

Lab2

GitHub Link:- https://github.com/ShreyBhatt/CMPE273_Lab2

Ebay with MongoDB and RabbitMQ

Introduction

The Ebay Marketplace Application is the replica of Ebay used to demonstrate REST web services using RabbitMQ as message broker and MongoDB as the database.

Goal & Purpose

- Goal of this application is to demonstrate REST web services and message oriented designing.

Basic Functionalities

1. Basic User functionalities:
 - a. Sign up the new users (First name, Last name, Email, Password, Birthdate, Location, Contact)
 - b. Sign in with existing users
 - c. Sign out
2. Profile:
About: Birthday, Ebay handle, contact information and location
3. User's advertisements for others to read. Which includes the item name, item description, seller information, item price and quantity
4. Selling Functionality
5. Bidding Functionality
6. Shopping Cart functionality
7. User tracking functionality
8. Credit card validations on payment.
9. Order History Functionality

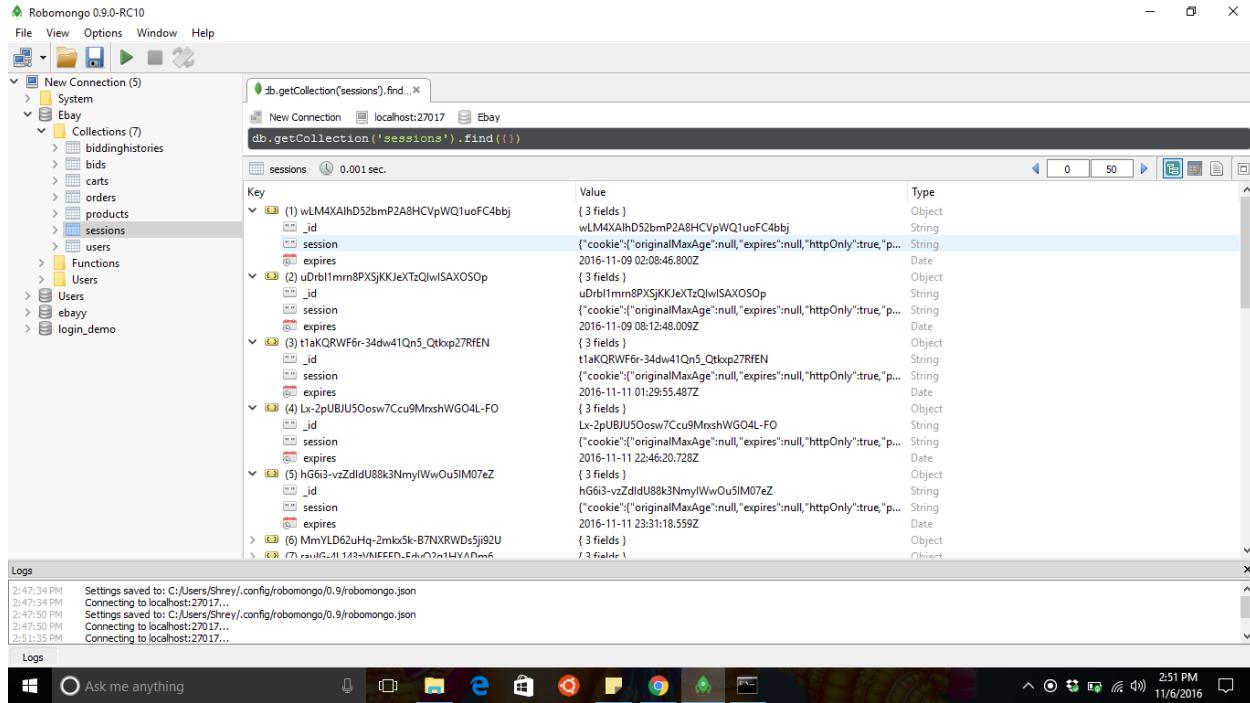
System Design

There are 14 pages in the entire application

1. Main page which will show the login and signup options.
2. Home page which will be shown only after successful login. It will display All Advertisements which are posted by other users.
3. All bids page
4. Profile page which shows further details of the user.
5. Advertisement posting page
6. Bid posting page
7. Product Detail page
8. Bid Product Detail Page
9. My advertisement page which will show advertisement posted by us
10. My bids page which will show bid posted by us
11. Shopping cart page
12. Last Login page
13. My orders page
14. Credit card Validation Page on checkout

The header contains a navigation bar which has options of shopping cart, create account and signout.

- Front End: Angular.js, Bootstrap, HTML
- Back End: Node.js
- Database: MongoDB
- Message Queue Broker: RabbitMQ
- Authentication: Passport
- Sessions are to be stored in MongoDB.



Client-Side:

```

var session = require('client-sessions');
var mongoSessionConnectURL = "mongodb://localhost:27017/Ebay";
var expressSession = require("express-session");
var mongoStore = require("connect-mongo")(expressSession);

```

```

var app = express();

app.use(expressSession({
  secret: 'Ebay',
  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
  })
)));

```

Server-Side:

```
var mongoose = require('mongoose');

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

mongoose.connect(mongoSessionConnectURL, {server: {poolSize: 10}});

mongoose.connection.on('connected', function() {
  console.log('Mongoose connected to ' + mongoSessionConnectURL);
});

mongoose.connection.on('error', function(err) {
  console.log('Mongoose connection error: ' + err);
});

mongoose.connection.on('disconnected', function() {
  console.log('Mongoose disconnected');
});

gracefulShutdown = function(msg, callback) {
  mongoose.connection.close(function() {
    console.log('Mongoose disconnected through ' + msg);
    callback();
  });
};

process.once('SIGUSR2', function() {
  gracefulShutdown('nodemon restart', function() {
    process.kill(process.pid, 'SIGUSR2');
  });
});

process.on('SIGINT', function() {
  gracefulShutdown('app termination', function() {
    process.exit(0);
  });
});
```

Client-side

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebony > Ebony_Client > app.js
Project Structure
app.js services\login.js user.js routes\login.js
app.js
44
45
46 var session = require('client-sessions');
47 var mongoSessionConnectURL = "mongodb://localhost:27017/Ebony";
48 var expressSession = require("express-session");
49 var mongoStore = require("connect-mongo")(expressSession);
50
51 // mongoose.connect(mongoSessionConnectURL, {server: {poolSize: 100}});
52
53
54
55 var app = express();
56
57 app.use(expressSession({
58   secret: 'Ebony',
59   resave: false, //don't save session if unmodified
60   saveUninitialized: false, // don't create session until something stored
61   duration: 30 * 60 * 1000,
62   activeDuration: 5 * 60 * 1000,
63   store: new mongoStore({
64     url: mongoSessionConnectURL
65   })
66 }));
67
68
69

```

Server-side

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebony > Ebony_Server > server.js
Project Structure
app.js server.js services\login.js passport.js user.js routes\login.js bid.js routes\biddinglogic.js
server.js
35
36
37 var mongoose = require('mongoose');
38
39 var mongoSessionConnectURL = "mongodb://localhost:27017/Ebony";
40
41 mongoose.connect(mongoSessionConnectURL, {server: {poolSize: 10}});
42
43 mongoose.connection.on('connected', function() {
44   console.log('Mongoose connected to ' + mongoSessionConnectURL);
45 });
46
47 mongoose.connection.on('error', function(err) {
48   console.log('Mongoose connection error: ' + err);
49 });
50
51 mongoose.connection.on('disconnected', function() {
52   console.log('Mongoose disconnected');
53 });
54
55
56
57 process.once('SIGUSR2', function() {
58   gracefulShutdown('nodemon restart', function() {
59     process.kill(process.pid, 'SIGUSR2');
60   });
61 });
62
63
64 process.on('SIGINT', function() {
65   gracefulShutdown('app termination', function() {
66     process.exit(0);
67   });
68 });

```

- Using Passport as Authentication

```

var passport = require('passport');
var LocalStrategy = require('passport-local').Strategy;
//var flash = require('connect-flash');
var bcrypt = require('bcrypt-nodejs');
var mq_client = require('../rpc/client');

module.exports = function (passport) {
  passport.use('login', new LocalStrategy({
    usernameField: 'username',
    passwordField: 'password'
  }, function (username, password, done) {

    var msg_payload = {"username":username,"password":password}
    mq_client.make_request('login_queue',msg_payload, function(err,results) {
      if (err)
      {
        return done(err);
      }
      if(results)
      {
        console.log(results);
        done(null, results);
      }
    });
  }));
};

});
```

The screenshot shows the WebStorm IDE interface with the following details:

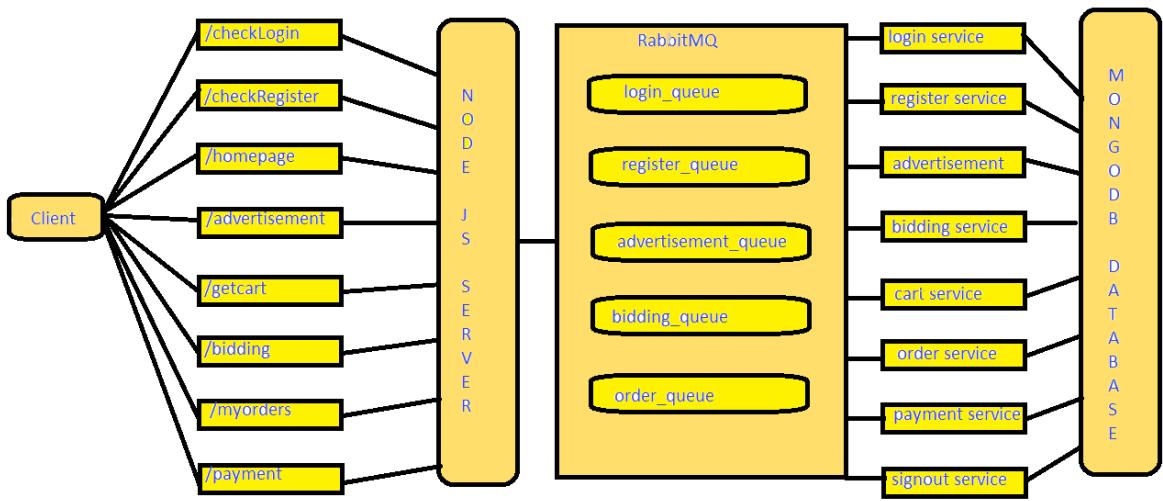
- Project Structure:** Shows the project hierarchy with files like app.js, services/login.js, passport.js, user.js, and various routes files (advertisement.js, bid.js, biddinglog.js, biddinglogic.js, bidproductdetail.js, creditcard.js, home.js, index.js, lastlogin.js, log.js, login.js, myebaybutton.js, myorders.js, payment.js, productdetail.js, profile.js, register.js, shoppingcart.js, users.js).
- Code Editor:** The login.js file is open, showing the implementation of the passport.authenticate middleware. It logs errors, checks for a successful login (code 200), renders a template, sets session variables, and returns a response.
- Toolbars and Status Bar:** Standard WebStorm toolbars and status bar at the bottom showing the current file path, encoding, and system information.

Ebay Server

The screenshot shows the WebStorm IDE interface with a terminal window open. The terminal window displays logs from a RabbitMQ application. The logs show various queues being listened to and messages being received, including a delivery info message. The terminal window has a title bar "Event Log" and a status bar at the bottom showing "7:1 CRLF: UTF-8" and the date/time "11/3/2016 11:30 PM".

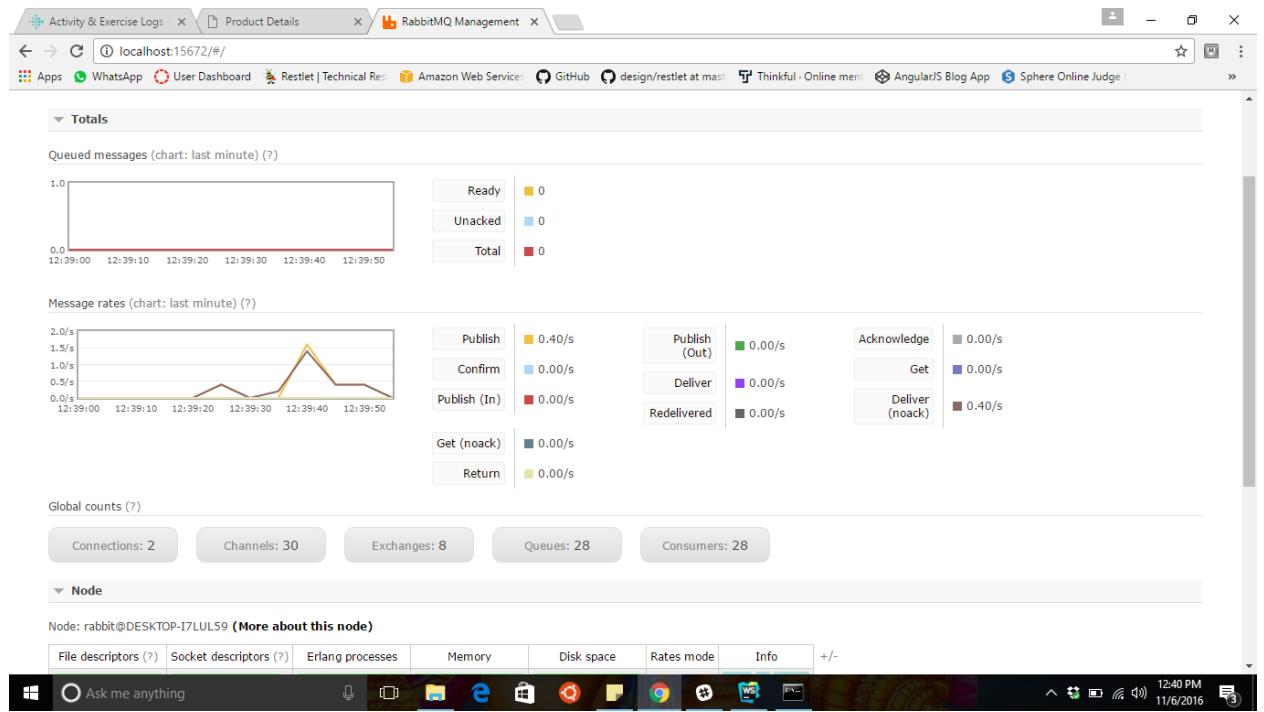
```
"C:\Program Files (x86)\JetBrains\WebStorm 2016.2.2\bin\runnerw.exe" "C:\Program Files\nodejs\node.exe" C:\Users\Shrey\WebstormProjects\Ebay\Ebay_Server\server.js
listening on login_queue
listening on register_queue
listening on checkUsername_queue
listening on checkEmail_queue
listening on signout_queue
listening on advertisement_queue
listening on displayadvertisel_queue
listening on displayalladvertisel_queue
listening on lastlogin_queue
listening on profile_queue
listening on productdetail_queue
listening on addcart_queue
listening on bid_queue
listening on displaybid_queue
listening on displayallbid_queue
listening on bidproductdetail_queue
listening on bidding_queue
listening on biddinghistory_queue
listening on getcart_queue
listening on remove_queue
listening on displayorders_queue
listening on emptycart_queue
listening on orderentry_queue
listening on bidorderentry_queue
listening on findmaxbidprice_queue
listening on biddinglogic_queue
listening on closethebid_queue
3 Nov 23:30:11 - biddinglogic_queue { isBidEnded: 0 }
3 Nov 23:30:11 - Message: { "isBidEnded":0 }
3 Nov 23:30:11 - DeliveryInfo: { "contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "7795155b6ef843add01b46664c6c0b49", "replyTo": "amq.gen-Kqbw2cMAUHwzrH-o
In handle request_of biddinglogic
```

