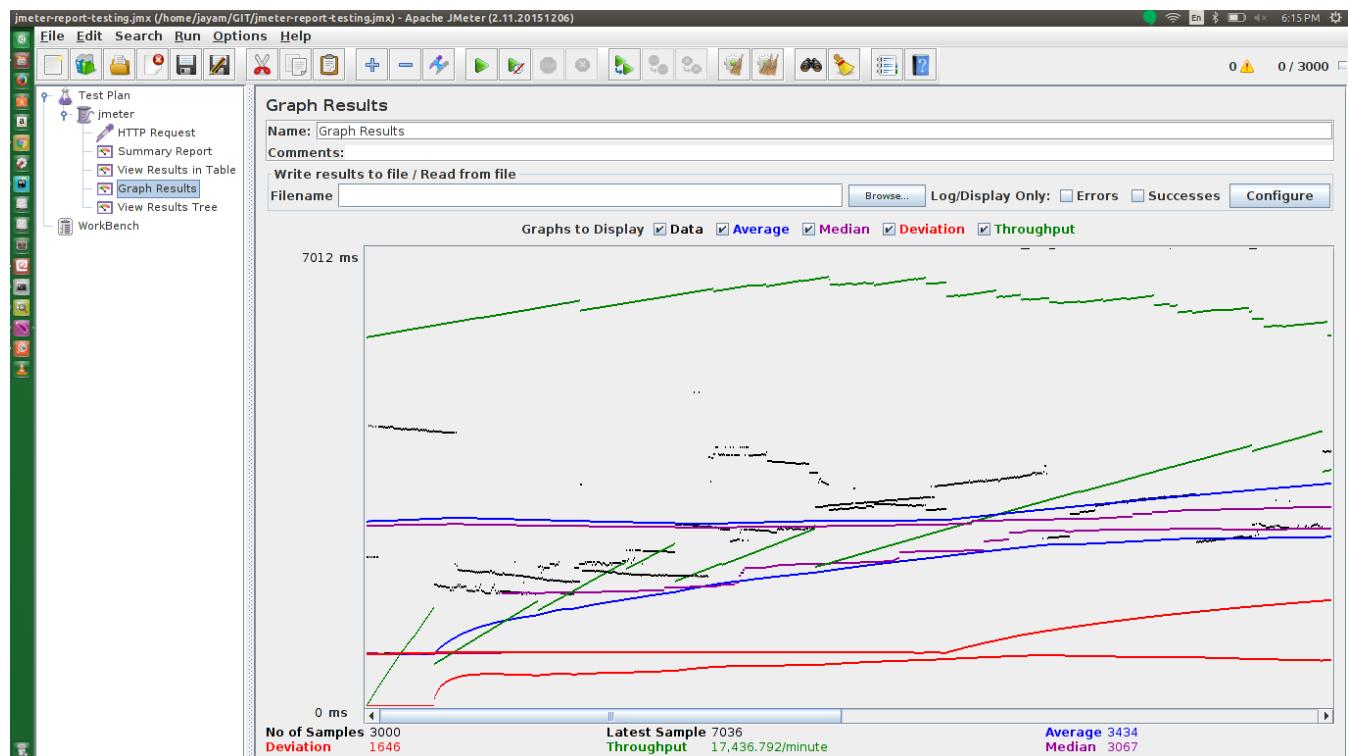
Average - 1638 ms



- **3000 Requests - Base (Simple Database Connection)**

Throughput / minute - 17436.792

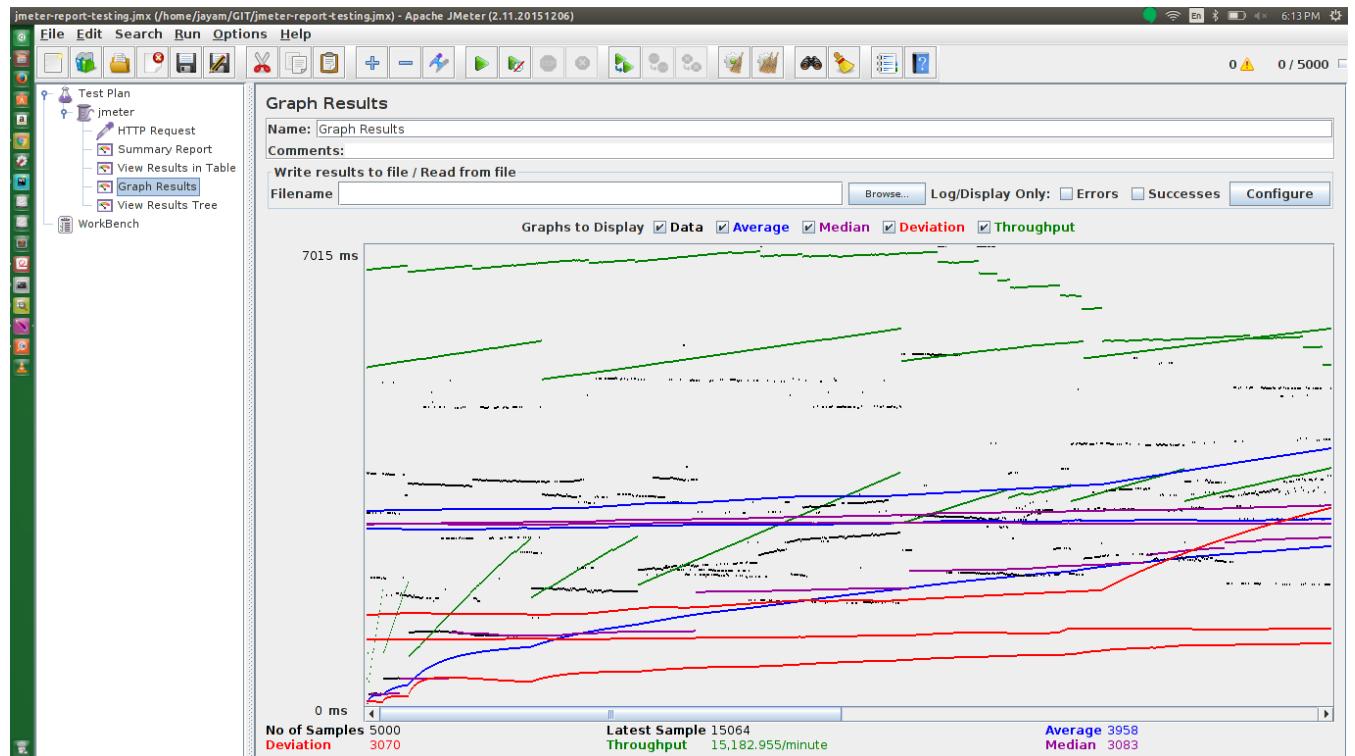
Average - 3434



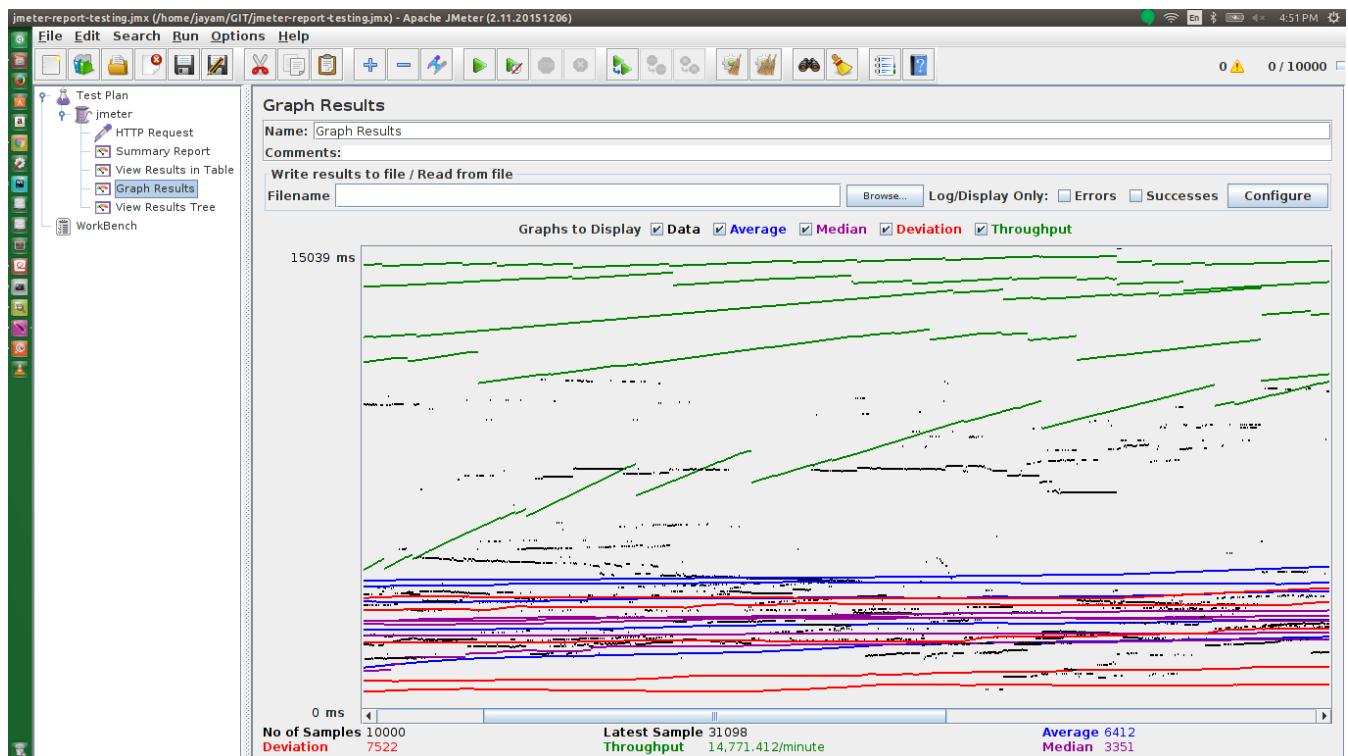
- **5000 Requests - Base (Simple Database Connection)**