Architecture of RabbitMQ Interaction With The Ebay Application



RabbitMQ Management Console

Overview

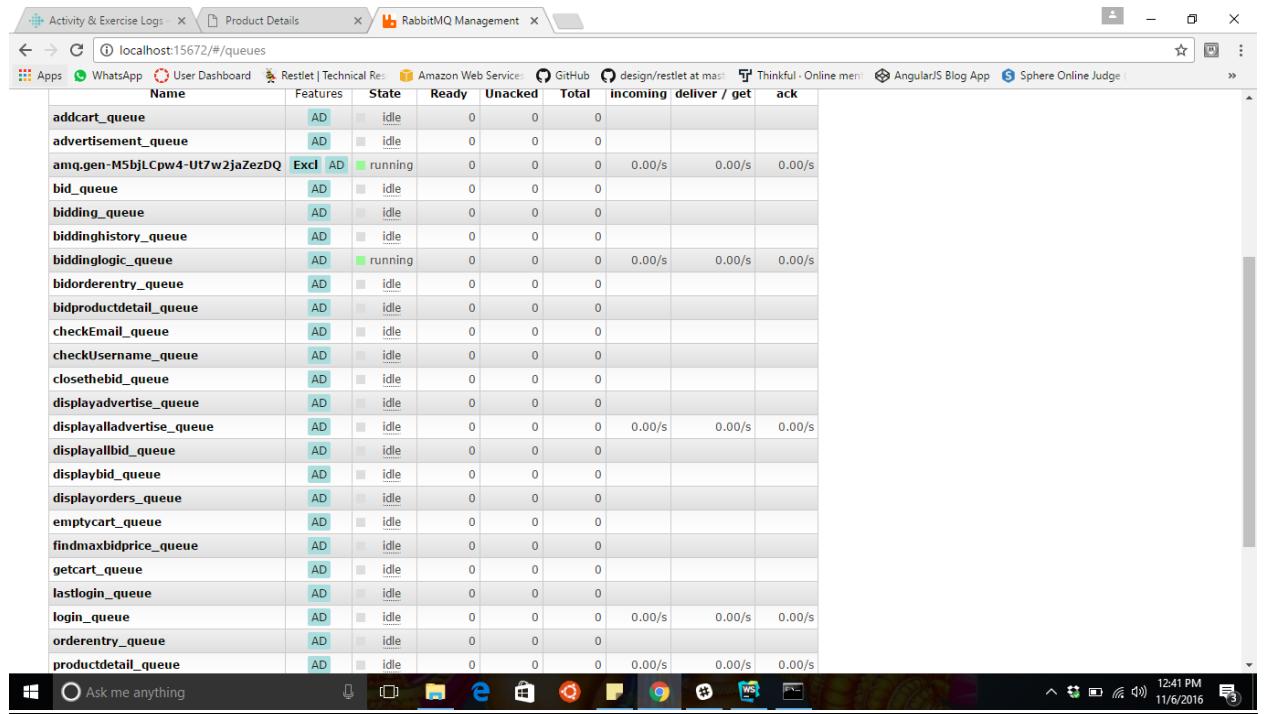


Channels

The screenshot shows the RabbitMQ Management interface at the URL localhost:15672/#/channels. The main area is titled "Channels". It displays a table of open channels:

Channel	User name	Mode (?)	State	Unconfirmed	Prefetch (?)	Unacked	publish	confirm	deliver / get	ack
127.0.0.1:57660 (1)	guest	running	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57660 (2)	guest	running	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (1)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (10)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (11)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (12)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (13)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (14)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (15)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (16)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (17)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (18)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (19)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (2)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (20)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (21)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (22)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (23)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (24)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (25)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (26)	guest	running	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (27)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (28)	guest	running	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s
127.0.0.1:57662 (3)	guest	idle	0	0	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s	0.00/s

Queues

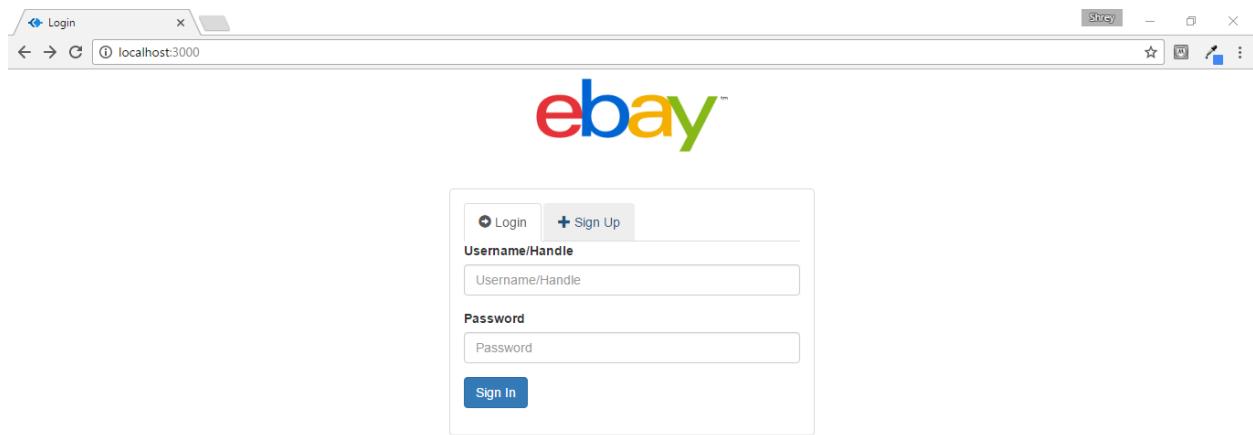


The screenshot shows a web browser window titled "RabbitMQ Management" at the URL "localhost:15672/#/queues". The page displays a table of queues with the following columns: Name, Features, State, Ready, Unacked, Total, incoming, deliver / get, and ack. The "Features" column includes icons for AD (Advanced Distribution), RD (Replicated Distribution), and EX (Exclusive Distribution). The "State" column indicates the current state of each queue. The "Total" column shows the total number of messages in the queue. The "incoming" column shows the rate of incoming messages per second. The "deliver / get" column shows the rate of delivered and received messages per second. The "ack" column shows the rate of acknowledged messages per second.

Name	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
addcart_queue	AD	idle	0	0	0			
advertisement_queue	AD	idle	0	0	0			
amq.gen-M5bjLCpw4-Ut7w2jaZezDQ	Excl	AD running	0	0	0	0.00/s	0.00/s	0.00/s
bid_queue	AD	idle	0	0	0			
bidding_queue	AD	idle	0	0	0			
biddinghistory_queue	AD	idle	0	0	0			
biddinglogic_queue	AD	running	0	0	0	0.00/s	0.00/s	0.00/s
bidorderentry_queue	AD	idle	0	0	0			
bidproductdetail_queue	AD	idle	0	0	0			
checkEmail_queue	AD	idle	0	0	0			
checkUsername_queue	AD	idle	0	0	0			
closethebid_queue	AD	idle	0	0	0			
displayadvertise_queue	AD	idle	0	0	0			
displayalladvertise_queue	AD	idle	0	0	0	0.00/s	0.00/s	0.00/s
displayallbid_queue	AD	idle	0	0	0			
displaybid_queue	AD	idle	0	0	0			
displayorders_queue	AD	idle	0	0	0			
emptycart_queue	AD	idle	0	0	0			
findmaxbidprice_queue	AD	idle	0	0	0			
getcart_queue	AD	idle	0	0	0			
lastlogin_queue	AD	idle	0	0	0			
login_queue	AD	idle	0	0	0	0.00/s	0.00/s	0.00/s
orderentry_queue	AD	idle	0	0	0			
productdetail_queue	AD	idle	0	0	0	0.00/s	0.00/s	0.00/s

- Application Screenshots

- **Login Page**
- First page to be displayed after running the application. User Can not use the application without Logging in.



Login Client

```

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\\services\\biddinglogic.js - WebStorm 2016.2.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Run: bin\\www server.js
C:\Program Files (x86)\JetBrains\WebStorm 2016.2.2\bin\runner.exe" "C:\Program Files\nodejs\node.exe" C:\Users\Shrey\WebstormProjects\Ebay\bin\www
  Ebay:server Listening on port 3000 +0ms
    { code: '200',
      data:
        [ { _id: '5819ab6f7fd6571904b4d157',
          lastlogin: 'November 3rd 2016, 9:11:17 pm',
          contact: '12345678901234',
          location: 'San Jose, California',
          birthdate: '2016-11-14T00:00:00.000Z',
          password: '62a6104jNS10z4xf1FkWkr9eSxcLejA7SVAvk/E9imJwo5limMfeWTipuoK',
          email: 'shreykbhatt@gmail.com',
          username: 'shreykbhatt',
          __v: 0 } ]
      Successful Login
      will renderfile
      { code: '200',
        data:
          [ { _id: '5819ab6f7fd6571904b4d157',
            lastlogin: 'November 3rd 2016, 9:11:17 pm',
            contact: '12345678901234',
            location: 'San Jose, California',
            birthdate: '2016-11-14T00:00:00.000Z',
            password: '62a6104jNS10z4xf1FkWkr9eSxcLejA7SVAvk/E9imJwo5limMfeWTipuoK',
            email: 'shreykbhatt@gmail.com',
            username: 'shreykbhatt',
            __v: 0 } ]
        Thursday, November 3, 2016 11:32 PM
        5819ab6f7fd6571904b4d157
        POST /checkLogin 200 90.668 ms - 5611
        GET /css/bootstrap.css 304 8.515 ms - -
        GET /css/font-awesome.css 304 14.809 ms - -
        GET /css/styles.css 304 14.638 ms - -
        GET /css/mystyles.css 304 14.389 ms - -
        GET /angularjs/advertisement.js 304 11.804 ms - -
      }
    }
  }
}

Event Log
1461 CRLF: UTF-8: 11:32 PM 11/3/2016

```

Login Server

```

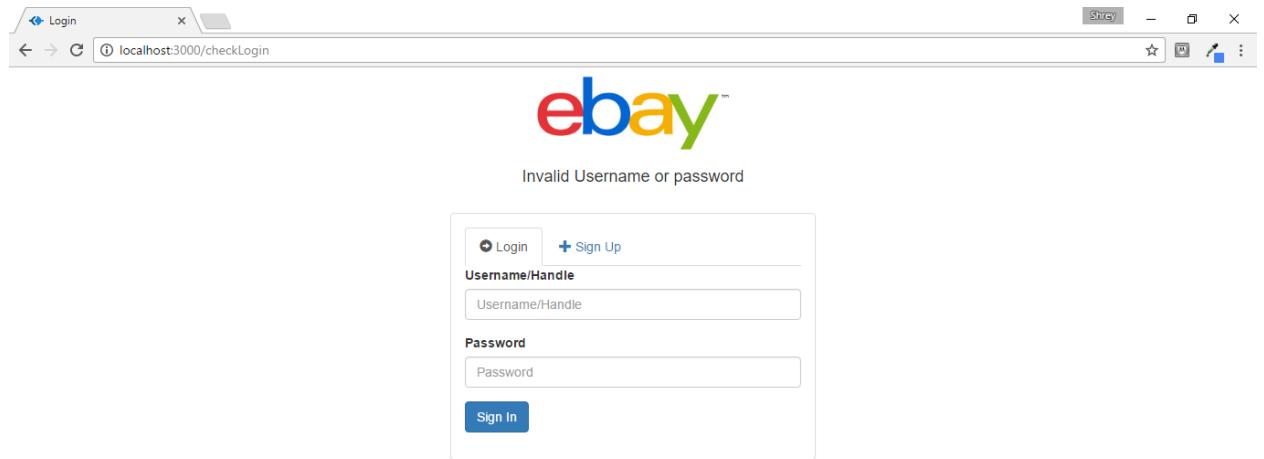
Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\\services\\biddinglogic.js - WebStorm 2016.2.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Run: bin\\www server.js
C:\Program Files (x86)\JetBrains\WebStorm 2016.2.2\bin\runner.exe" "C:\Program Files\nodejs\node.exe" C:\Users\Shrey\WebstormProjects\Ebay\bin\www
  Ebay:server Listening on port 3000 +0ms
    listening on displayallbid_queue
    listening on bidproductdetail_queue
    listening on bidding_queue
    listening on biddinghistory_queue
    listening on getcart_queue
    listening on remove_queue
    listening on displayorders_queue
    listening on emptycart_queue
    listening on orderentry_queue
    listening on bidorderentry_queue
    listening on findmaxbidprice_queue
    listening on biddinglogic_queue
    listening on closethebid_queue
  3 Nov 23:32:03 - login_queue { username: 'shreykbhatt', password: 'test' }
  3 Nov 23:32:03 - Message: { "username": "shreykbhatt", "password": "test" }
  3 Nov 23:32:03 - DeliveryInfo: { "contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "dc4b0646a417c772ef3be0dcc1c36b3a", "replyTo": "amq.gen-D3Wxuf5_vmwCbTjP" }
  In handle request of LOGIN
  { _id: '5819ab6f7fd6571904b4d157',
    lastlogin: 'November 3rd 2016, 9:11:17 pm',
    contact: '12345678901234',
    location: 'San Jose, California',
    birthdate: 'Sun Nov 13 2016 16:00:00 GMT-0800 (Pacific Standard Time)',
    password: '62a6104jNS10z4xf1FkWkr9eSxcLejA7SVAvk/E9imJwo5limMfeWTipuoK',
    email: 'shreykbhatt@gmail.com',
    username: 'shreykbhatt',
    __v: 0 }
  Login successful
  3 Nov 23:32:03 - displayalladvertisel_queue { sellerid: '5819ab6f7fd6571904b4d157' }
  3 Nov 23:32:03 - Message: { "sellerid": "5819ab6f7fd6571904b4d157" }
  3 Nov 23:32:03 - DeliveryInfo: { "contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "0173b4b1ca21ddeff4658894bee14402", "replyTo": "amq.gen-D3Wxuf5_vmwCbTjP" }
  In handle request of Display all advertisement
  All Advertisement Display successful
  |

Event Log
1461 CRLF: UTF-8: 11:32 PM 11/3/2016

```

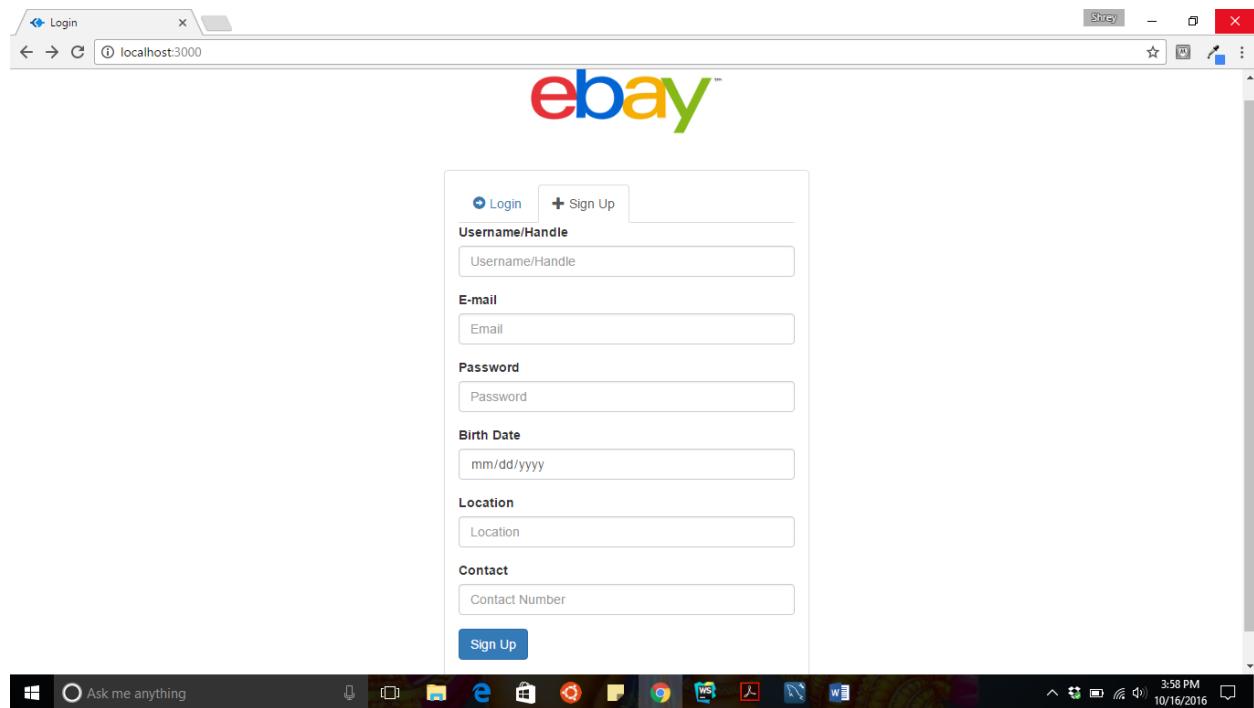
- Invalid Username/password

- If the username/password is not correct it will display “invalid username or password” message.

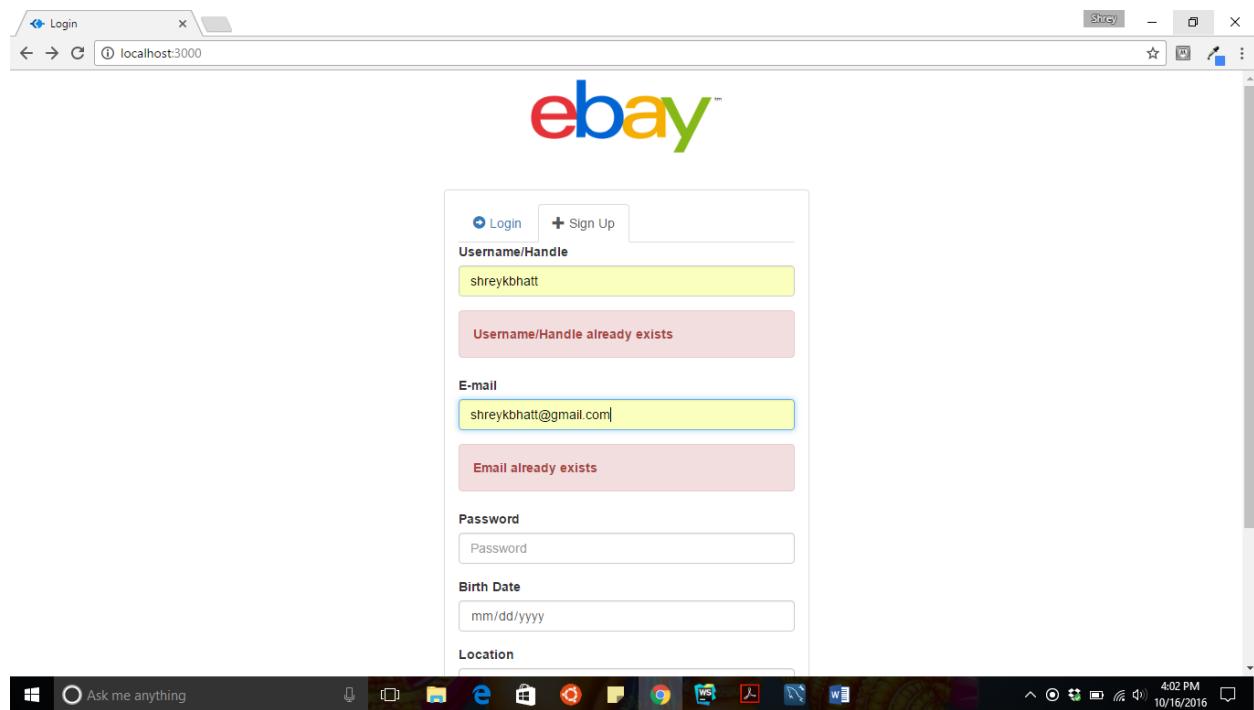


- **SignUp/Register Page**

- Enter Username/Ebay Handle,E-mail,Password,BirthDate,Location and Contact Details.



- If the User Already Exists then It will not allow to add.



Signup Client

The screenshot shows the WebStorm IDE interface with the title bar "Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\services\biddinglogic.js - WebStorm 2016.2.4". The "Run" tab is selected, showing the command "server.js". The main pane displays a log of HTTP requests and responses:

```
{ code: '200' }  
POST /checkUsername 200 16.784 ms - 18  
{ code: '200' }  
POST /checkUsername 200 12.551 ms - 18  
{ code: '200' }  
POST /checkUsername 200 13.550 ms - 18  
{ code: '200' }  
POST /checkUsername 200 16.331 ms - 18  
{ code: '200' }  
POST /checkUsername 200 13.289 ms - 18  
{ code: '200' }  
POST /checkEmail 200 19.641 ms - 18  
{ code: '200' }  
POST /checkEmail 200 14.657 ms - 18  
{ code: '200' }  
POST /checkEmail 200 16.757 ms - 18  
{ code: '200' }  
POST /checkEmail 200 7.726 ms - 18  
{ code: '200' }  
POST /checkEmail 200 19.087 ms - 18  
{ code: '200' }  
POST /checkEmail 200 15.264 ms - 18  
{ code: '200' }  
POST /checkEmail 200 12.043 ms - 18  
{ code: '200' }  
POST /checkEmail 200 10.671 ms - 18  
{ code: '200' }  
POST /checkEmail 200 14.161 ms - 18  
POST /checkRegister 200 18.824 ms - 6952  
GET /css/login.css 304 9.398 ms - -  
GET /angularjs/Login.js 304 8.204 ms - -  
Successful Signup  
GET /img/ebay.png 304 1.671 ms - -
```

The bottom status bar shows "70:1 CRLF: UTF-8" and the date "11/3/2016".

SignUp server

The screenshot shows the WebStorm IDE interface with the title bar "Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\services\biddinglogic.js - WebStorm 2016.2.4". The "Run" tab is selected, showing the command "server.js". The main pane displays a log of messages and errors:

```
3 Nov 23:34:58 - checkEmail_queue ()  
3 Nov 23:34:58 - Message: {}  
3 Nov 23:34:58 - DeliveryInfo: {"contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "0edf8321fc9ee162162efc003719f5b1", "replyTo": "amq.gen-X-GqHkudQDoyW5rx", "In handle request of checkEmail  
data not found  
3 Nov 23:34:58 - checkEmail_queue { email1: 'maitray@gmail.c' }  
3 Nov 23:34:58 - Message: {"email1": "maitray@gmail.c"}  
3 Nov 23:34:58 - DeliveryInfo: {"contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "b611ff7b3d6a41c2111df5de0b4431", "replyTo": "amq.gen-X-GqHkudQDoyW5rx", "In handle request of checkEmail  
data not found  
3 Nov 23:34:58 - checkEmail_queue { email1: 'maitray@gmail.co' }  
3 Nov 23:34:58 - Message: {"email1": "maitray@gmail.co"}  
3 Nov 23:34:58 - DeliveryInfo: {"contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "a69fd23afbc715d7c420007eb98f2d6c", "replyTo": "amq.gen-X-GqHkudQDoyW5rx", "In handle request of checkEmail  
data not found  
3 Nov 23:34:58 - checkEmail_queue { email1: 'maitray@gmail.com' }  
3 Nov 23:34:58 - Message: {"email1": "maitray@gmail.com"}  
3 Nov 23:34:58 - DeliveryInfo: {"contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "3e005b1425f36c52ef9531a6ba6fe428", "replyTo": "amq.gen-X-GqHkudQDoyW5rx", "In handle request of checkEmail  
data not found  
3 Nov 23:35:10 - register_queue { username: 'maitray',  
email: 'maitray@gmail.com',  
password: 'test',  
birthdate: '2016-11-30',  
location: 'San Jose,California',  
contact: '12345678901234',  
lastLogin: 'November 3rd, 2016, 11:35:10 pm' }  
3 Nov 23:35:10 - Message: {"username": "maitray", "email": "maitray@gmail.com", "password": "test", "birthdate": "2016-11-30", "location": "San Jose, California", "contact": "12345678901234", "In handle request of Register  
Mongoose: mpromise (mongoose's default promise library) is deprecated, plug in your own promise library instead: http://mongoosejs.com/docs/promises.html  
Registration successful
```

The bottom status bar shows "17:1 CRLF: UTF-8" and the date "11/3/2016".

- **After Login Screen**
- It will display all the advertisement which are not posted by the user who is currently logged in.

Macbook Pro Quantity: 3 Price: 1500	Car Quantity: 3 Price: 50000	Webcam Quantity: 3 Price: 500	Camera Quantity: 2 Price: 900
Macbook Air Quantity: 2 Price: 1500	Webcam Quantity: 3 Price: 300	Mobile Quantity: 3 Price: 900	Mercedes Benz Quantity: 4 Price: 35000
Refrigerator Quantity: 4 Price: 400	Smart TV Quantity: 4 Price: 800	Macbook Quantity: 3 Price: 1500	Toyota Quantity: 4 Price: 30000

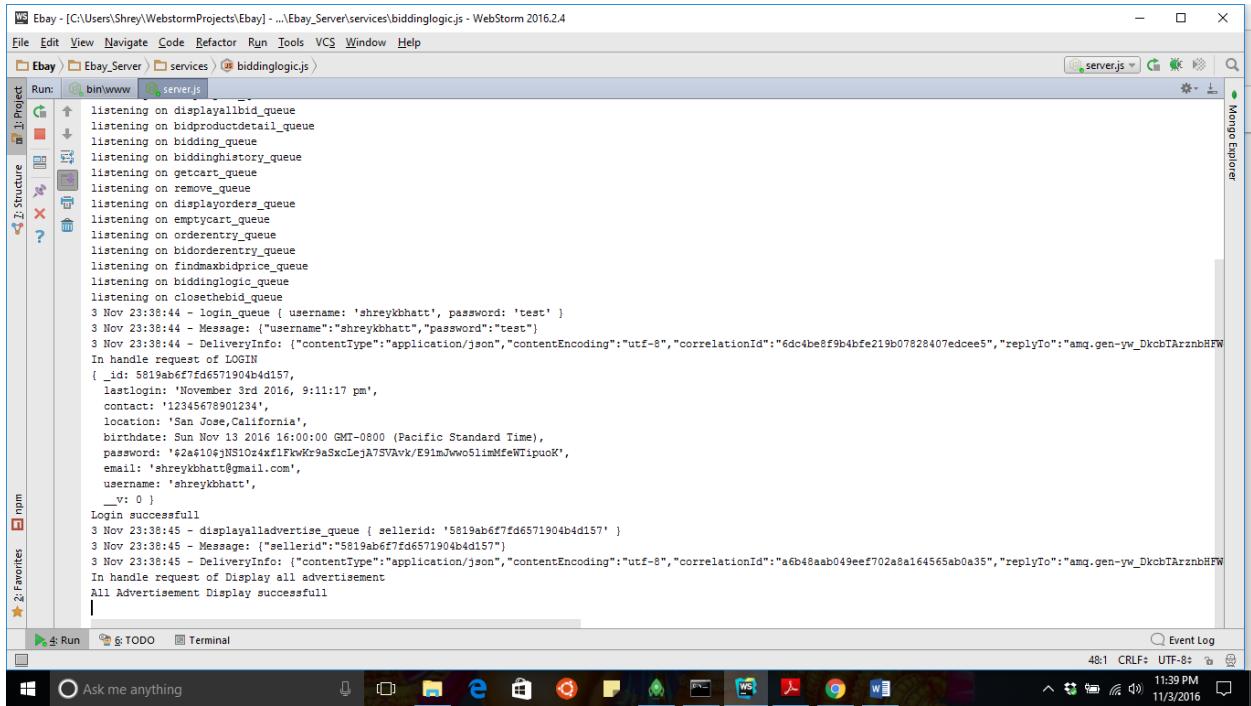
Display All Advertisement Client

```

[{"_id": "581a539fa9ff770604e69e53",  
 "shippingfrom: 'NY',  
 "sellerid: '581aef67a9ff770604e69e4c',  
 "description: 'Advanced Cooling',  
 "price: '400',  
 "quantity: '4',  
 "productname: 'Refrigerator',  
 "__v: 0 },  
 {"_id": "581a53d4a5ff770604e69e54",  
 "shippingfrom: 'SF',  
 "sellerid: '581aef67a9ff770604e69e4c',  
 "description: 'Ultra HD',  
 "price: '800',  
 "quantity: '4',  
 "productname: 'Smart TV',  
 "__v: 0 },  
 {"_id": "581a53ea9ff770604e69e55",  
 "shippingfrom: 'WA',  
 "sellerid: '581aef67a9ff770604e69e4c',  
 "description: '256 GHZ',  
 "price: '1500',  
 "quantity: '3',  
 "productname: 'Macbook',  
 "__v: 0 },  
 {"_id": "581a540ca9ff770604e69e56",  
 "shippingfrom: 'SF',  
 "sellerid: '581aef67a9ff770604e69e4c',  
 "description: 'High Speed',  
 "price: '30000',  
 "quantity: '4',  
 "productname: 'Toyota',  
 "__v: 0 } ] }  
 GET /displayalladvertis 304 30.843 ms - -

```

Display All Advertisement Server

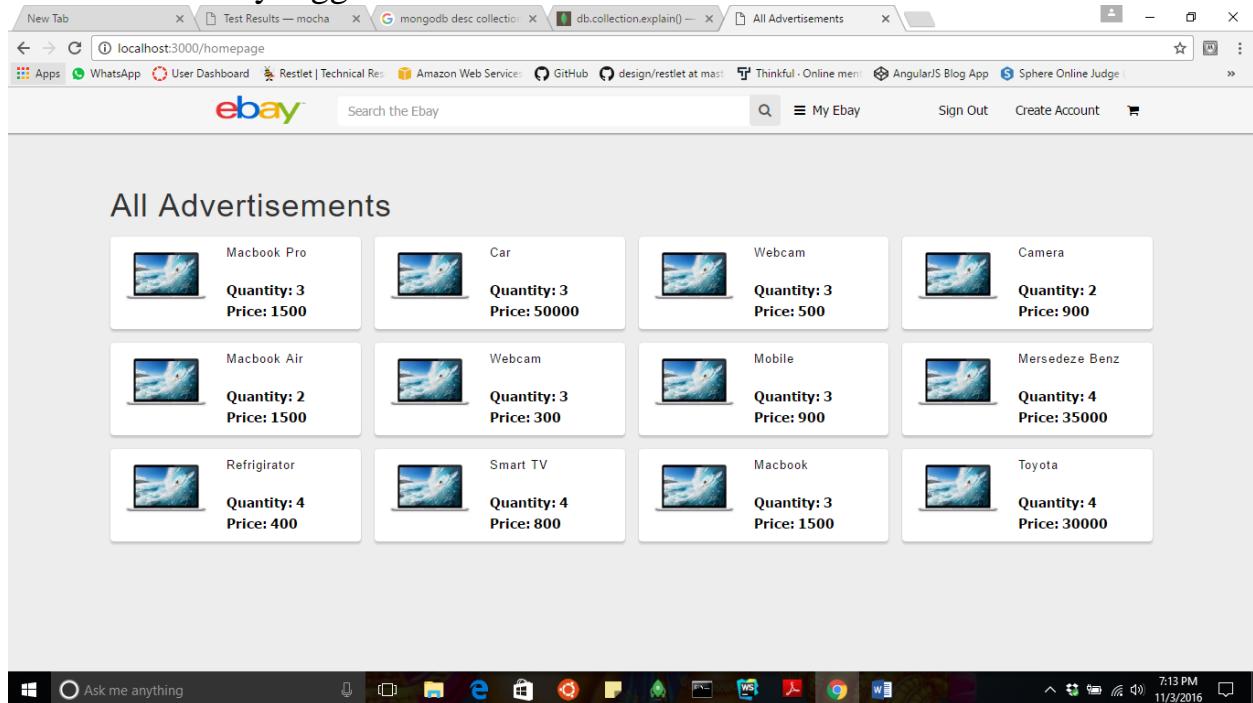


The screenshot shows the WebStorm IDE interface with the following details:

- Title Bar:** Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\\services\\biddinglogic.js - WebStorm 2016.2.4
- File Menu:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help
- Project Structure:** Shows a tree view of the project structure under the 'Ebay' root, including 'bin\\www' and 'server.js'.
- Code Editor:** The 'server.js' file is open, displaying code related to message queues and user authentication. Key snippets include:
 - listening on displayallbid_queue
 - listening on bidproductdetail_queue
 - listening on bidding_queue
 - listening on biddinghistory_queue
 - listening on getcart_queue
 - listening on displayorders_queue
 - listening on emptycart_queue
 - listening on orderentry_queue
 - listening on bidorderentry_queue
 - listening on findmaxbidprice_queue
 - listening on biddinglogic_queue
 - listening on closethebit_queue
- Output Panel:** Shows log messages from the application, including successful LOGIN and DISPLAY requests.
- Bottom Status Bar:** Event Log, 48:1 CRLF: UTF-8, 11:39 PM, 11/3/2016.

- HomePage

- On clicking the Ebay icon the Homepage will be opened.
- It will display All the advertisements which are not posted by the user who is currently logged in.



- Menu Pops Up on clicking the MyEbay

The screenshot shows a web browser window with multiple tabs open, including 'Test Results — mocha', 'mongodb desc collection', 'db.collection.explain() —', and 'All Advertisements'. The main content area displays a grid of advertisements:

Item	Quantity	Price
Macbook Pro	3	1500
Car	3	50000
Web	3	900
Macbook Air	2	1500
Webcam	3	300
Mobile	3	900
Refrigerator	4	400
Smart TV	4	800
Macbook	3	1500
Camera	2	900
Mersedez Benz	4	35000
Toyota	4	30000

The sidebar on the right includes links such as 'Sell', 'Bid', 'All Bids', 'My Bids', 'My Orders', 'My advertisements', and 'Last Login'. The bottom status bar shows the URL 'localhost3000/ProfilePage', the date '11/3/2016', and the time '7:14 PM'.