Throughput / minute - 15182. 955

Average - 3958



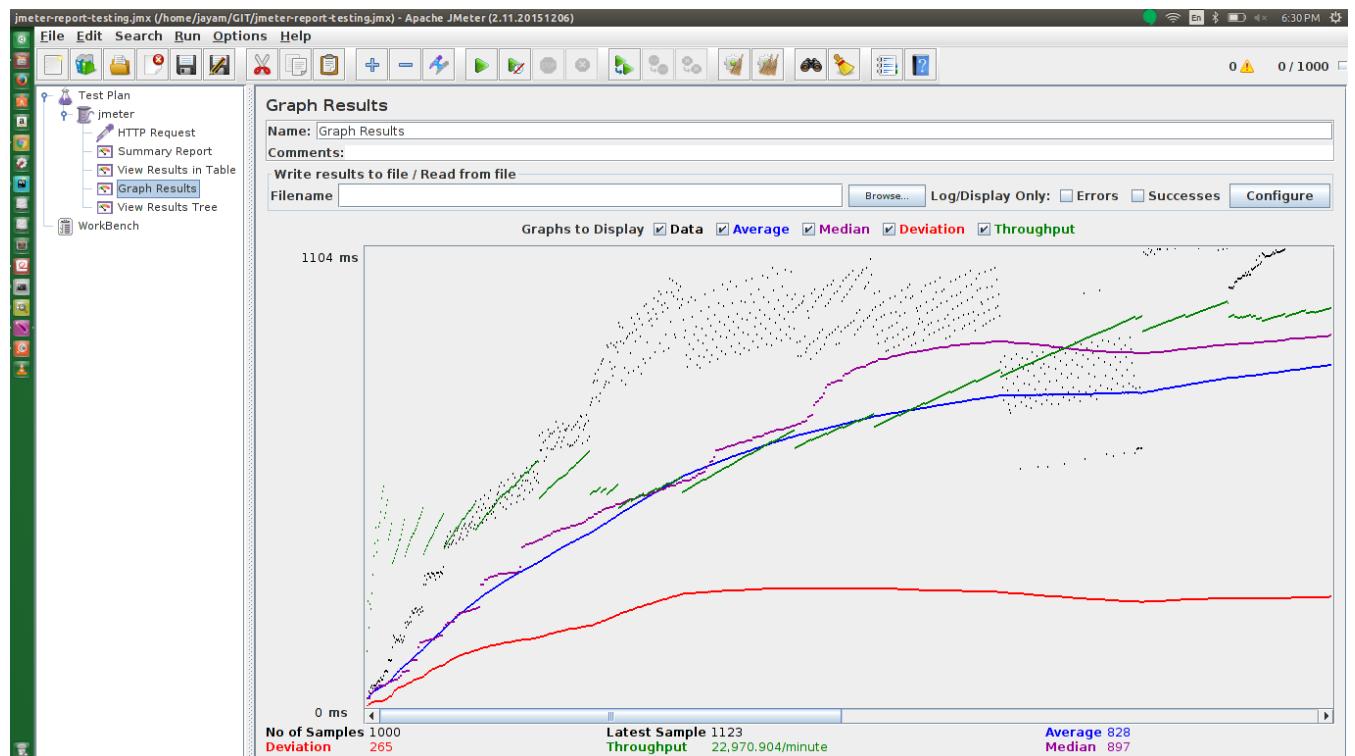
- **10000 Requests - Base (Simple Database Connection)**
Throughput / minute - 14771.412 Average - 6412



- **1000 Requests - Base + Connection Pooling**

Throughput / minute - 22970.904

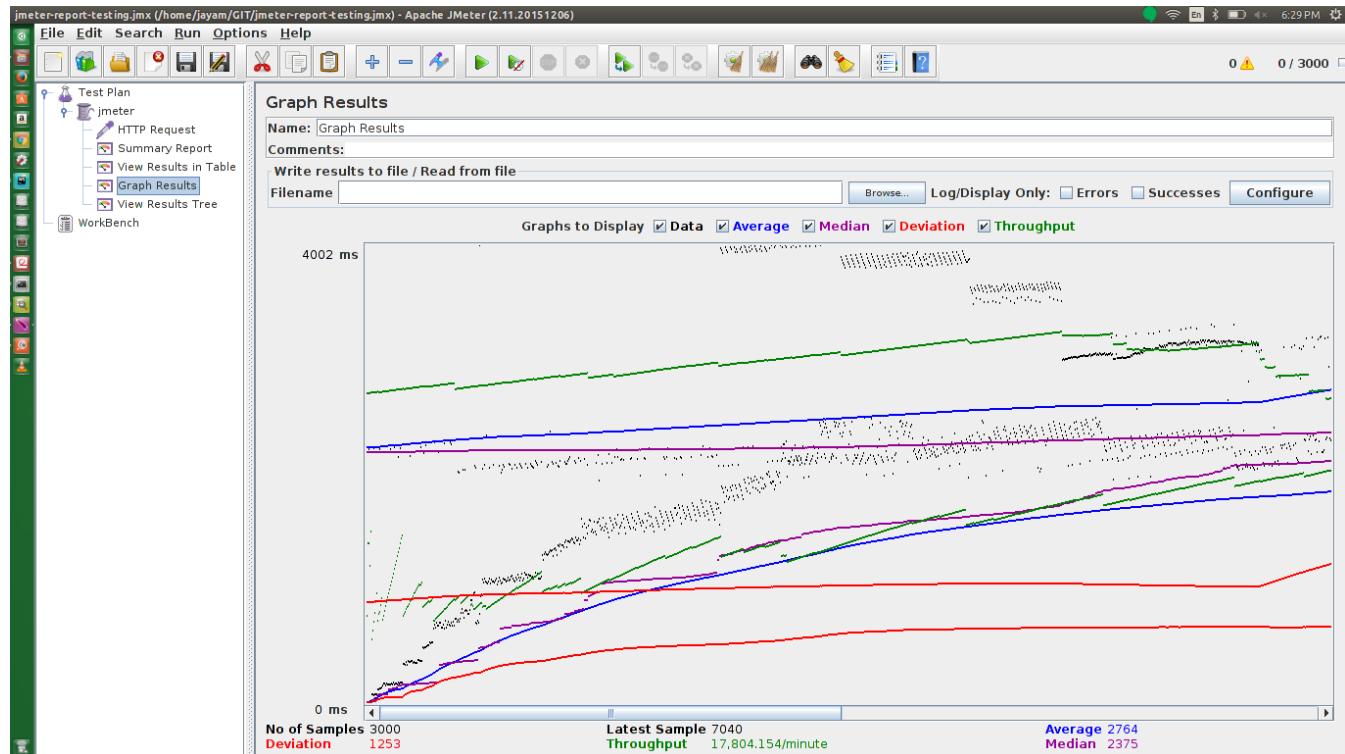
Average – 828



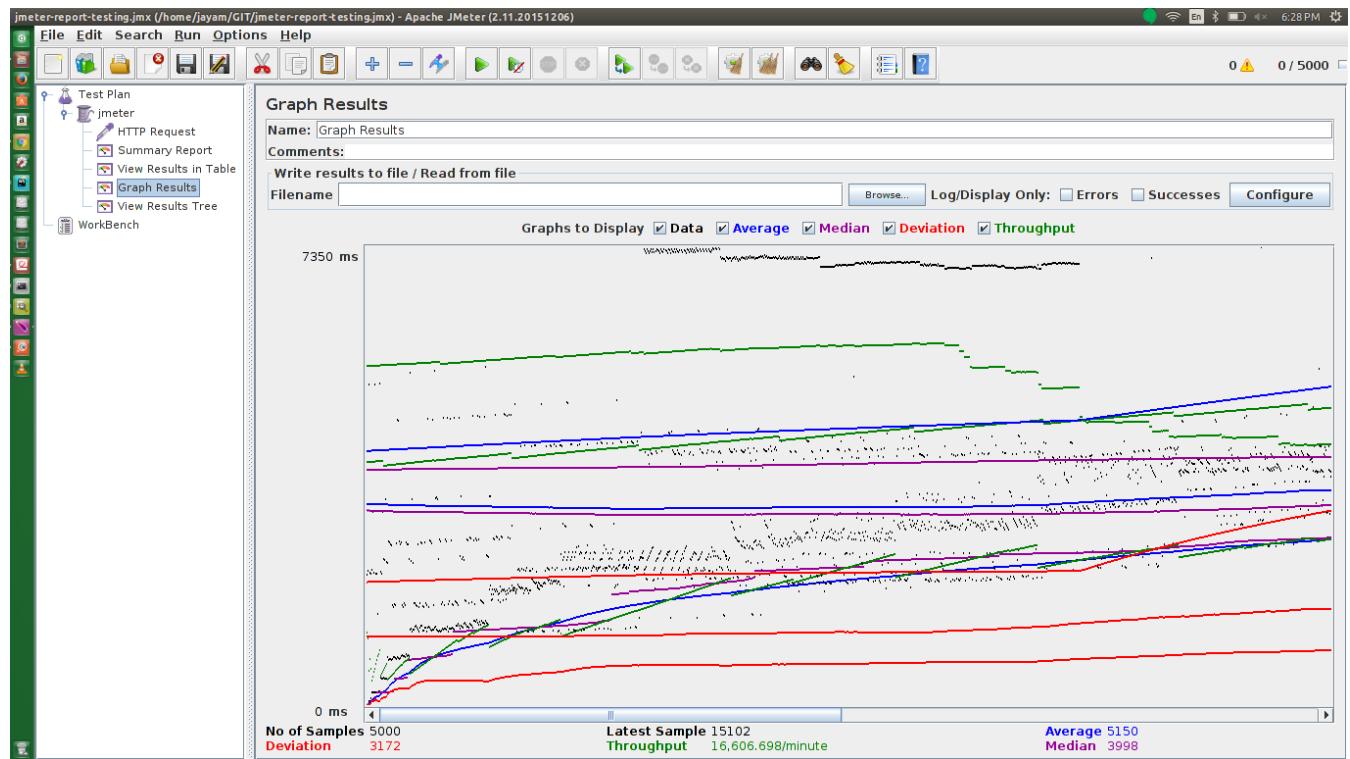
- **3000 Requests - Base + Connection Pooling**