- **ProfilePage**

The screenshot shows a web browser window titled 'Profile' with the URL 'localhost:3000/ProfilePage'. The main content area displays the user's profile information:

My Profile

Ebay Handle: shreykbhatt

Birth Date: 2016-10-18

Location: San Jose, California

The bottom status bar shows the URL 'localhost3000/ProfilePage', the date '10/16/2016', and the time '4:14 PM'.

Profile Client

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ..\Ebay_Server\services\biddinglogic.js - WebStorm 2016.2.4

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay Ebay_Server services biddinglogic.js
Run: bin\www server.js
Project: 1: Project
  location: 'San Jose, California',
  birthdate: '2016-11-14T00:00:00.000Z',
  password: '62a4f0eNS104xf1FkWkr9a5xcLejA7SVAvk/E91mJwwo5lmMfeWTipucK',
  email: 'shreykhatt@gmail.com',
  username: 'shreykhatt',
  _v: 0 } } )
GET /ProfilePage 304 84.077 ms -
GET /ProfilePage_files/w3data.js.download 304 16.252 ms -
GET /ProfilePage_files/css(1) 304 7.600 ms -
GET /ProfilePage_files/font-awesome.css 304 6.946 ms -
GET /ProfilePage_files/css 304 10.014 ms -
GET /ProfilePage_files/bootstrap.css 304 9.222 ms -
GET /ProfilePage_files/productSectionStyle.css 304 10.601 ms -
GET /ProfilePage_files/jquery.min.js.download 304 7.433 ms -
GET /ProfilePage_files/mystyles.css 304 8.770 ms -
GET /ProfilePage_files/bootstrap.min.js.download 304 11.205 ms -
GET /angularjs/profile.js 304 8.049 ms -
GET /ProfilePage_files/styles.css 200 14.339 ms - 130766
GET /img/ebay.png 304 1.241 ms -
GET /img/mypic.jpg 304 1.523 ms -
GET /fonts/fontawesome-webfont.woff2?v=4.3.0 304 3.053 ms -
{ code: '200',
  data:
  [ { _id: '5819ah6f7fd6571904b4d157',
    lastlogin: 'November 3rd 2016, 9:11:17 pm',
    contact: '12345678901234',
    location: 'San Jose, California',
    birthdate: '2016-11-14T00:00:00.000Z',
    password: '62a4f0eNS104xf1FkWkr9a5xcLejA7SVAvk/E91mJwwo5lmMfeWTipucK',
    email: 'shreykhatt@gmail.com',
    username: 'shreykhatt',
    _v: 0 } } )
GET /ProfilePage 304 25.847 ms -

```

Event Log

41:1 CRLF: UTF-8 11:43 PM 11/3/2016

Profile Server

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ..\Ebay_Server\services\biddinglogic.js - WebStorm 2016.2.4

```

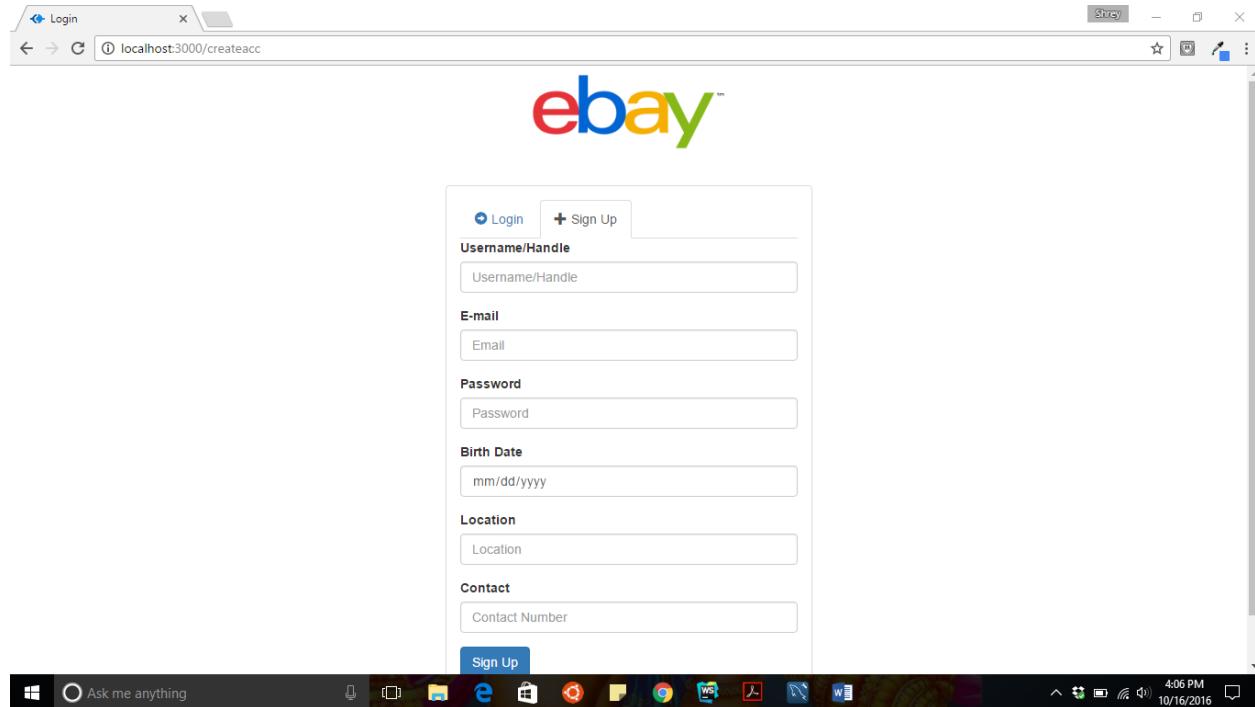
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay Ebay_Server services biddinglogic.js
Run: bin\www server.js
Project: 1: Project
  listening on displayalladvertise_queue
  listening on lastlogin_queue
  listening on profile_queue
  listening on productdetail_queue
  listening on addcart_queue
  listening on bid_queue
  listening on displaybid_queue
  listening on displayallbid_queue
  listening on bidproductdetail_queue
  listening on bidding_queue
  listening on biddinghistory_queue
  listening on getcart_queue
  listening on remove_queue
  listening on displayorders_queue
  listening on emptycart_queue
  listening on orderentry_queue
  listening on bidorderentry_queue
  listening on findmaxbidprice_queue
  listening on biddinglogic_queue
  listening on closethebid_queue
3 Nov 23:42:26 - lastlogin_queue { username: 'shreykhatt' }
3 Nov 23:42:26 - Message: {"username":"shreykhatt"}
3 Nov 23:42:26 - DeliveryInfo: {"contentType":"application/json","contentEncoding":"utf-8","correlationId":"da13cc52a40ec6e8b5f79b8a38b596e8","replyTo":"amq.gen-c5_x0XZPiQvc_708"
In handle request of LOGIN
In handle request of LOGIN
Lastlogin update successful
3 Nov 23:42:27 - lastlogin_queue { username: 'shreykhatt' }
3 Nov 23:42:27 - Message: {"username":"shreykhatt"}
3 Nov 23:42:27 - DeliveryInfo: {"contentType":"application/json","contentEncoding":"utf-8","correlationId":"12515e462370c8b052a53c25bb99f4b6","replyTo":"amq.gen-c5_x0XZPiQvc_708"
In handle request of LOGIN
In handle request of LOGIN
Lastlogin update successful

```

Event Log

41:1 CRLF: UTF-8 11:44 PM 11/3/2016

- **Create Account**
- On clicking the create account page in the title bar the registration/SignUp page appears.



- **All Bids**
- This page displays all the products available for bidding which are not posted by the user who is currently logged in.

Item	Description	Price
Washing Machine	950	
BMW	30000	
MAC	1200	
Refrigerator	900	
Iphone	900	
Mobile	2	
Webcam	900	
Microwave	1200	
Car	35500	
Toyota	35000	
Machine	35000	
Refrigerator	3600	

All Bids Page Client

The screenshot shows the WebStorm IDE interface with the project 'Ebay' open. The 'server.js' file is selected in the 'Run' tab. The code displays a list of bids from the database:

```
    bidderid: null,
    isBidEnded: false,
    bidendtime: '1478578007685',
    bidstarttime: '1478232407685',
    bidshippingfrom: 'NJ',
    bidsellerid: '581a4f67a9ff770604e69e4c',
    biddescription: 'a',
    bidprice: '35000',
    bidproductname: 'Machine',
    __v: 0 },
    { _id: '581c09720c68583fd4f0b349',
    bidderid: null,
    isBidEnded: false,
    bidendtime: '1478578034140',
    bidstarttime: '1478232434140',
    bidshippingfrom: 'WA',
    bidsellerid: '581a4f67a9ff770604e69e4c',
    biddescription: 'O',
    bidprice: '3600',
    bidproductname: 'Refrigerator',
    __v: 0 },
    { _id: '581c09840c68583fd4f0b34a',
    bidderid: null,
    isBidEnded: false,
    bidendtime: '1478578052088',
    bidstarttime: '1478232452088',
    bidshippingfrom: 'NJ',
    bidsellerid: '581a4f67a9ff770604e69e4c',
    biddescription: 'a',
    bidprice: '1200',
    bidproductname: 'Mobile',
    __v: 0 } ] }
```

The terminal output shows the response to the GET request:

```
GET /displayallbid 304 63.328 ms - -
```

All Bid Page Server

The screenshot shows the WebStorm IDE interface with the project 'Ebay' open. The 'server.js' file is selected in the 'Run' tab. The code handles a 'displayallbid' request by listening on various queues and sending a response:

```
"C:\Program Files (x86)\JetBrains\WebStorm 2016.2\bin\runnerw.exe" "C:\Program Files\nodejs\node.exe" C:\Users\Shrey\WebstormProjects\Ebay\Ebay_Server\server.js
listening on login_queue
listening on register_queue
listening on checkUsername_queue
listening on checkEmail_queue
listening on signout_queue
listening on advertisement_queue
listening on displayadvertise_queue
listening on displayalladvertise_queue
listening on lastlogin_queue
listening on profile_queue
listening on productdetail_queue
listening on addcart_queue
listening on bid_queue
listening on displaybid_queue
listening on displayallbid_queue
listening on bidproductdetail_queue
listening on bidding_queue
listening on biddinghistory_queue
listening on getcart_queue
listening on remove_queue
listening on displayorders_queue
listening on emptycart_queue
listening on orderentry_queue
listening on bidorderentry_queue
listening on findmaxbidprice_queue
listening on biddinglogic_queue
listening on closethebid_queue
3 Nov 23:45:59 - displayallbid_queue { bidsellerid: '5819ab6f7fd6571904b4d157' }
3 Nov 23:45:59 - Message: { "bidsellerid": "5819ab6f7fd6571904b4d157" }
3 Nov 23:45:59 - DeliveryInfo: { "contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "cd14835f5a742d8488f7bc639b27634e", "replyTo": "amq.gen-c5_x0XZPiQvc_708" }
In handle request of Display all Bids
All Bid Display successful
```

- **My Advertisements**
- This page displays all the advertisements which are posted by the user who is currently logged in.

My Advertisement Client

```

// ...
app.get('/displayallbid', function (req, res) {
  var biddescription = 'a',
    bidprice = '1200',
    bidproductname = 'Mobile',
    __v: 0
  ];
  res.json({
    biddescription: 'a',
    bidprice: '1200',
    bidproductname: 'Mobile',
    __v: 0
  });
});

app.get('/myadvertisement', function (req, res) {
  res.json([
    {
      _id: '581a52e1a9ff770604e69e4d',
      shippingfrom: 'CA',
      sellerid: '5819ab6b7fd6571904b4d157',
      description: 'Brand New',
      price: '1500',
      quantity: '2',
      productname: 'Book',
      __v: 0
    }
  ]);
});

app.get('/displayadvertise', function (req, res) {
  res.json([
    {
      _id: '581a52e1a9ff770604e69e4d',
      shippingfrom: 'CA',
      sellerid: '5819ab6b7fd6571904b4d157',
      description: 'Brand New',
      price: '1500',
      quantity: '2',
      productname: 'Book',
      __v: 0
    }
  ]);
});
// ...

```

My Advertisement Server

```

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\services\\biddinglogic.js - WebStorm 2016.2.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Project Run: bin/www server.js
1: Project 2: Structure 3: Run 4: Bin 5: Npm 6: Favorites
listening on advertisement_queue
listening on displayadvertise_queue
listening on displayalladvertise_queue
listening on lastlogin_queue
listening on profile_queue
listening on productdetail_queue
listening on addcart_queue
listening on bid_queue
listening on displaybid_queue
listening on displayallbid_queue
listening on bidproductdetail_queue
listening on bidding_queue
listening on biddinghistory_queue
listening on getcart_queue
listening on remove_queue
listening on displayorders_queue
listening on emptycart_queue
listening on orderentry_queue
listening on bidorderentry_queue
listening on findmaxbidprice_queue
listening on biddinglogic_queue
listening on closethebid_queue
3 Nov 23:45:59 - displayallbid_queue { bidderid: '5819ab6f7fd6571904b4d157' }
3 Nov 23:45:59 - Message: {"bidderid":"5819ab6f7fd6571904b4d157"}
3 Nov 23:45:59 - DeliveryInfo: {"contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "cd14835f5a742d8488f7bc639b27634e", "replyTo": "amq.gen-c5_x0XZPiQve_708"}
In handle request of Display all Bids
All Bid Display successful
3 Nov 23:47:35 - displayadvertise_queue { sellerid: '5819ab6f7fd6571904b4d157' }
3 Nov 23:47:35 - Message: {"sellerid": "5819ab6f7fd6571904b4d157"}
3 Nov 23:47:35 - DeliveryInfo: {"content_type": "application/json", "contentEncoding": "utf-8", "correlationId": "3c0e222fad1bcf3fabfb828d5e357dc1", "replyTo": "amq.gen-c5_x0XZPiQve_708"}
In handle request of Display my advertisement
My Advertisement display successful

```

- **My Bids**
- This page displays all the products available for bidding which are posted by the user who is currently logged in.

Item	Description	Price
BMW	30000	1200
MAC	900	900
Refrigerator	900	900
Iphone	900	35500
Mobile	2	1200
Webcam	900	1200
Microwave	1200	Car
Toyota	35000	3600
Machine	35000	Mobile
Refrigerator	3600	1200
Mobile	1200	

My Bid Page Client

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\\services\\biddinglogic.js - WebStorm 2016.2.4

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Run: bin\\www server.js
Project Structure
? 2 Favorites
  2 npm
  Ask me anything
  Event Log
  44:1 CRLF: UTF-8
  11:49 PM 11/3/2016

GET /js/custom.js 304 4.503 ms -
GET /js/owl-carousel.js 304 5.423 ms -
GET /img/macbook.jpg 304 3.812 ms -
{ code: '200',
  data:
    [ { _id: '581a52ela9ff770604e69e4d',
      shippingfrom: 'CA',
      sellerid: '5819ab6f7fd6571904b4d157',
      description: 'Brand New',
      price: '1500',
      quantity: '2',
      productname: 'Book',
      _v: 0 } ] }
GET /displayadvertis 200 9.428 ms - 180
GET /mybids 304 5.708 ms -
GET /css/bootstrap.css 304 13.935 ms -
GET /css/styles.css 304 6.470 ms -
GET /css/font-awesome.css 304 8.800 ms -
GET /css/mystyles.css 304 5.653 ms -
GET /angularjs/bid.js 304 4.577 ms -
GET /img/ebay.png 304 1.320 ms -
GET /js/jquery.js 304 0.944 ms -
GET /js/bootstrap.js 304 0.928 ms -
GET /js/checkbox.js 304 1.194 ms -
GET /js/ionrangeslider.js 304 1.169 ms -
GET /js/jqzoom.js 304 5.177 ms -
GET /js/card-payment.js 304 4.291 ms -
GET /img/macbook.jpg 304 2.980 ms -
GET /js/owl-carousel.js 304 4.921 ms -
GET /js/magnific.js 304 4.504 ms -
GET /js/custom.js 304 4.152 ms -
{ code: '200', data: [] }
GET /displaybid 200 7.247 ms - 2

```

Run TODO Terminal

Event Log
44:1 CRLF: UTF-8
11:49 PM 11/3/2016

My Bid Page Server

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\\services\\biddinglogic.js - WebStorm 2016.2.4

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Run: bin\\www server.js
Project Structure
? 2 Favorites
  2 npm
  Ask me anything
  Event Log
  44:1 CRLF: UTF-8
  11:49 PM 11/3/2016

listening on productdetail_queue
listening on addcart_queue
listening on bid_queue
listening on displaybid_queue
listening on displayallbid_queue
listening on bidproductdetail_queue
listening on bidding_queue
listening on biddinghistory_queue
listening on getcart_queue
listening on remove_queue
listening on displayorders_queue
listening on emptycart_queue
listening on orderentry_queue
listening on bidorderentry_queue
listening on findmaxbidprice_queue
listening on biddinglogic_queue
listening on closethebid_queue
3 Nov 23:45:59 - displayallbid_queue { sellerid: '5819ab6f7fd6571904b4d157' }
3 Nov 23:45:59 - Message: {"bidSellerId":"5819ab6f7fd6571904b4d157"}
3 Nov 23:45:59 - DeliveryInfo: {"contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "cdi4835f5a742d8488f7bc639b27634e", "replyTo": "amq.gen-c5_x0XZPiQvc_708"
In handle request of Display all Bids
All Bid Display successful
3 Nov 23:47:35 - displayadvertiset_queue { sellerid: '5819ab6f7fd6571904b4d157' }
3 Nov 23:47:35 - Message: {"sellerId": "5819ab6f7fd6571904b4d157"}
3 Nov 23:47:35 - DeliveryInfo: {"contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "3c0e222fad1bcf3fabfb828d5e357dc1", "replyTo": "amq.gen-c5_x0XZPiQvc_708"
In handle request of Display my advertisement
My Advertisement display successful
3 Nov 23:49:12 - displaybid_queue { bidsellerid: '5819ab6f7fd6571904b4d157' }
3 Nov 23:49:12 - Message: {"bidSellerId": "5819ab6f7fd6571904b4d157"}
3 Nov 23:49:12 - DeliveryInfo: {"contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "6356efeca3636655600381110768e95e", "replyTo": "amq.gen-c5_x0XZPiQvc_708"
In handle request of Display my bid
My Bid display successful

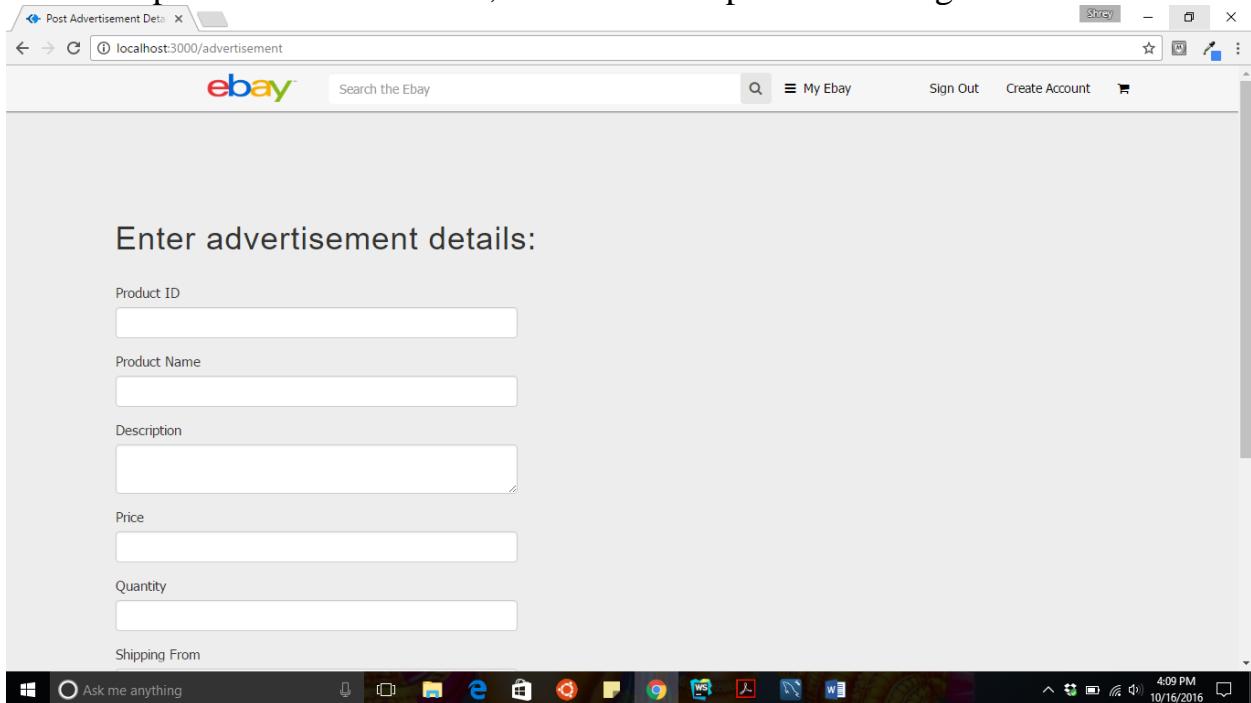
```

Run TODO Terminal

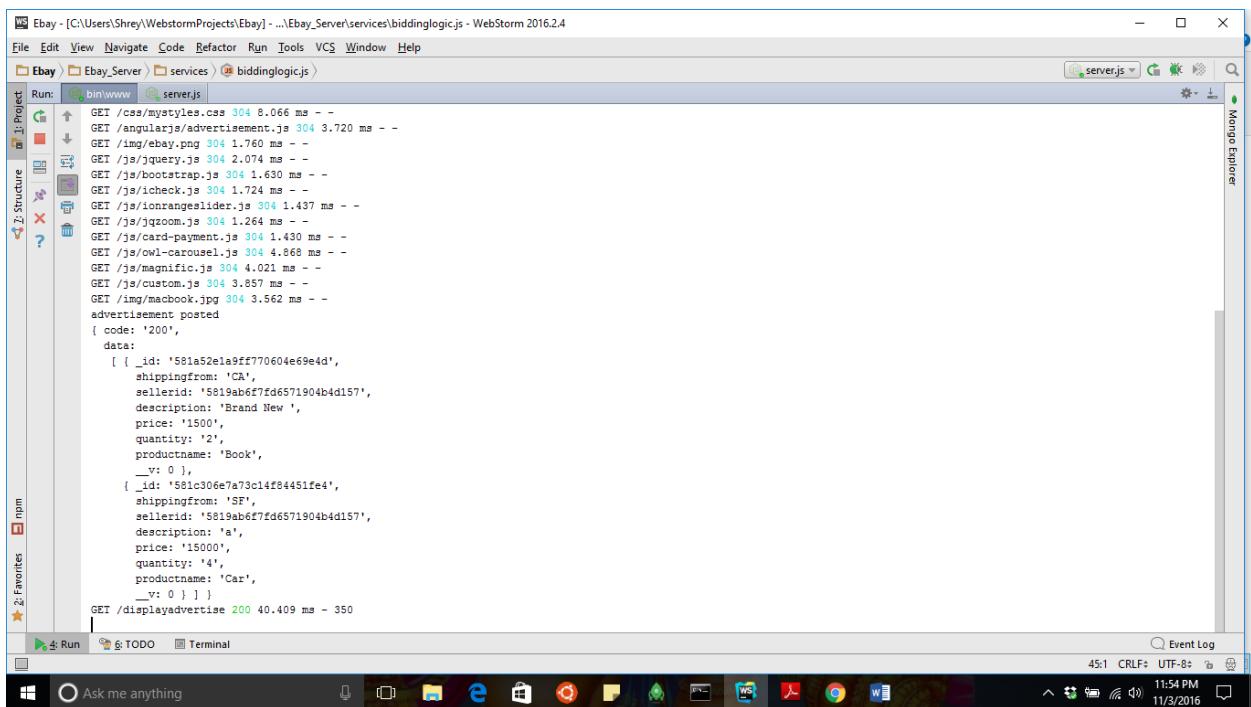
Event Log
44:1 CRLF: UTF-8
11:49 PM 11/3/2016

- Post Advertise

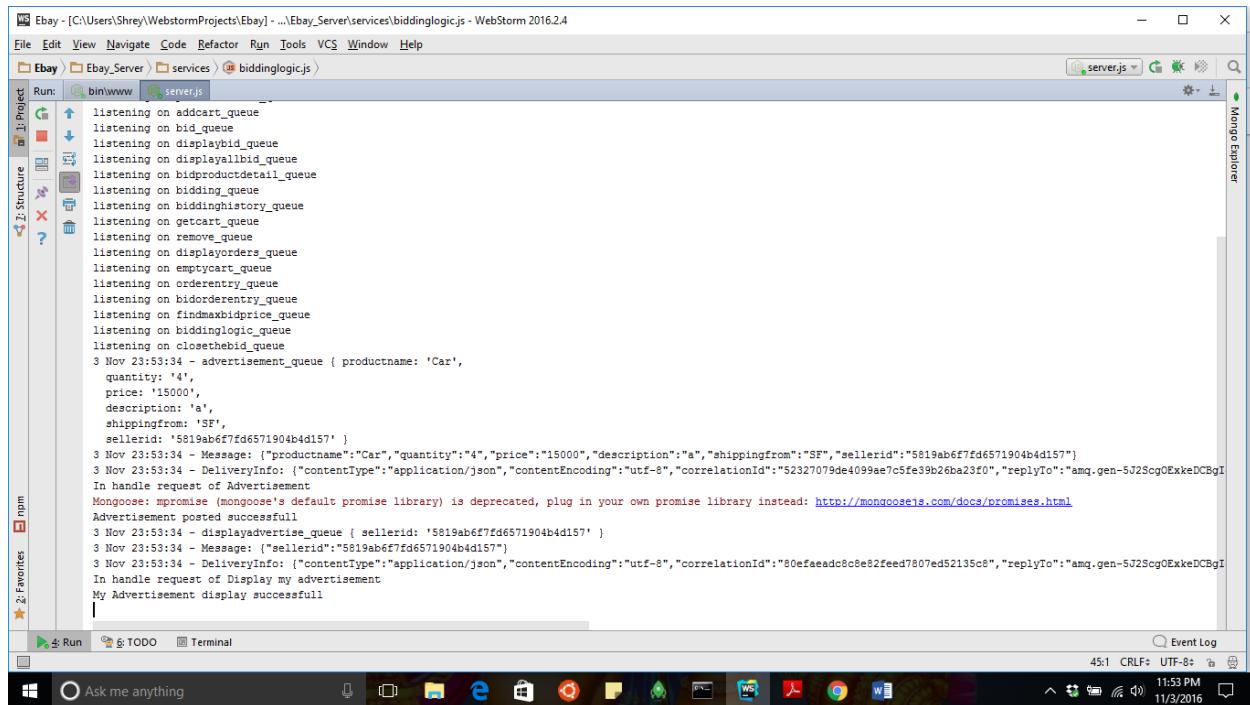
- To post the advertisement, User will fill up the following form.



Post Advertisement Client



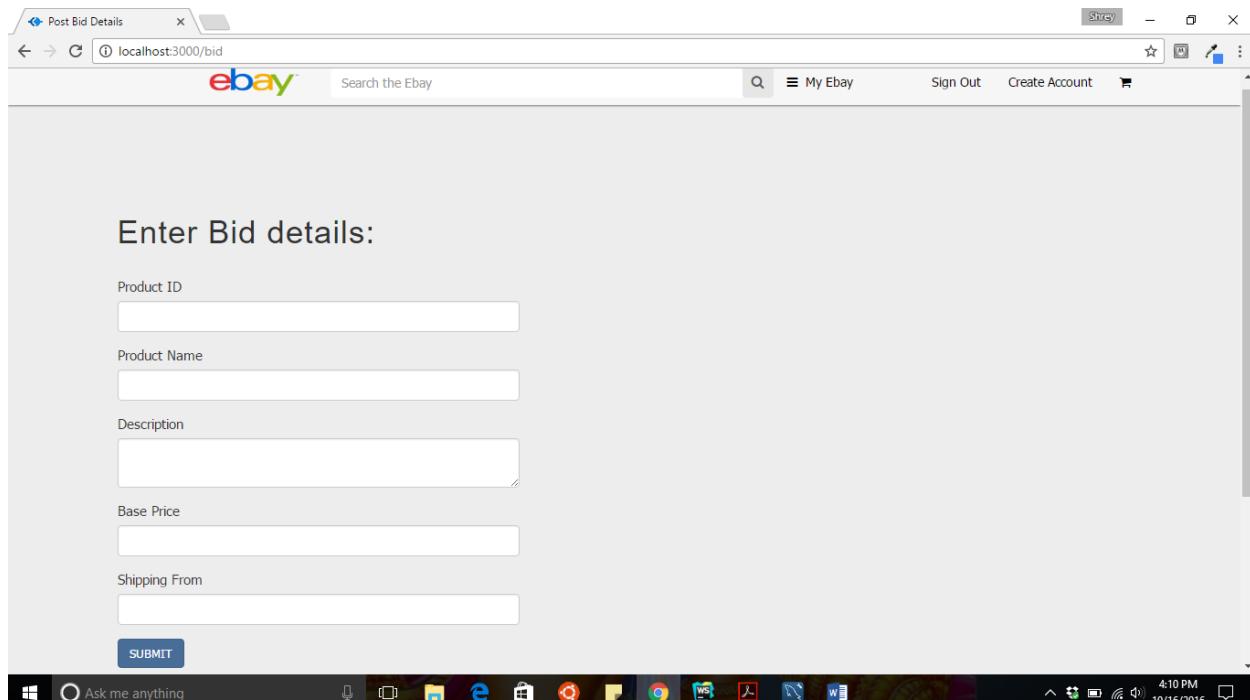
Post Advertisement Server



```

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\services\biddinglogic.js - WebStorm 2016.2.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Project Run: bin/www server.js
1: Project 2: Structure 3: Run 4: Bin 5: Npm 6: Favorites
listening on addcart_queue
listening on bid_queue
listening on displaybid_queue
listening on displayallbid_queue
listening on bidproductdetail_queue
listening on bidding_queue
listening on biddinghistory_queue
listening on getcart_queue
listening on remove_queue
listening on displayorders_queue
listening on emptycart_queue
listening on orderentry_queue
listening on bidorderentry_queue
listening on findmaxbidprice_queue
listening on biddinglogic_queue
listening on closethebid_queue
3 Nov 23:53:34 - advertisement_queue { productname: 'Car',
  quantity: '4',
  price: '15000',
  description: 'a',
  shippingfrom: 'SF',
  sellerid: '5819ab6f7fd6571904b4d157' }
3 Nov 23:53:34 - Message: {"productname": "Car", "quantity": "4", "price": "15000", "description": "a", "shippingfrom": "SF", "sellerid": "5819ab6f7fd6571904b4d157"}
3 Nov 23:53:34 - DeliveryInfo: {"contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "52327079de4099ae7c5fe39b26ba23f0", "replyTo": "amq.gen-5J2ScgOEExkeDCBgI"
In handle request of Advertisement
Mongoose: mpromise (mongoose's default promise library) is deprecated, plug in your own promise library instead: http://mongoosejs.com/docs/promises.html
Advertisement posted successful
3 Nov 23:53:34 - displayadvertise_queue { sellerid: '5819ab6f7fd6571904b4d157' }
3 Nov 23:53:34 - Message: {"sellerid": "5819ab6f7fd6571904b4d157"}
3 Nov 23:53:34 - DeliveryInfo: {"contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "80efaaeadc8c8e82feed7807ed52135c8", "replyTo": "amq.gen-5J2ScgOEExkeDCBgI"
In handle request of Display my advertisement
My Advertisement display successful
|
```

- **Post Bid**
- To post the bid, User will fill up the following form.



Post Bid Details

localhost:3000/bid

ebay Search the Ebay My Ebay Sign Out Create Account

Enter Bid details:

Product ID

Product Name

Description

Base Price

Shipping From

SUBMIT

Post bid Client