Throughput / minute - 17804.154

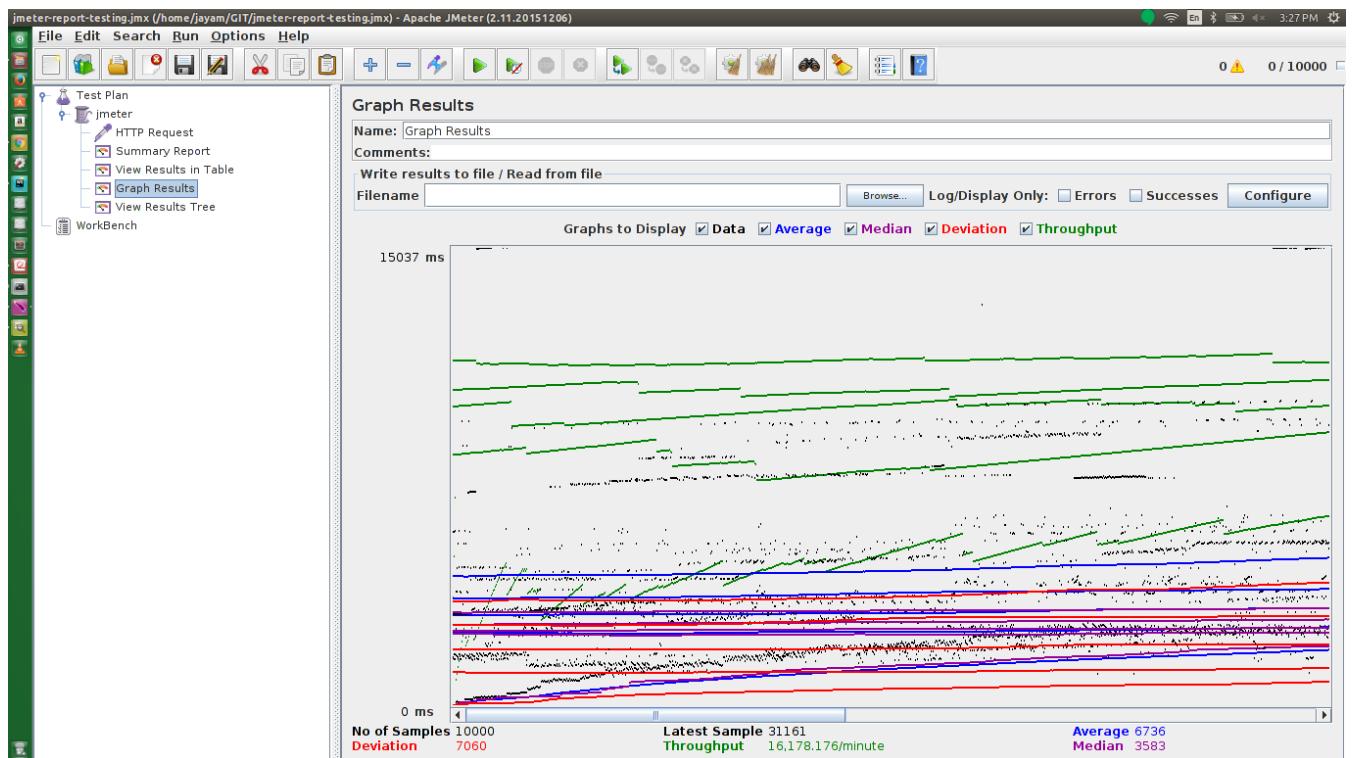
Average – 2764



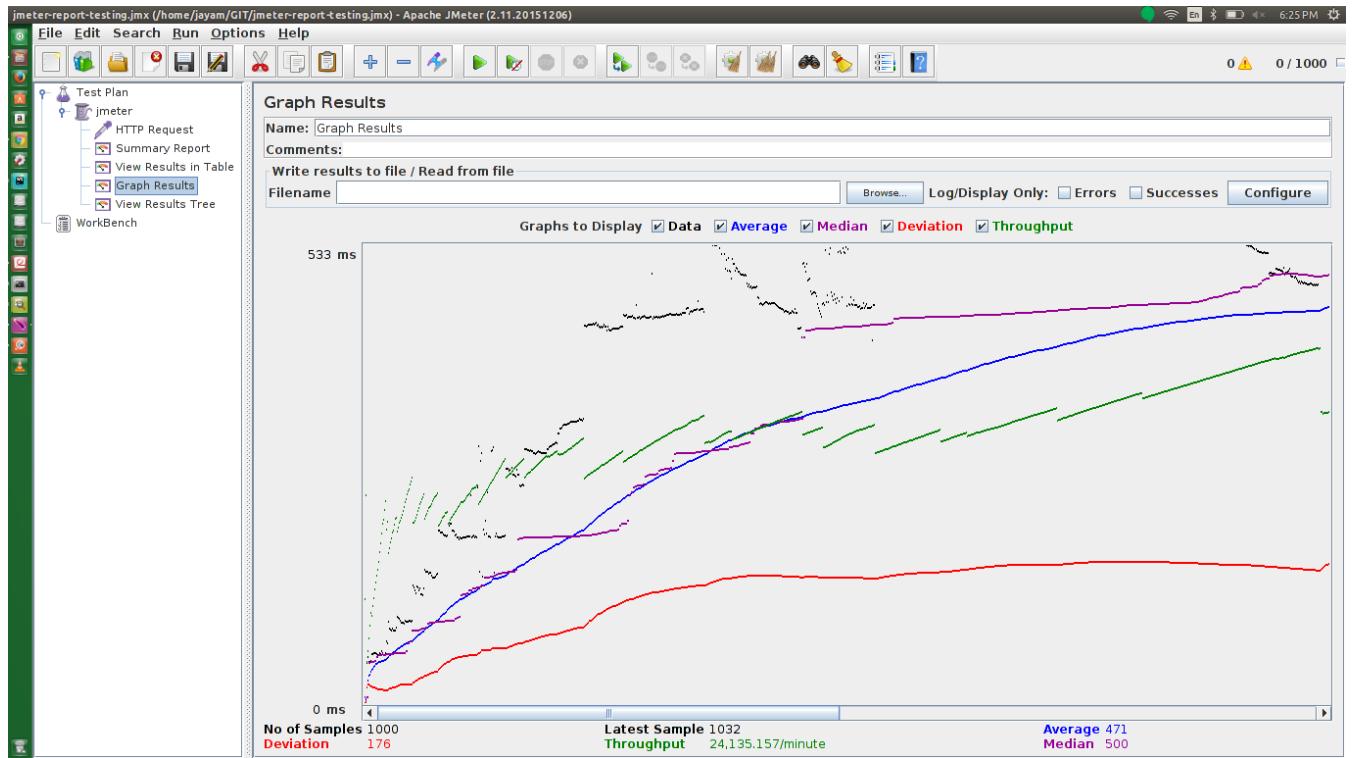
- **5000 Requests** - Base + Connection Pooling
 Throughput / minute - 16606.698 Average – 5150



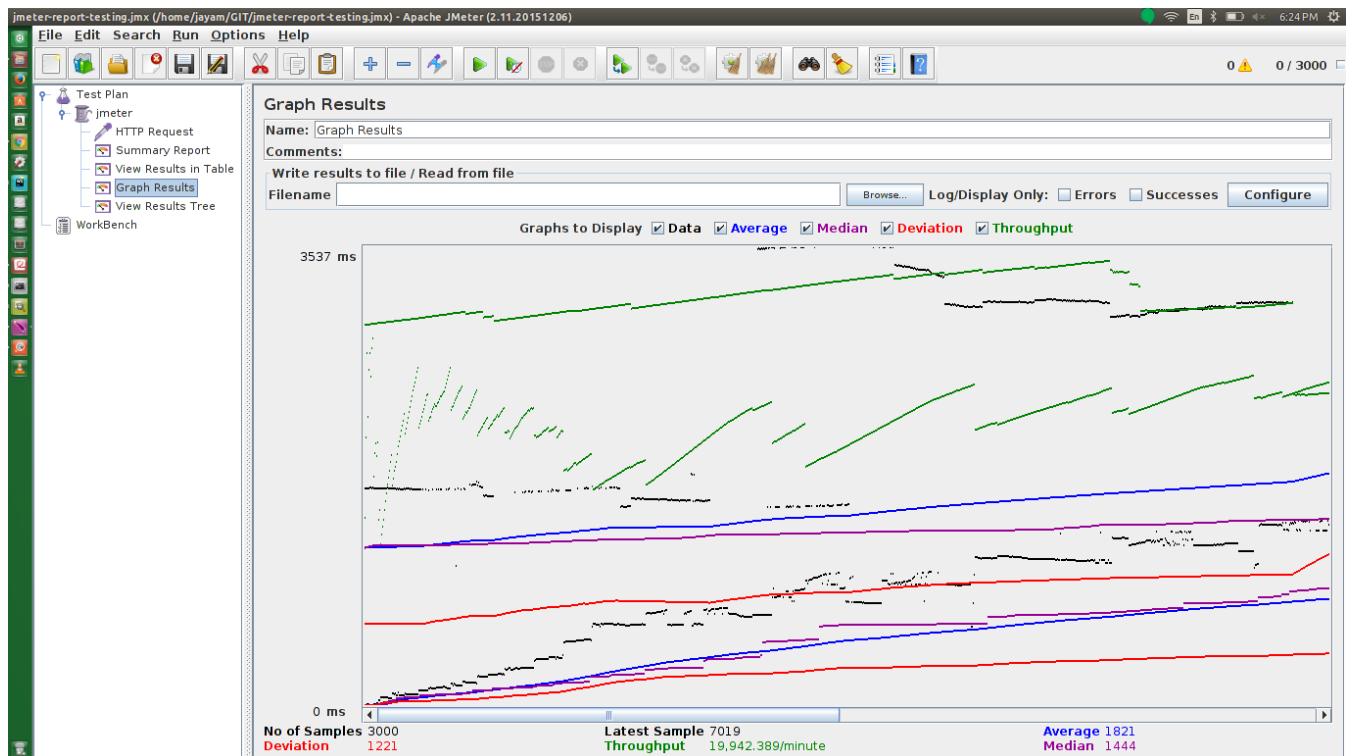
- **10000 Requests** - Base + Connection Pooling
Throughput / minute - 16178.176 Average – 6736