```

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\services\\biddinglogic.js - WebStorm 2016.2.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Run: bin\\www server.js
Project Structure
in post bid
POST /postbid 200 35.131 ms - 4989
GET /css/font-awesome.css 304 7.544 ms -
GET /css/bootstrap.css 304 7.406 ms -
GET /css/styles.css 304 7.099 ms -
GET /css/mystyles.css 304 7.433 ms -
GET /angularjs/bid.js 304 7.855 ms -
GET /js/jquery.js 304 2.979 ms -
GET /js/bootstrap.js 304 1.142 ms -
GET /js/iCheck.js 304 1.916 ms -
GET /js/ionrangeslider.js 304 1.286 ms -
GET /js/jqzoom.js 304 3.125 ms -
GET /js/card-payment.js 304 1.389 ms -
GET /js/owl-carousel.js 304 1.431 ms -
advertisements posted
GET /js/magnific.js 304 8.395 ms -
GET /js/custom.js 304 2.275 ms -
GET /img/ebay.png 304 6.532 ms -
GET /img/macbook.jpg 304 6.232 ms -
{ code: '200',
  data:
    [ { _id: '581c33add8cd84ebc3ce25d',
        bidderid: null,
        isBidEnded: false,
        bidendtime: '1478588845235',
        bidstarttime: '1478243245235',
        bidshippingfrom: 'NJ',
        bidsellerid: '5819ab6f7fd6571904b4d157',
        biddescription: 'a',
        bidprice: '35500',
        bidproductname: 'BMW',
        ...v: 0 } ] }
GET /displaybid 200 49.975 ms - 266

```

Event Log
52:1 CRLF: UTF-8 12:07 AM 11/4/2016

Post Bid Server

```

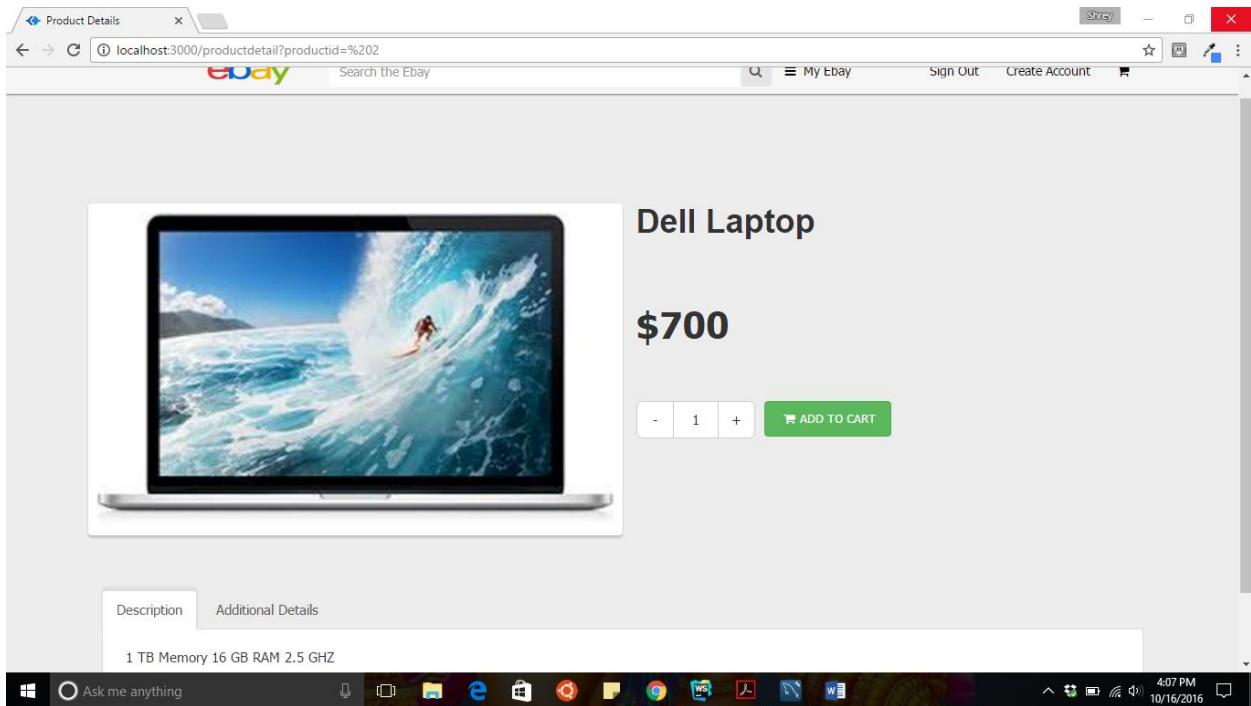
Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\services\\biddinglogic.js - WebStorm 2016.2.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Run: bin\\www server.js
Project Structure
listening on displaybid_queue
listening on displayallbid_queue
listening on bidproductdetail_queue
listening on bidding_queue
listening on biddinghistory_queue
listening on getcart_queue
listening on remove_queue
listening on displayorders_queue
listening on emptycart_queue
listening on orderentry_queue
listening on bidorderentry_queue
listening on findmaxbidprice_queue
listening on biddinglogic_queue
listening on closethebid_queue
4 Nov 00:07:25 - bid_queue { bidproductname: 'BMW',
  bidprice: '35500',
  biddescription: 'a',
  bidshippingfrom: 'NJ',
  bidsellerid: '5819ab6f7fd6571904b4d157',
  start: 1478243245235,
  end: 1478588845235,
  isBidEnded: 0,
  bidderid: null }
4 Nov 00:07:25 - Message: { "bidproductname": "BMW", "bidprice": "35500", "biddescription": "a", "bidshippingfrom": "NJ", "bidsellerid": "5819ab6f7fd6571904b4d157", "start": 1478243245235, "end": 1478588845235 }
4 Nov 00:07:25 - DeliveryInfo: { "contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "4c40c3ca86d496590110f7537ed", "replyTo": "amq.gen-VOXywerDQcQkDvKB" }
In handle request of Bid
Mongoose: mpromise (mongoose's default promise library) is deprecated, plug in your own promise library instead: http://mongoosejs.com/docs/promises.html
Bid posted successful
4 Nov 00:07:25 - displaybid_queue { bidsellerid: '5819ab6f7fd6571904b4d157' }
4 Nov 00:07:25 - Message: { "bidsellerid": "5819ab6f7fd6571904b4d157" }
4 Nov 00:07:25 - DeliveryInfo: { "contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "b526351984f5d40e5bbe8a4884577b38", "replyTo": "amq.gen-VOXywerDQcQkDvKB" }
In handle request of Display my bid
My Bid display successfull

```

Event Log
52:1 CRLF: UTF-8 12:07 AM 11/4/2016

- Product Details

- It will display product details like Product name, Price and Quantity(can be edited) and product description.



Product Detail Client

```

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\\services\\biddinglogic.js - WebStorm 2016.2.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Run: bin\\www server.js
data:
[ { _id: '5812d8f133abaea220c2cad89',
  shippingfrom: 'SF',
  sellerid: null,
  description: '256 GB 3 GHZ',
  price: '1500',
  quantity: '3',
  productname: 'Macbook Pro',
  __v: 0 } ]
GET /productdetail?productid=5812d8f133abaea220c2cad89 500 37.824 ms - 3079
hello in productdetail 581a5355a9ff770604e69e51
Successful productdetails
{ code: '200',
  data:
  [ { _id: '581a5355a9ff770604e69e51',
    shippingfrom: 'SF',
    sellerid: [Object],
    description: 'Samsung Galaxy S7',
    price: '900',
    quantity: '3',
    productname: 'Mobile',
    __v: 0 } ] }
{ code: '200',
  data:
  [ { _id: '581a5355a9ff770604e69e51',
    shippingfrom: 'SF',
    sellerid: [Object],
    description: 'Samsung Galaxy S7',
    price: '900',
    quantity: '3',
    productname: 'Mobile',
    __v: 0 } ] }
GET /productdetail?productid=581a5355a9ff770604e69e51 304 33.398 ms -
GET /css/bootstrap.css 304 5.952 ms -

```

Product Detail Server

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\\services\\biddinglogic.js - WebStorm 2016.2.4

File Edit View Navigate Code Refactor Run Tools VCS Window Help

Ebay > Ebay_Server > services > biddinglogic.js

```

Run: bin/www server.js
Project Structure
  quantity: '4',
  price: '15000',
  description: 'a',
  shippingfrom: 'SF',
  sellerid: '5819ab6f7fd6571904b4d157' }

3 Nov 23:53:34 - Message: {"productName":"Car","quantity":4,"price":15000,"description":a,"shippingfrom":SF,"sellerid":5819ab6f7fd6571904b4d157}
3 Nov 23:53:34 - DeliveryInfo: {"contentType":"application/json","contentEncoding":"utf-8","correlationId":52327079de4099ae7c5fe39b2fba2f0,"replyTo":amq.gen-5J2SogOExkeDCBgI
In handle request of Advertisement
Mongoose: mpromise (mongoose's default promise library) is deprecated, plug in your own promise library instead: http://mongoosejs.com/docs/promises.html
Advertisement posted successful
3 Nov 23:53:34 - displayadvertise_queue { sellerid: '5819ab6f7fd6571904b4d157' }
3 Nov 23:53:34 - Message: {"sellerid":5819ab6f7fd6571904b4d157}
3 Nov 23:53:34 - DeliveryInfo: {"contentType":"application/json","contentEncoding":"utf-8","correlationId":80efeadc8c8e82feed7807ed52135c8,"replyTo":amq.gen-5J2SogOExkeDCBgI
In handle request of Display my advertisement
My Advertisement display successful
3 Nov 23:57:08 - displayalladvertise_queue { sellerid: '5819ab6f7fd6571904b4d157' }
3 Nov 23:57:08 - Message: {"sellerid":5819ab6f7fd6571904b4d157}
3 Nov 23:57:08 - DeliveryInfo: {"contentType":"application/json","contentEncoding":"utf-8","correlationId":a1cefa758dd50d20f7c0e4d4057c0be7,"replyTo":amq.gen-5J2SogOExkeDCBgI
In handle request of Display all advertisement
All Advertisement Display successful
3 Nov 23:57:10 - productdetail_queue { productid: '5812d8f133abea220c2cad89' }
3 Nov 23:57:10 - Message: {"productid":5812d8f133abea220c2cad89}
3 Nov 23:57:10 - DeliveryInfo: {"contentType":"application/json","contentEncoding":"utf-8","correlationId":8897009e249612867ec148194a6bea82,"replyTo":amq.gen-5J2SogOExkeDCBgI
In handle request of productdetail
5812d8f133abea220c2cad89
Product detail fetching successful
3 Nov 23:57:14 - productdetail_queue { productid: '581a5355a9ff770604e69e51' }
3 Nov 23:57:14 - Message: {"productid":581a5355a9ff770604e69e51}
3 Nov 23:57:14 - DeliveryInfo: {"contentType":"application/json","contentEncoding":"utf-8","correlationId":cd885a7d2ff29128ad483d4ffffd137fd,"replyTo":amq.gen-5J2SogOExkeDCBgI
In handle request of productdetail
581a5355a9ff770604e69e51
Product detail fetching successful

```

Run TODO Terminal Event Log

62:1 CRLF: UTF-8 11:58 PM 11/3/2016

- Additional Details**

Additional details will contain Seller and the location from where the product is shipped.

Dell Laptop

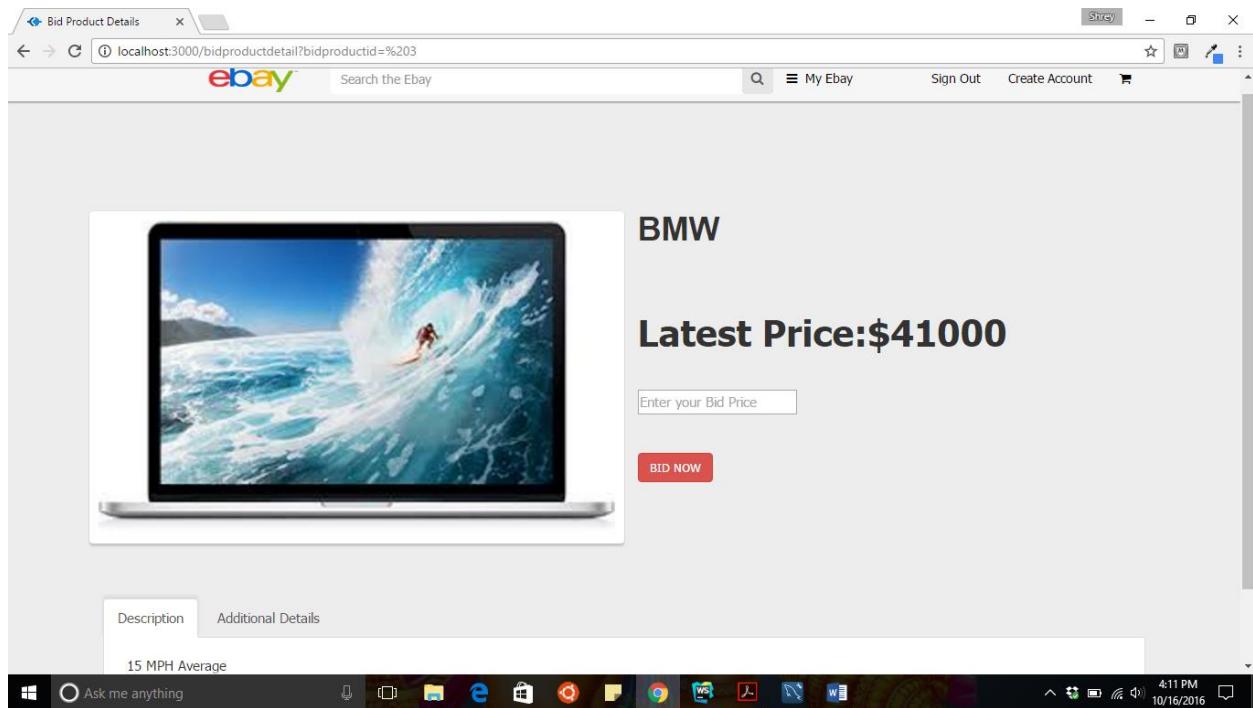
\$700

- 1 + ADD TO CART

Description	Additional Details
Seller	parth
Shipped From	NY

- Bid Product Details**

- It will display product details like Product name, Base Price and product description.
- User can Bid on the product by entering the Bid price. After bidding the entry for that product and that particular user will be their in the bid and biddinghistory table. So when User will go to “all bids page” or “bid product detail page” then the price will be updated to latest maximum bid price.



- Shopping-Cart

- User can add or remove items from the shopping card.

The screenshot shows a web browser window titled "Shopping Cart" with the URL "localhost:3000/cart". The page has a header with the eBay logo and a search bar. Below the header, it says "My Shopping Cart". A table lists the items in the cart:

Product	Title	Price	Quantity	Total	Remove
	Dell Laptop	700	7	4900	
	Macbook Pro	1000	3	3000	
	WebCam	500	2	1000	

On the right side of the cart table, there are summary details:

- Subtotal: 8900
- Shipping: Free
- Total: 8900

A blue "CHECKOUT" button is located to the right of the summary details. At the bottom left of the page, there is a "CONTINUE SHOPPING" button. The browser's taskbar at the bottom shows various pinned icons and the time "4:09 PM".

Shooping Cart Client

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\\services\\biddinglogic.js - WebStorm 2016.2.4

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Run: bin\\www server.js
Project Structure
? & bin\\www server.js
GET /css/font-awesome.css 304 5.871 ms -
GET /css/styles.css 304 4.833 ms -
GET /css/mystyles.css 304 4.461 ms -
GET /angularjs/shoppingcart.js 304 4.323 ms -
GET /img/ebay.png 304 1.067 ms -
GET /js/jquery.js 304 1.650 ms -
GET /js/bootstrap.js 304 1.346 ms -
GET /js/checkbox.js 304 1.258 ms -
GET /js/ionrangeslider.js 304 13.385 ms -
GET /js/jqzoom.js 304 12.685 ms -
GET /js/owl-carousel.js 304 10.962 ms -
GET /js/magnific.js 304 10.108 ms -
GET /js/card-payment.js 304 13.893 ms -
GET /js/custom.js 304 10.131 ms -
GET /fonts/fontawesome-webfont.woff2?v=4.3.0 304 2.843 ms -
GET /img/macbook2.jpg 304 2.760 ms -
In getcart node
[ { _id: '581bee940c68583fd4f0b33e',
  total: '120000',
  userid: '5819ab6f7fd6571904b4d157',
  price: '30000',
  quantity: '4',
  productname: 'Toyota',
  __v: 0 },
  { _id: '581c0a5527b01846d449c942',
  total: '1800',
  userid: '5819ab6f7fd6571904b4d157',
  price: '900',
  quantity: '2',
  productname: 'Camera',
  __v: 0 } ]
581bee940c68583fd4f0b33e
POST /getcart 200 12.406 ms - 297