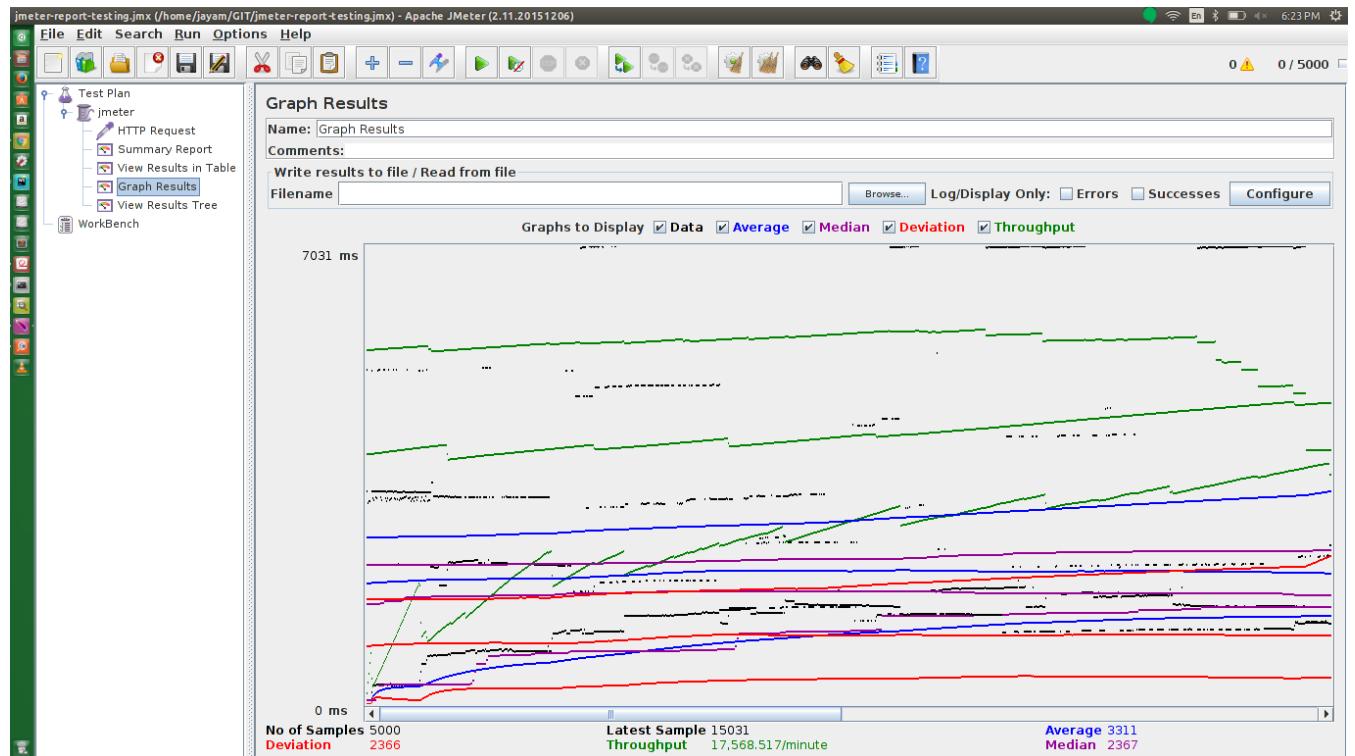
- **1000 Requests** - Base + Connection Pooling + RabbitMQ
 Throughput / minute - 24135.157 Average – 471



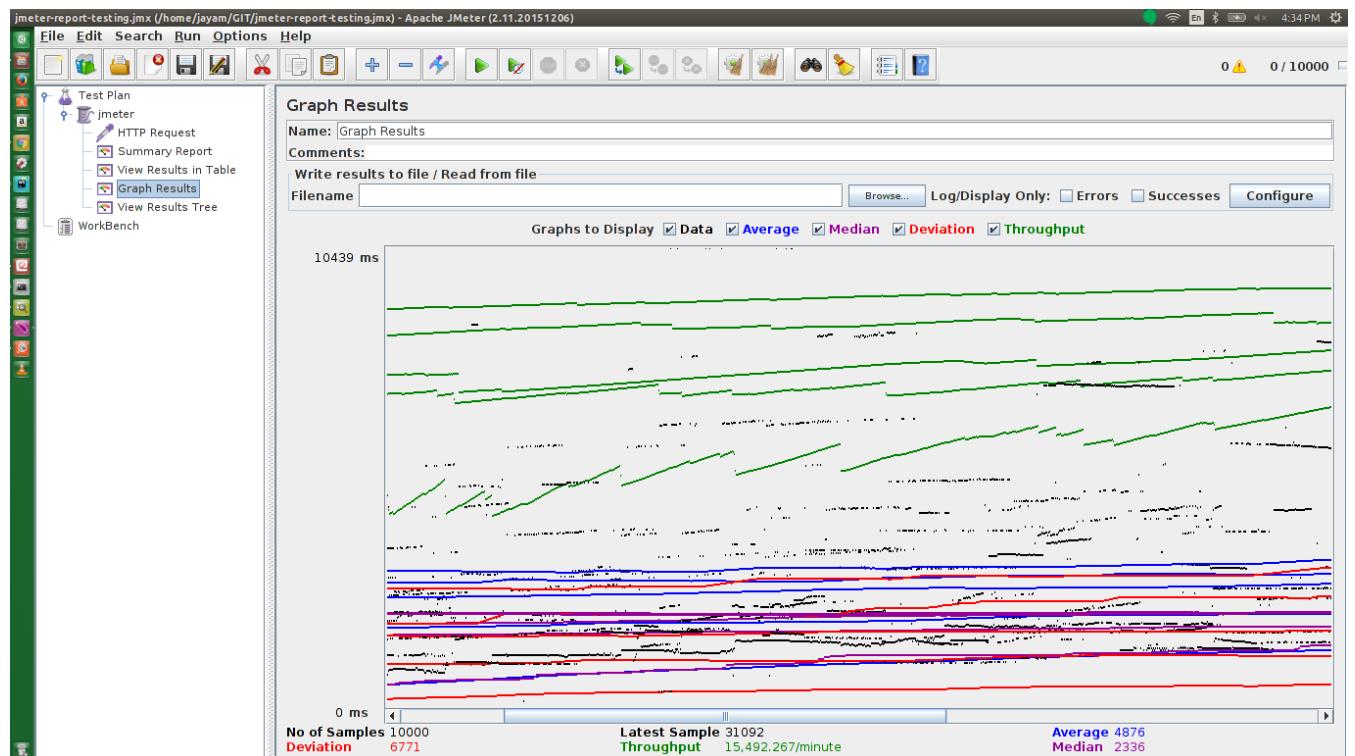
- **3000 Requests** - Base + Connection Pooling + RabbitMQ
 Throughput / minute - 19942.389 Average – 1821



- **5000 Requests** - Base + Connection Pooling + RabbitMQ
 Throughput / minute - 17568.517 Average – 3311



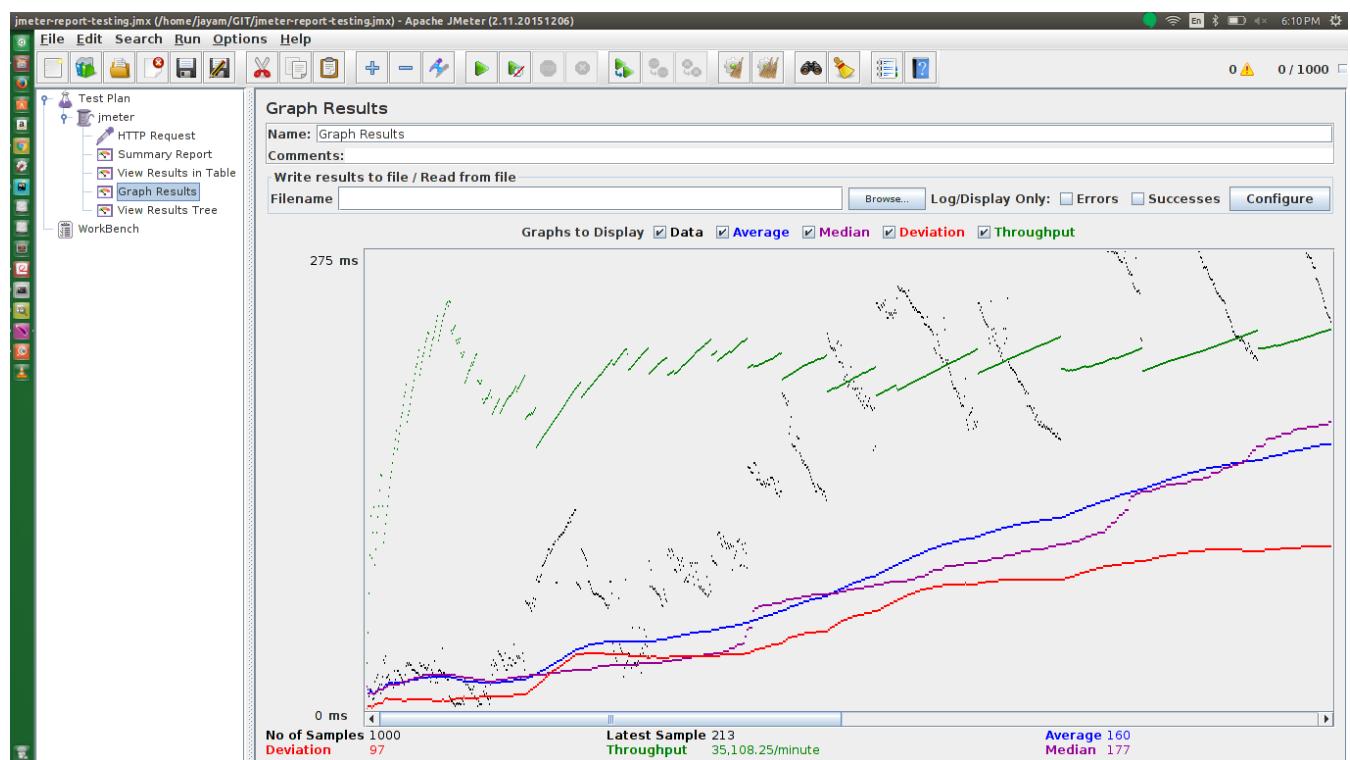
- **10000 Requests** - Base + Connection Pooling + RabbitMQ
Throughput / minute - 15492.267 Average – 4876