```

Event Log
180:1 CRLF: UTF-8
12:10 AM 11/4/2016

Shopping Cart Server

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Server\\services\\biddinglogic.js - WebStorm 2016.2.4

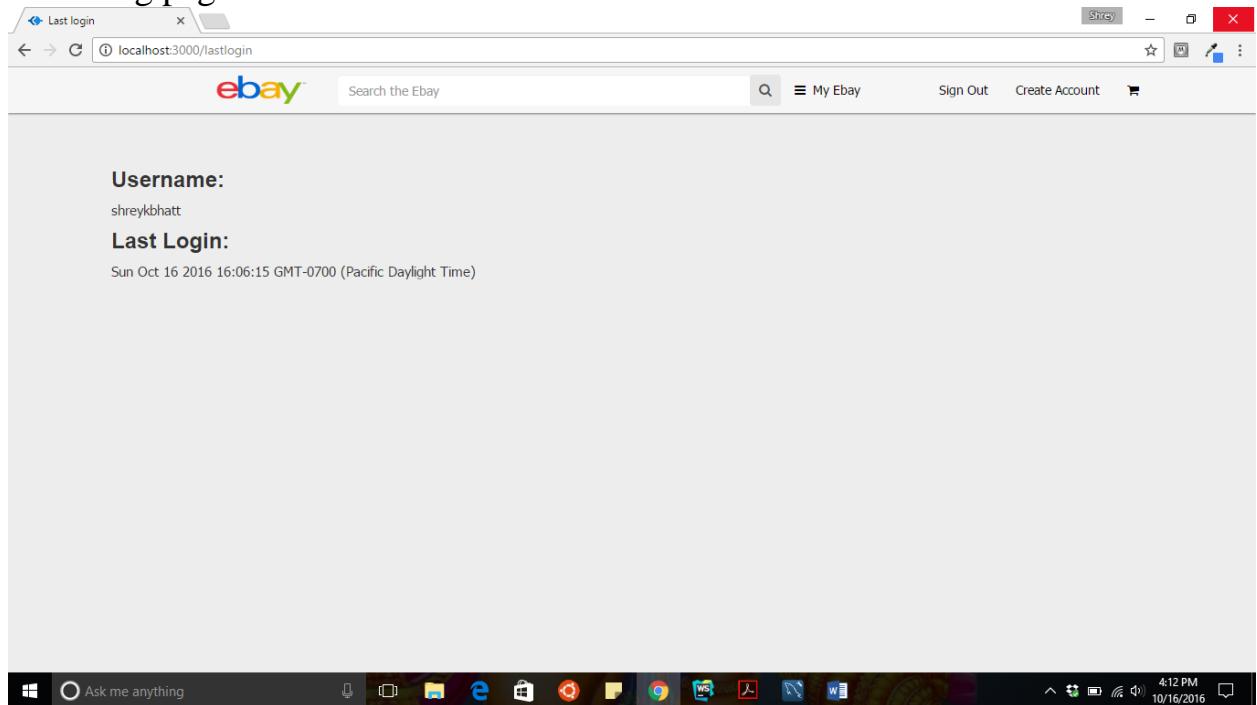
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Ebay > Ebay_Server > services > biddinglogic.js
Run: bin\\www server.js
Project Structure
? & bin\\www server.js
contact: '12345678901234',
location: 'San Jose, California',
birthdate: Sun Nov 13 2016 16:00:00 GMT-0800 (Pacific Standard Time),
password: '$2a$10$OsN51024xfIKwKr9sSxcLejA7SVAvk/E91mJwo5limMfeWIipuoK',
email: 'shreykbhatt@gmail.com',
username: 'shreykbhatt',
__v: 0
Login successful
4 Nov 00:10:37 - displayalladvertiset_queue { sellerid: '5819ab6f7fd6571904b4d157' }
4 Nov 00:10:37 - Message: { "sellerid": "5819ab6f7fd6571904b4d157" }
4 Nov 00:10:37 - DeliveryInfo: { "contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "803f0b7e4e92f904cfcebbd70c2f9676", "replyTo": "amq.gen-gTXJwSeg9PcgCvXO" }
In handle request of Display all advertisement
All Advertisements Display successful
4 Nov 00:10:39 - getcart_queue { userid: '5819ab6f7fd6571904b4d157' }
4 Nov 00:10:39 - Message: { "userid": "5819ab6f7fd6571904b4d157" }
4 Nov 00:10:39 - DeliveryInfo: { "contentType": "application/json", "contentEncoding": "utf-8", "correlationId": "cfffa15c2730f62edfae4faa04e8129d0", "replyTo": "amq.gen-gTXJwSeg9PcgCvXO" }
In handle request of getcart
5819ab6f7fd6571904b4d157
Cart Display successful
[ { _id: '581bee940c68583fd4f0b33e',
  total: '120000',
  userid: '5819ab6f7fd6571904b4d157',
  price: '30000',
  quantity: '4',
  productname: 'Toyota',
  __v: 0 },
  { _id: '581c0a5527b01846d449c942',
  total: '1800',
  userid: '5819ab6f7fd6571904b4d157',
  price: '900',
  quantity: '2',
  productname: 'Camera',
  __v: 0 } ]

```

Event Log
180:1 CRLF: UTF-8
12:11 AM 11/4/2016

- **Last Login Page**
- Last Login time is stored in the user database. It will be updated everytime user logs out. So username and lastlogin time will be displayed in the following page.



- **My Orders**

- Orders of the current user will be displayed on the following page.

Order Date	Product Name	Quantity	Price	Total
Wed Oct 12 2016	WebCam	2	500	1000
Thu Oct 13 2016	WebCam	2	500	1000
Thu Oct 13 2016	Dell Laptop	7	700	4900
Sat Oct 15 2016	Macbook Pro	3	1000	3000

My Orders Client