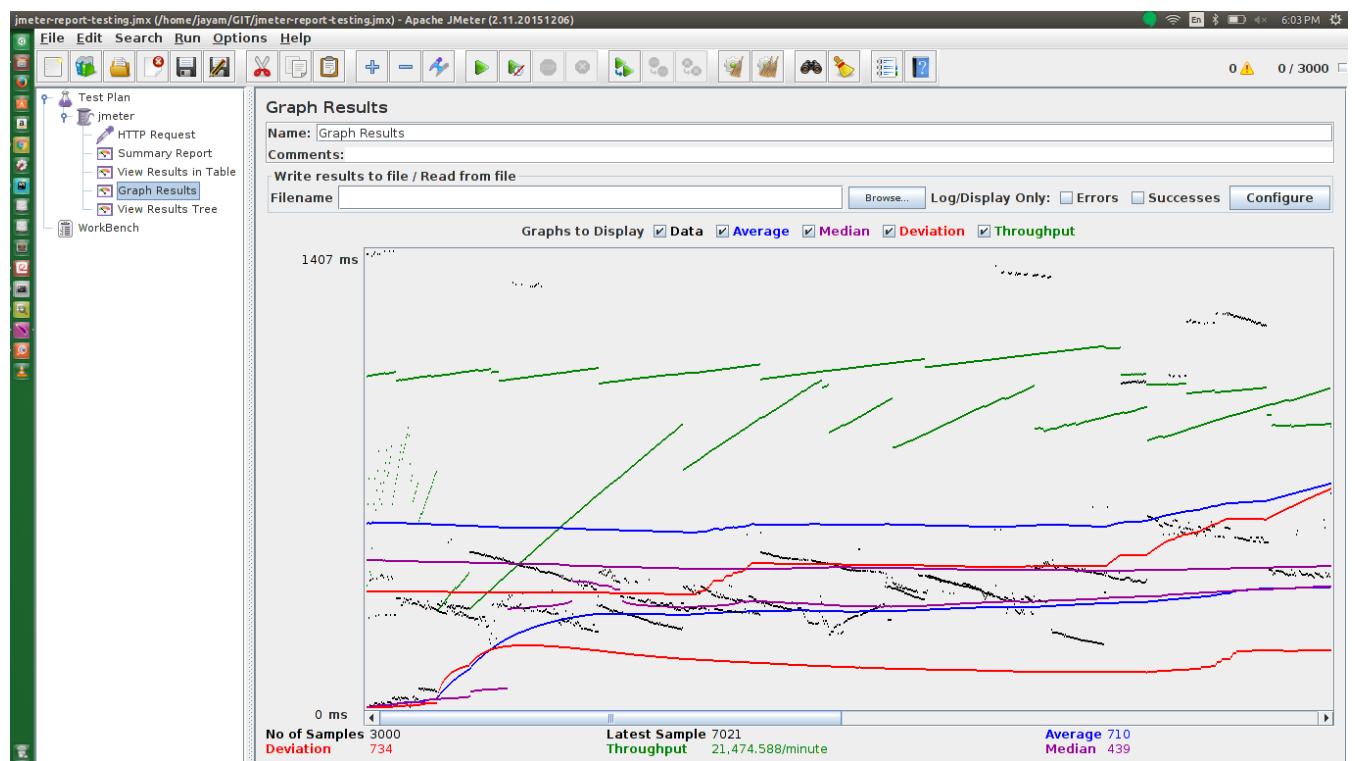
- **1000 Requests** - Base + Connection Pooling + RabbitMQ + Redis

Throughput / minute - 35108.25

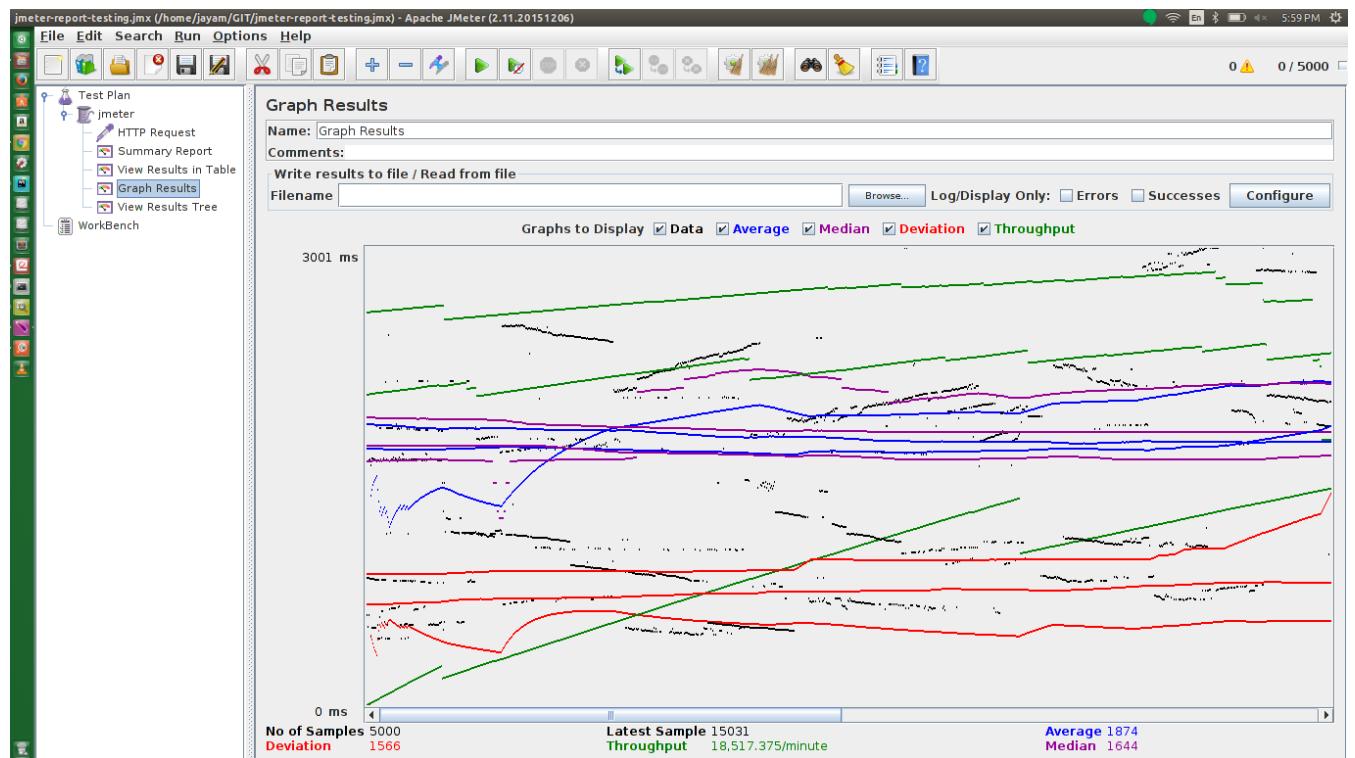
Average – 160



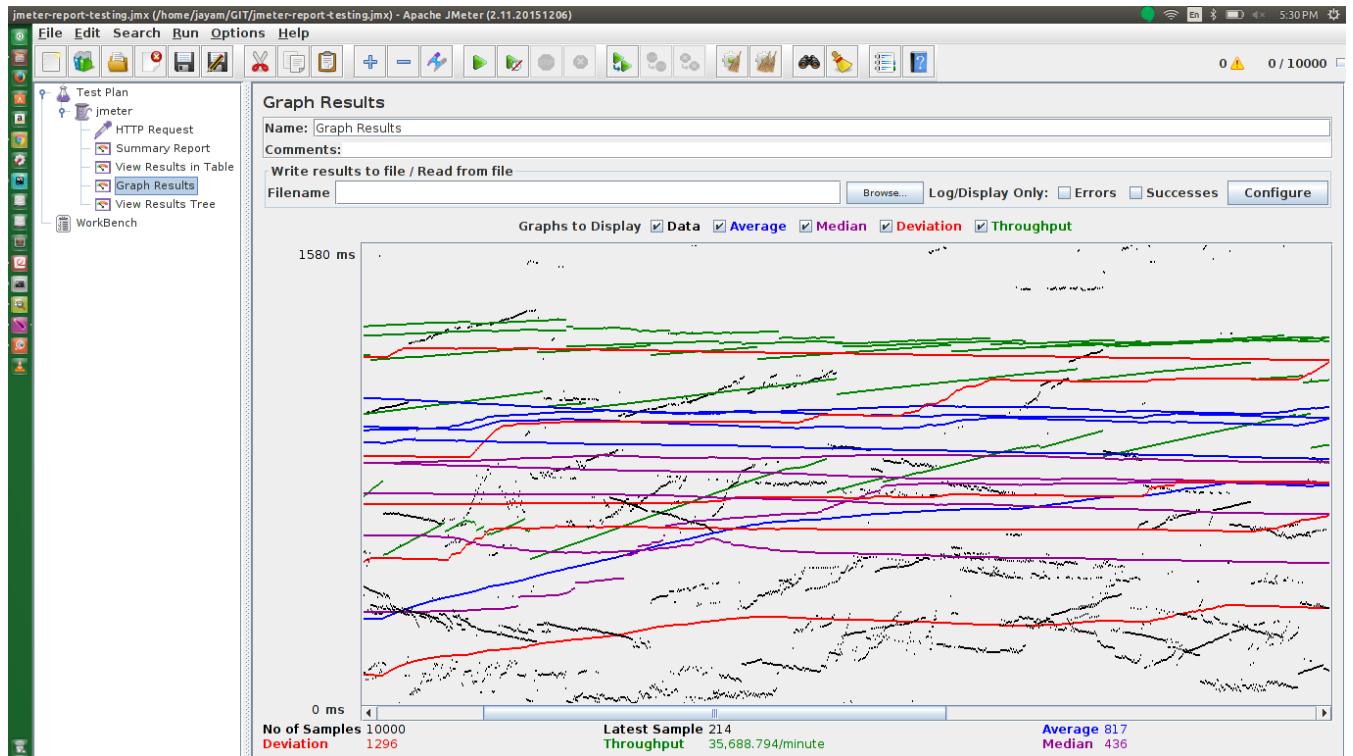
- **3000 Requests** - Base + Connection Pooling + RabbitMQ + Redis
 Throughput / minute - 21474.588 Average – 710



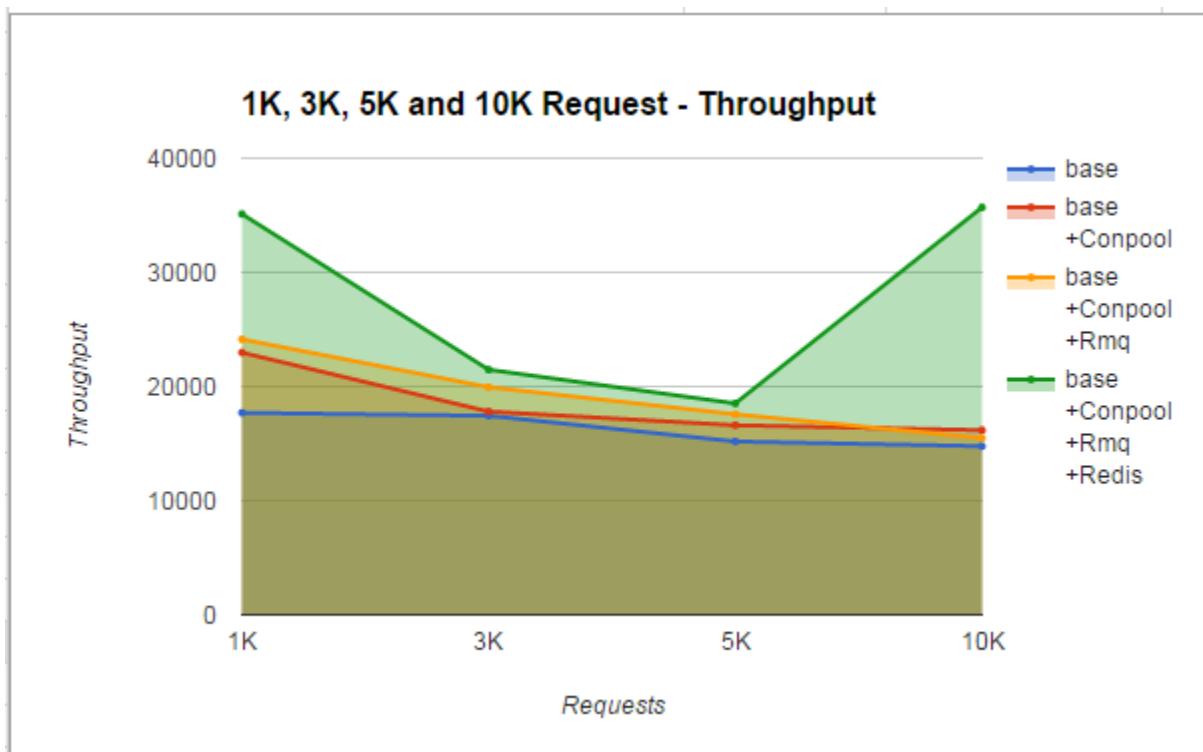
- **5000 Requests** - Base + Connection Pooling + RabbitMQ + Redis
Throughput / minute - 18517.375 Average – 1874



- **10000 Requests** - Base + Connection Pooling + RabbitMQ + Redis
 Throughput / minute - 35688.794 Average – 817



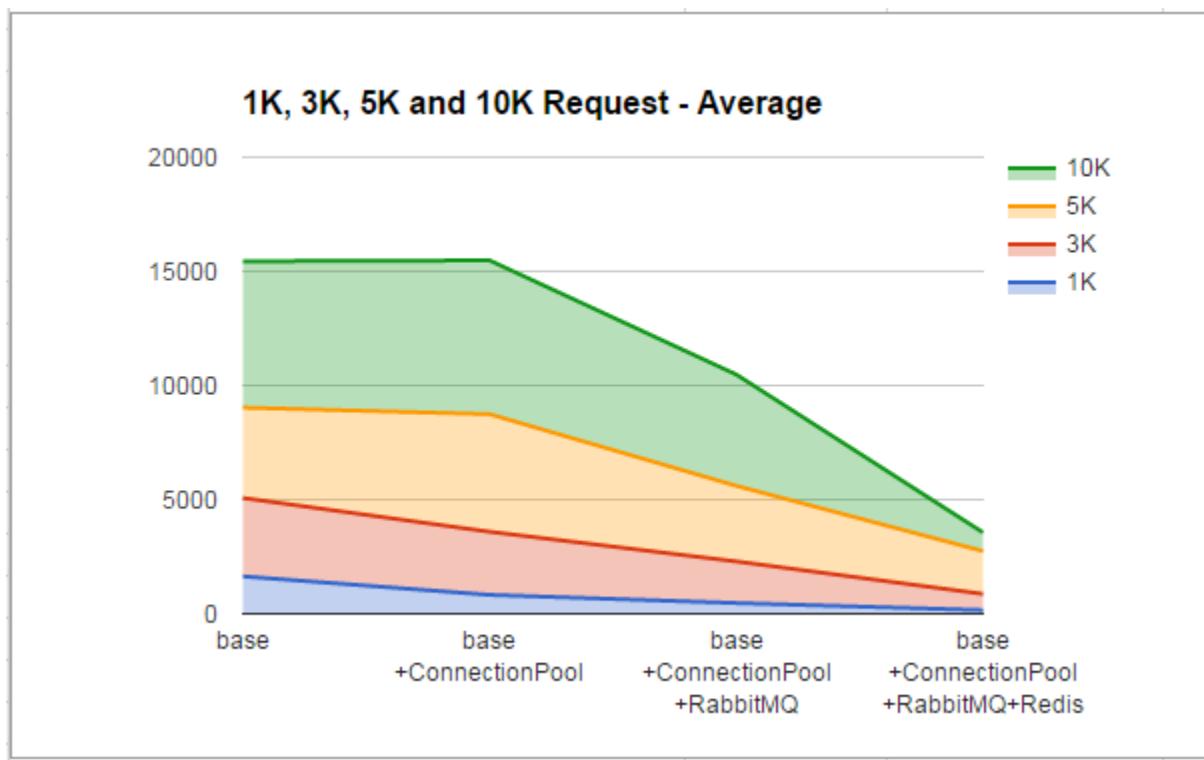
❖ Comparison of Throughput gain after adding features



As we can observe from the above graph, we have improved our applications overall throughput after adding performance gaining features like database connection pooling, RabbitMQ and Redis. Among these we have found to gain maximum throughput when using RabbitMQ and Redis in conjunction. RabbitMQ gives us ability to handle huge number of requests by pushing them in to asynchronous message queues, this gives us ability to absorb huge traffic of incoming requests and still respond to each without losing any request. On top of this we have used Redis, Redis gives us this unique ability to cache data in memory. Redis is an in memory database, this makes it an ideal candidate to cache resources effectively. This is a great advantage for huge

traffic and load bearing servers. With these unique abilities technologies working in harmony with each other, we have been able to achieve this performance.

Comparison of Average time after adding features



❖ **Mocha Testing**

```
cmd C:\Windows\system32\cmd.exe
fixedPrice: 123,
title: 'Paradise Centre',
description: 'Beautiful Location',
toDate: '2017-01-31T08:00:00.000Z',
fromDate: '2016-11-30T08:00:00.000Z',
auctionPrice: null,
city: 'San Jose',
longitude: '-121.8944445',
latitude: '37.3403981',
listId: '769-40-8913',
status: 'rejected',
hostId: 'katy@gmail.com',
__v: 0,
review: [],
created_at: '2016-11-25T08:00:00.000Z',
listingImages: [ [Object] ],
facilities: [ 'Laundry - dryer' ],
safetyAmmneties: [ 'WifSmoke Detector' ],
basicAmmneties: [ 'TV', 'Heat' ],
address: [ [Object] ] ]
Got all the rejected listings
POST /adminGetAllRejectedListings 200 10.515 ms - 988
[ { _id: '583b28946f6cab8e4ac14518',
  hostName: 'Katy',
  hostProfilePic: 'https://s3.amazonaws.com/cmpe273airbnb/Katy.jpg147985544888
9',
  guestAllowed: 2,
  bathCount: 1,
  roomType: 'Entire Place',
  fixedPrice: 123,
  title: 'Paradise Centre',
  description: 'Beautiful Location',
  toDate: '2017-01-31T08:00:00.000Z',
  fromDate: '2016-11-30T08:00:00.000Z',
  auctionPrice: null,
  city: 'San Jose',
  longitude: '-121.8944445',
  latitude: '37.3403981',
  listId: '769-40-8913',
  status: 'rejected',
  hostId: 'katy@gmail.com',
  __v: 0,
  review: [],
  created_at: '2016-11-25T08:00:00.000Z',
  listingImages: [ [Object] ],
  facilities: [ 'Laundry - dryer' ],
  safetyAmmneties: [ 'WifSmoke Detector' ],
  basicAmmneties: [ 'TV', 'Heat' ],
  address: [ [Object] ] } ]
  ✓ Admin: should get all rejected listings

  10 passing (2s)

Clean Redis
C:\Projects\273 Project 20161204\--AirBnb--\airbnb-client\node_modules\.bin>
```

Dynamic Pricing:

Dynamic Pricing - Dynamic pricing will be applied using four parameters.

- 1: Property Clicks**
- 2: User Reviews**
- 3: User Rating**
- 4: Property Rating**
- 5: Festival Days Special Offer**

Our first most prominent parameter for doing dynamic pricing for a property will be the property popularity it self, the property that has more number of clicks will be considered will have more impact for dynamic pricing. On top of that we will consider the user reviews for the property, the property that is more user favourable will be more costlier as compared to rest. After that we will consider user rating of the user having listing, if the user has more rating than his property will have more affinity for costlier price & vice versa. After that we will consider the property rating, the property having more ratings will be costlier & vice versa. After all these parameters we will also consider the Festival days, these will be the final dynamic part. We will apply this model for dynamic pricing.

❖ **Ionic Screenshots**



Welcome to Airbnb

Conitnue with facebook

Create Account

Log In



Welcome to Airbnb

Conitnue with facebook

Create Account

Log In

Android

The screenshot shows a mobile application interface for a travel or vacation booking platform. At the top, there is a navigation bar with a back arrow icon on the left and four tabs on the right: HOME (highlighted with a blue underline), TRIPS, HOST, and PROFILE. Below the navigation bar is a red header bar with the text "Featured Destinations". Underneath this header, there are four square cards, each featuring a scenic sunset image of the Eiffel Tower and a caption identifying the destination. The cards are arranged in a 2x2 grid. The top-left card is labeled "San Jose", the top-right is "Los Angeles", the bottom-left is "London", and the bottom-right is "Beverly Hills".

- 

San Jose
- 

Los Angeles
- 

London
- 

Beverly Hills

Lets's Signup

Firstname

Lastname

Email

Password

Login

Log In

Invalid Password

mak@gmail.com

.....

Login

Log In

Email id not found, Please register

makasdsad@gmail.com

.....

Login

Check In date

mm/dd/yyyy

Check Out date

mm/dd/yyyy

Guest

Room type



Entire Home



Private Room



Shared Room

Filter

Rentals •

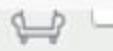




Entire Home



Private Room



Shared Room

Filter

Rentals •

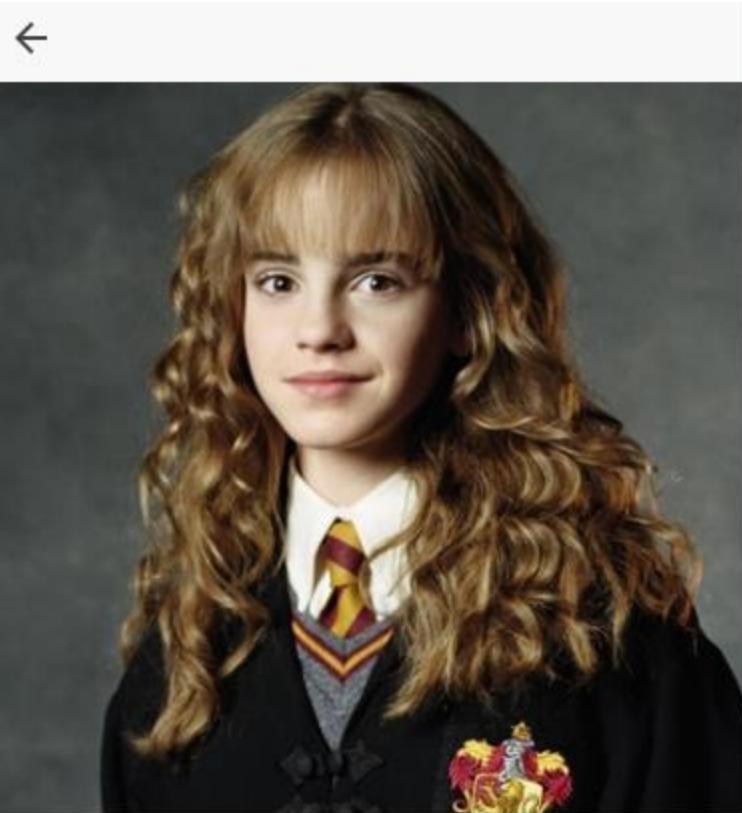


\$123 ⚡

Harry Potter Wizard

Entire Place • 1 guests •





Harry Potter Wizard

London, GB, England • ★★★★☆ 10 reviews



Shaktiman





Entire
Place



1 Guests



2
Bedroom



2 Beds



100% refundable

Cancel up to 30 days before your
trip and get a full refund, including
service fees.

About this listing

Wizardly Magically true

Accommodates: **1**

Available From: **2016-12-09**

Bathrooms: **1**

Available Till:

2017-03-31

Bedrooms: **2**

Property type: **Apartment**

Beds: **2**

Room type: **Entire Place**

Amenities

Essentials

Wifi

Shampoo

Closet/drawers

TV

Linear

TV

Heat

Air Conditioning

Smoke Detector

Air Conditioning

First Aid and Kit

Prices

Extra people: **\$123 / night**

after the first day

\$123

Check In

mm/dd/yyyy

Check Out

mm/dd/yy

Guests



Book

You would not be charged yet

Code Listing:

Database:

The screenshot shows a MongoDB interface with a left sidebar and a main query results area.

Left Sidebar:

- New Connection (3)
 - System
 - airbnb
 - Collections (5)
 - analysis
 - listings
 - sessions
 - userprofiles
 - users
 - Indexes
 - Functions
 - Users
 - foodCoin

Main Area:

Query: db.getCollection('listings').find()

Results:

Key	Value	Type
(2)	ObjectId("582fbded12856b80a437... { 20 fields }	Object
(3)	ObjectId("582fc029f854998203eb... { 20 fields }	Object
(4)	ObjectId("582fc0aa02b62ef8d9a28... { 21 fields }	Object
(5)	ObjectId("583013a795acc9c33c89... { 21 fields }	Object
(6)	ObjectId("5832c483dcce8dc281d1... { 20 fields }	Object
(7)	ObjectId("5833945b23783f6d991... { 21 fields }	Object
(8)	ObjectId("583395903714476e703... { 21 fields }	Object
(9)	ObjectId("583395d8f1924c6f33b8... { 21 fields }	Object
(10)	ObjectId("583396c7f4163c6f405... { 21 fields }	Object
(11)	ObjectId("5833e4d3f4163c6f405... { 21 fields }	Object
(12)	ObjectId("5833f346f4163c6f4058... { 21 fields }	Object
(13)	ObjectId("5833f6d5f4163c6f4058... { 21 fields }	Object
(14)	ObjectId("583422caf4163c6f405... { 21 fields }	Object
(15)	ObjectId("58342546f4163c6f405... { 22 fields }	Object
(16)	ObjectId("5834356f4b5fb87b3e2... { 23 fields }	Object
(17)	ObjectId("5834ce2f767ad8c96be... { 23 fields }	Object
(18)	ObjectId("5834d1e0917336cef5... { 23 fields }	Object
(19)	ObjectId("5834d42799b186d0f4f... { 25 fields }	Object
(20)	ObjectId("5834d5e09316fbdb2b6... { 24 fields }	Object
(21)	ObjectId("5839189eaed22d8d3f5... { 25 fields }	Object
(22)	ObjectId("58391ac6aed22d8d3f5... { 26 fields }	Object
(23)	ObjectId("5840b8fc7f87b879112... { 27 fields }	Object
(24)	ObjectId("5840c6ed7450258691... { 27 fields }	Object
(25)	ObjectId("58413e3e16f94d5a44d... { 27 fields }	Object
(26)	ObjectId("5841c2a0737f2480aa3... { 26 fields }	Object

The screenshot shows the MySQL Workbench interface. On the left, the sidebar includes sections for Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore, INSTANCE, Performance, and Schemas. The Schemas section shows the 'airbnb' database selected, containing Tables, Views, Stored Procedures, and Functions. The sys schema is also listed. At the bottom of the sidebar are Object Info and Session tabs.

The main area contains a Query Editor titled "SQL File 1" with the following SQL code:

```

1 • use airbnb;
2 • select * from trips;
3 • select * from trips where userEmail = 'shaktiman@gmail.com' AND

```

Below the query editor is a Result Grid showing the output of the first two SELECT statements. The columns are tripId, tripsequen..., userEmail, hostEmail, listId, and fixedPrice. The data includes rows for various trips, mostly from katy@gmail.com and shaktiman@gmail.com.

tripId	tripsequen...	userEmail	hostEmail	listId	fixedPrice
507-72-8526	9	katy@gmail.com	katy@gmail.com	444-48-5756	345
507-37-6021	10	katy@gmail.com	katy@gmail.com	530-81-1332	170
002-01-8602	11	shaktiman@gmail.com	krist@gmail.com	155-66-3862	245
044-60-7131	12	shaktiman@gmail.com	rock@gmail.com	003-32-9377	125
007-38-1824	13	katy@gmail.com	krist@gmail.com	155-66-3862	245
423-01-1031	14	katy@gmail.com	krist@gmail.com	155-66-3862	245
652-03-3156	15	krist@gmail.com	shaktiman@gmail.com	228-53-0418	123
750-03-2255	16		katy@gmail.com	769-40-8913	123
NULL	NULL	NULL	NULL	NULL	NULL

Below the result grid is an Action Output table showing the execution details for each statement:

Action	Time	Ac...	Response	Duration / Fetch Time
1	23:50:06	sel...	Error Code: 1046. No database selected Select the de...	0.058 sec
2	23:50:09	us...	0 row(s) affected	0.00042 sec
3	23:50:11	sel...	8 row(s) returned	0.00062 sec / 0.000...

Code Listing: Server Side:

```
www scheduler.js server.js
in the editListing_queue queue
in the isLoggedIn_queue queue
in the analysis_queue
in the getAllApprovedListings_queue queue
in the analysis_queue
in the getPaymentDetails_queue queue
in the getAllRejectedListings_queue queue
in the deleteAccount_queue
in the user_track
in the bidding_track_queue
in the analysis_queue
in the analysis_queue
in the getBillfromBillDate_queue queue
in the getUserfromCity_queue queue
in the getReviews_queue queue
in the total_revenue_queue
in the updateAuctionTrips_queue
in the placeBid_queue
in the getAllAuctionableProperties_queue
in the user_track
in the getAllProperties_queue
in the allCities_queue
in the total_listings_queue
in the checkDates_queue
in the getProfile_queue queue
in the bookTrip_queue
in the total_revenue_queue
in the getPendingTrips_queue queue
in the approveRequests_queue queue
in the editProfile_queue queue
in the paymentDetails_queue queue
in the pendingRequests_queue queue
in the approvedRequests_queue queue
in the completedTripsForUser_queue queue
in the rejectRequests_queue queue
in the reviewUser_queue queue
in the getAuctionTrips_queue queue
in the completedTripsForHost_queue queue
in the yourReservations_shows_shows
```

Express Session:

sion]

```

var uploadMyVideo;
//var imageProfileColl=[];

var mongoStore = require("connect-mongo")(expressSession);

var mongoSessionConnectURL = "mongodb://localhost:27017/airbnb";

var app = express();
var db = require('./models/db');
require('./routes/passport')(passport);

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

//express-session

app.use(expressSession({
  secret: 'cmpe273_teststring',
  resave: false, //don't save session if unmodified
  saveUninitialized: false, // don't create session until something stored
  duration: 30 * 60 * 1000,
  activeDuration: 5 * 60 * 1000,
  store: new mongoStore({
    url: mongoSessionConnectURL
  })
}));

//express-session
// app.use(expressSession({

```

Code Listing for client side:

--Airbnb-- /Volumes/Personal/github2/Airbnb-client

Project

```

var mq_client = require('../rpc/client');

//for redis-----
var redis = require('redis');
var redisClient = redis.createClient(6379, '127.0.0.1');
//---redis end---

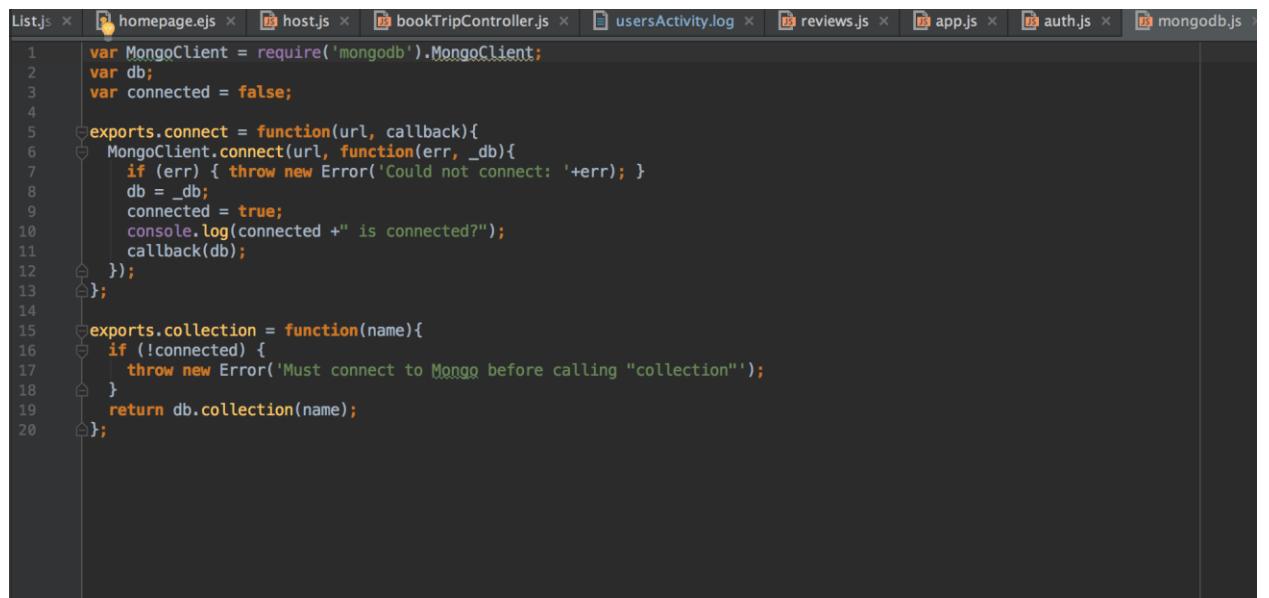
function signin(req,res ,next)
{
  passport.authenticate('local-login', function(err, user, info) {
    console.log(" I am in login authenticate");
    if(err)
    {
      console.log("printing err ERROR ERROR ERROR ERROR");
      if(err=="Timeout")
      {
        return res.send(err);
      }
      return next(err);
    }

    console.log("printing the info message",info);
    if(!user && info.message=="notRegistered")
    {
      console.log("user not found notRegistered");
      return res.send({msg:"404"});
    }
    if(!user && info.message=="IncorrectPassword") {
      console.log("user not found IncorrectPassword");
      return res.send({msg:"IncorrectPassword"});
    }
  });
}

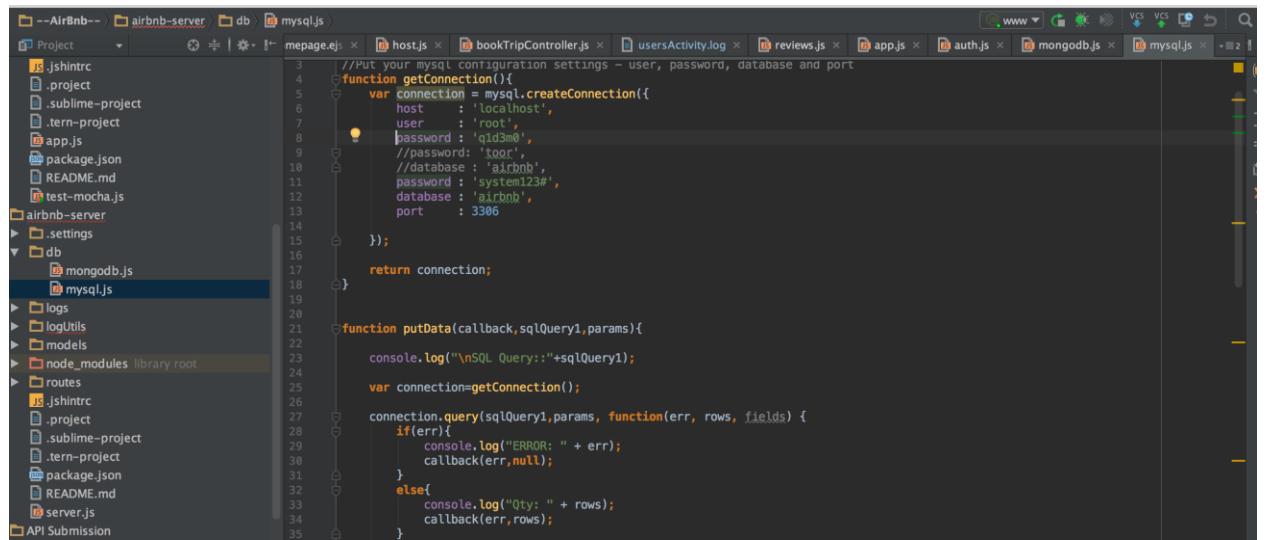
Run: www scheduler.js server.js

```

Database access classes:



```
1 var MongoClient = require('mongodb').MongoClient;
2 var db;
3 var connected = false;
4
5 exports.connect = function(url, callback){
6   MongoClient.connect(url, function(err, _db){
7     if (err) { throw new Error('Could not connect: ' + err); }
8     db = _db;
9     connected = true;
10    console.log(connected + " is connected?");
11    callback(db);
12  });
13 }
14
15 exports.collection = function(name){
16   if (!connected) {
17     throw new Error('Must connect to Mongo before calling "collection"');
18   }
19   return db.collection(name);
20 }
```



```
3 //Put your mysql configuration settings - user, password, database and port
4 function getConnection(){
5   var connection = mysql.createConnection({
6     host : 'localhost',
7     user : 'root',
8     password : 'q1d3m@',
9     //password: '0001',
10    //database : 'airbnb',
11    password : 'system123#',
12    database : 'airbnb',
13    port : 3306
14  });
15
16  return connection;
17
18 }
19
20
21 function putData(callback,sqlQuery1,params){
22
23   console.log("\nSQL Query::"+sqlQuery1);
24
25   var connection=getConnection();
26
27   connection.query(sqlQuery1,params, function(err, rows, fields) {
28     if(err){
29       console.log("ERROR: " + err);
30       callback(err,null);
31     }
32     else{
33       console.log("Qty: " + rows);
34       callback(null,rows);
35     }
36   })
37 }
```

```
function yourListings(req,callback){
  console.log('In your listings queue' + req.username);
  Listing.find({hostId: req.username}, function(err, result){
    if(err){
      throw err;
    }else{
      console.log(result);
      if(result.length > 0){
        callback(null, {'code': '200', 'value': result});
      }else{
        callback(null, {'code': '201'});
      }
    }
  });
}

function deleteListing(req,callback){
  console.log('In delete listings queue' + req.id);
  Listing.update({listId: req.id}, { $set: {"status": req.status} }, function(err, result){
    if(err){
      throw err;
    }else{
      if(result != ""){
        callback(null, {'code': '200'});
      }
    }
  });
}

function pendingRequests(req,callback){
  console.log('In pending requests queue' + req.email);
}
```

Lessons Learned

The System has taught us in great detail on how the Enterprise solutions should be architected in order to create a large scalable and maintainable web application. We have created RESTful stateless service as a uniform communication API for requesting & sending resource. At the database level connection polling helped us in gaining better throughput by maintaining already created connections. We have also learnt that AMQP protocol used in our application via RabbitMQ can be used as a asynchronous message passing mechanism. This gives us a great deal of scalability. On top of that applied with Redis, will make a web application even quicker by caching and sharing the resources from in memory cache.