```

[ 2] 2016-10-10T12:13:45.000Z - info: Listening on port 3001
  
```

My Orders Server

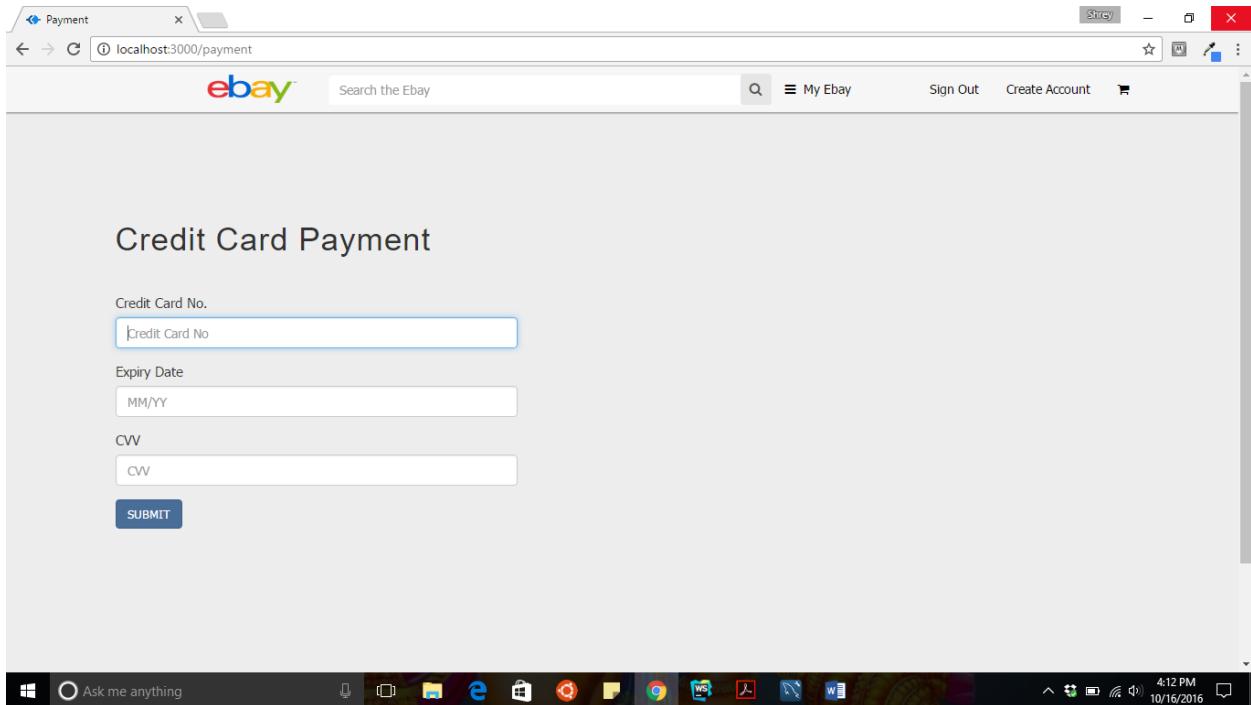
The screenshot shows the WebStorm IDE interface with the following details:

- Project Structure:** Shows the project tree with files like `server.js`, `bin/www`, and `biddinglogic.js`.
- Logs:** The terminal window displays log messages from the application. Key logs include:
 - 4 Nov 00:10:37 - Login successful
 - 4 Nov 00:10:37 - displayalladvertise_queue { sellerid: '5819ab6f7fd6571904b4d157' }
 - 4 Nov 00:10:37 - Message: {"sellerid":"5819ab6f7fd6571904b4d157"}
 - 4 Nov 00:10:37 - DeliveryInfo: {"contentType":"application/json","contentEncoding":"utf-8","correlationId":"803f0b7e4e92f904cfcebbd70c2f9676","replyTo":"amq.gen-gTXJwSeg9PcgCvX0"}
 - In handle request of Display all advertisement
 - All Advertisement Display successfull
 - 4 Nov 00:10:39 - getcart_queue { userid: '5819ab6f7fd6571904b4d157' }
 - 4 Nov 00:10:39 - Message: {"userid":"5819ab6f7fd6571904b4d157"}
 - 4 Nov 00:10:39 - DeliveryInfo: {"contentType":"application/json","contentEncoding":"utf-8","correlationId":"cffa15c2730f62edfae4faa04e8129d0","replyTo":"amq.gen-gTXJwSeg9PcgCvX0"}
 - In handle request of getcart
 - 5819ab6f7fd6571904b4d157
 - Cart Display successfull
 - [{ _id: 581bee940c68583fd4f0b33e,
 - total: '120000',
 - userid: '5819ab6f7fd6571904b4d157',
 - price: '30000',
 - quantity: '4',
 - productname: 'Toyota',
 - _v: 0,
 - { _id: 581c0a5527b01846d449c942,
 - total: '1800',
 - userid: '5819ab6f7fd6571904b4d157',
 - price: '900',
 - quantity: '2',
 - productname: 'Camera',
 - _v: 0,
- 4 Nov 00:13:00 - displayorders_queue { userid: '5819ab6f7fd6571904b4d157' }
- 4 Nov 00:13:00 - Message: {"userid":"5819ab6f7fd6571904b4d157"}
- 4 Nov 00:13:00 - DeliveryInfo: {"contentType":"application/json","contentEncoding":"utf-8","correlationId":"993f74e7e1e7dee29e4545a09abc6d4d","replyTo":"amq.gen-gTXJwSeg9PcgCvX0"}
- In handle request of Display my orders
- My Orders display successfull
- |

- Bottom Bar:** Shows standard Windows-style icons for Run, Terminal, and Task View.
- System Tray:** Shows the date and time (11/4/2016), battery level, and other system status indicators.

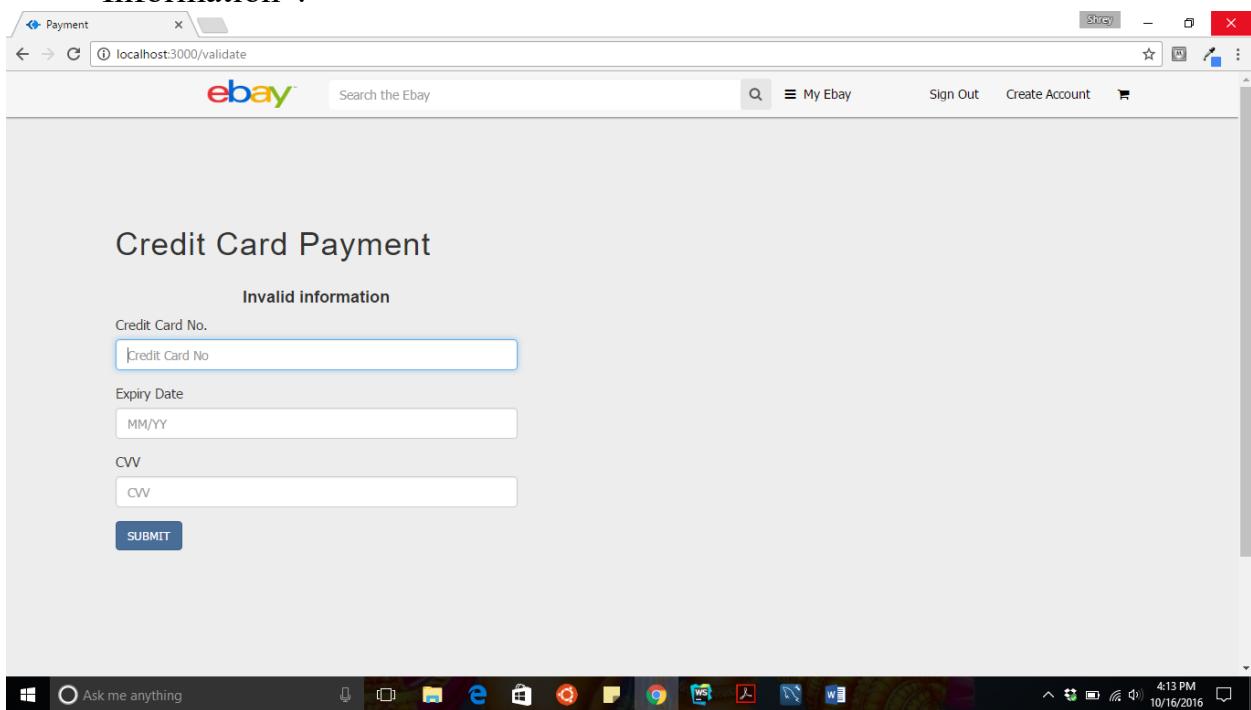
- Credit Cart Validation

- On clicking the checkout, the credit card payment page will be rendered.



The screenshot shows a web browser window titled "Payment" with the URL "localhost:3000/payment". The page has an eBay header with the logo and search bar. Below the header, the title "Credit Card Payment" is displayed. The form contains three input fields: "Credit Card No.", "Expiry Date", and "CVV", each with a placeholder text ("credit Card No.", "MM/YY", and "CVV" respectively). A "SUBMIT" button is located at the bottom left of the form. The browser's status bar at the bottom shows the date and time as 10/16/2016 4:12 PM.

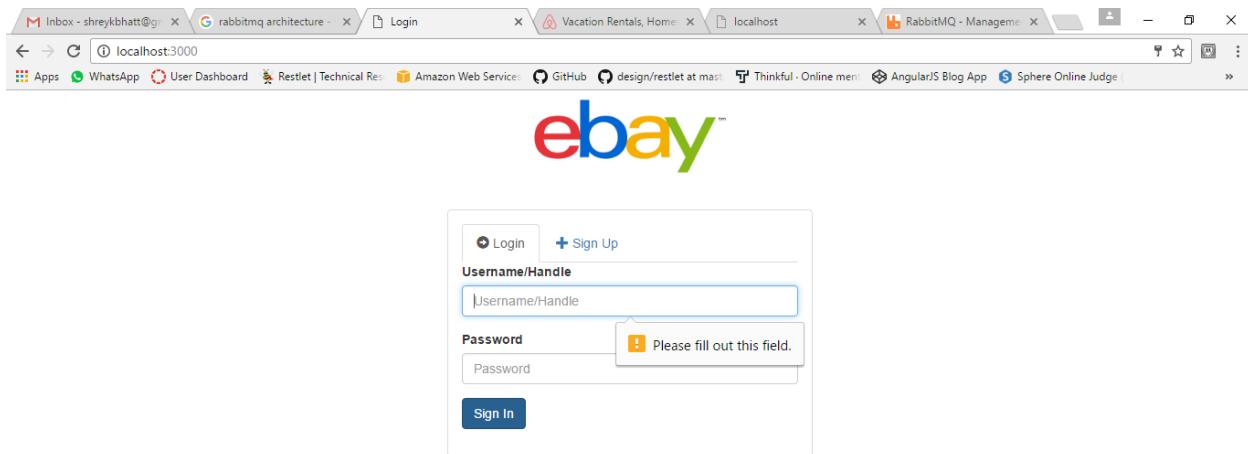
- Invalid Card**
- If the card info is invalid then it will show the message of “Invalid Information”.



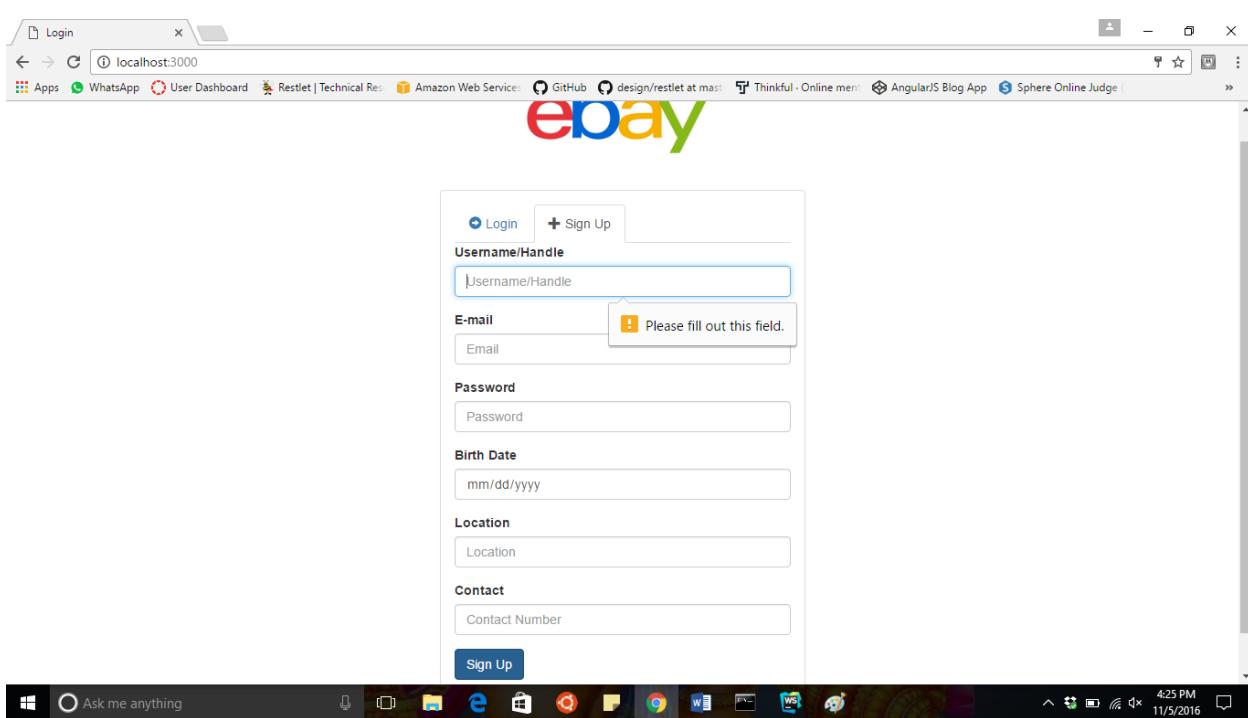
The screenshot shows a web browser window titled "Payment" with the URL "localhost:3000/validate". The page has an eBay header with the logo and search bar. Below the header, the title "Credit Card Payment" is displayed. The form contains three input fields: "Credit Card No.", "Expiry Date", and "CVV", each with a placeholder text ("credit Card No.", "MM/YY", and "CVV" respectively). Above the form, the message "Invalid information" is displayed. The browser's status bar at the bottom shows the date and time as 10/16/2016 4:13 PM.

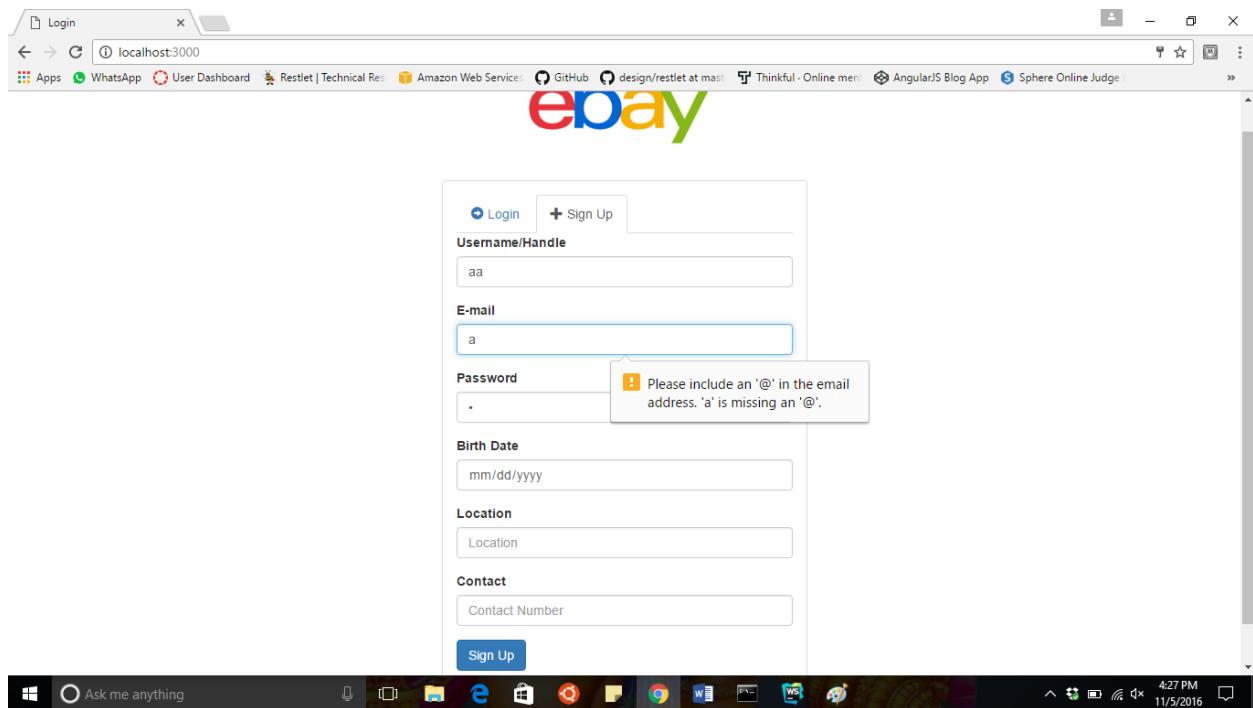
Validation

Login Screen

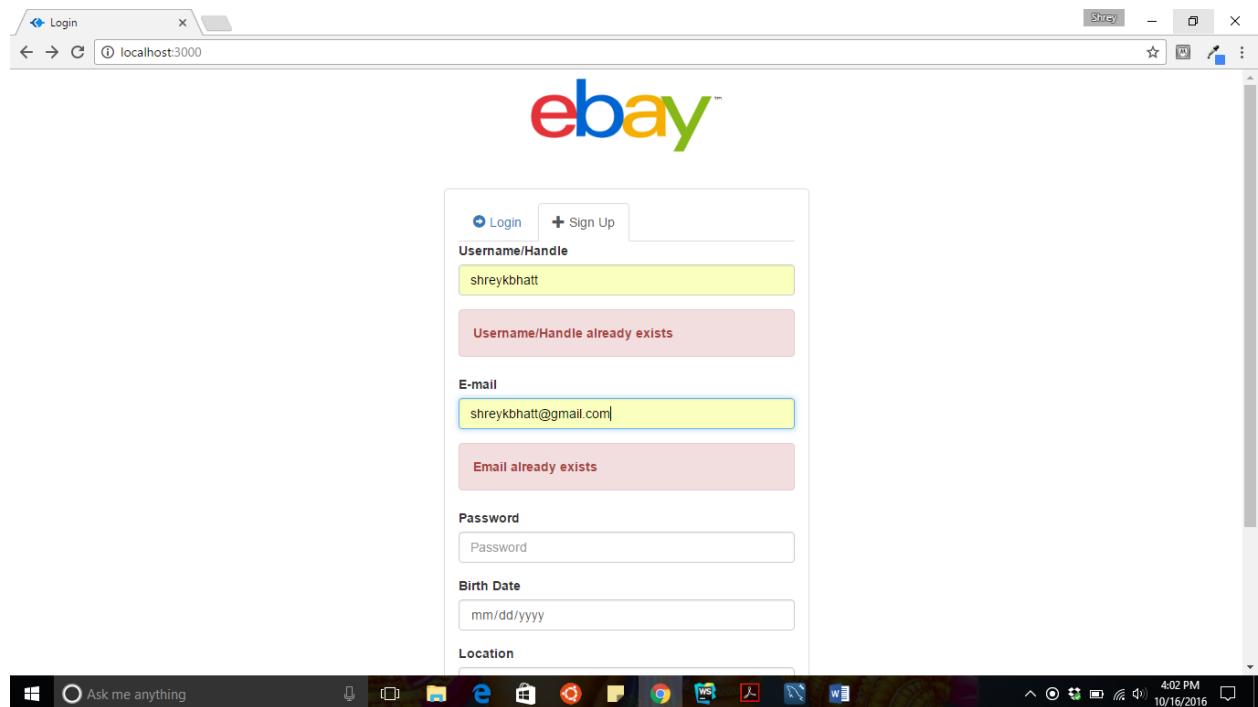


Sign Up

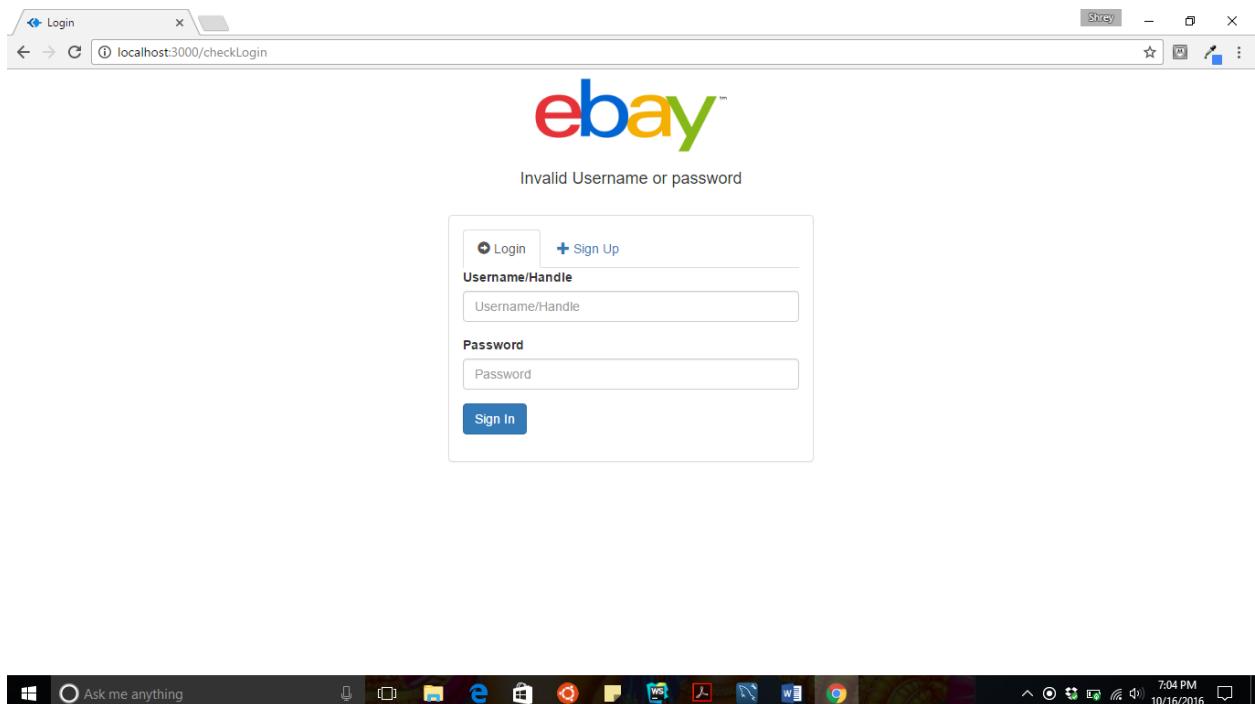




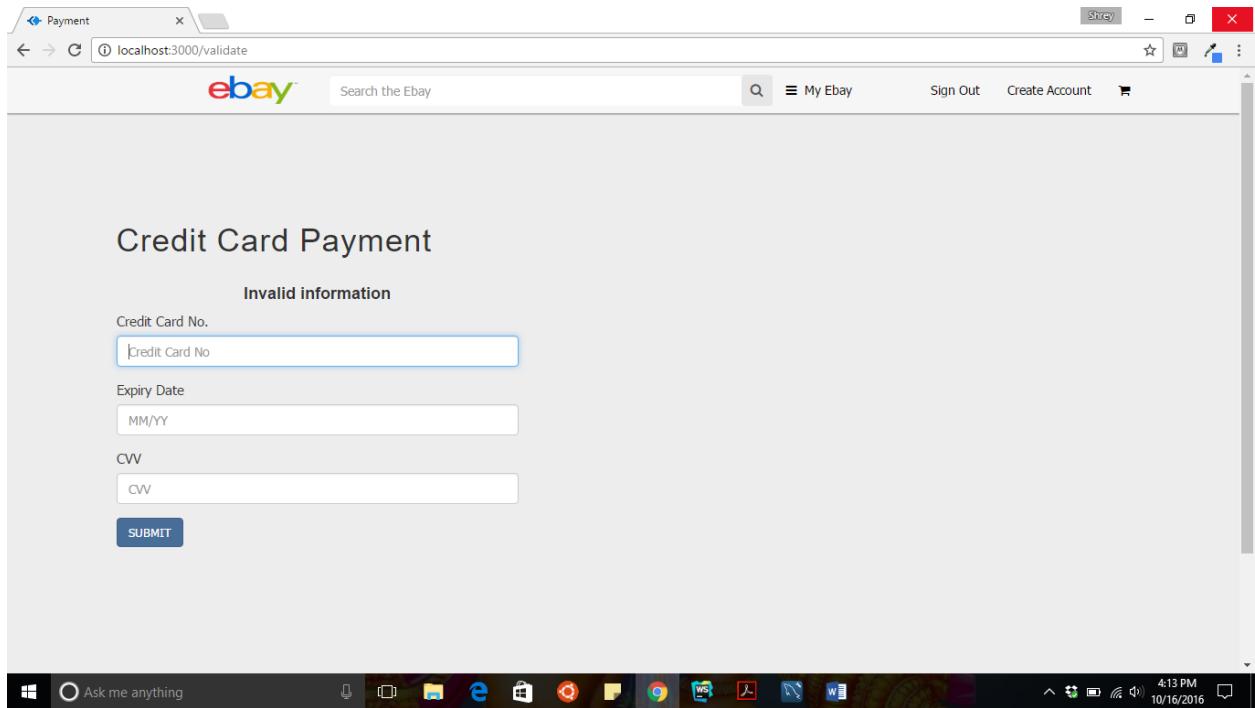
If Username and Email Already Exists



Invalid Username or password



Invalid Credit card



Exception Handling

Credit Card validation

The screenshot shows the WebStorm IDE interface with the project 'Ebay' open. The 'creditcard.js' file is the active editor. The code implements credit card validation logic, including regex checks for card number, expiration date, and CVV, and a try-catch block for rendering a success response or catching errors. The IDE's navigation bar, toolbars, and status bar are visible at the top and bottom respectively.

```
function validate(request, res, err) {
    var cardRegex=new RegExp("^(4[0-9]{12}|3[4|7][0-9]{13})\\s?","g");
    var expDateRegex=new RegExp("(0[1-9]|1[0-2])\\/(0[1-9]|1[0-2])\\/(0[1-9]{2}|2[0-4]\\d)$","g");
    var CVVRegex=new RegExp("^(0-9){3,4}$","g");
    var inputCVV = request.param("inputCVV");
    var inputExpDate = request.param("inputExpDate");
    var inputCardNo = request.param("inputCardNo");

    try{
        if(CVVRegex.test(inputCVV) && cardRegex.test(inputCardNo) && expDateRegex.test(inputExpDate)){
            checkData(request, res);
            log.log('info', request.session.username + " | CreditCard Validated | " + new Date().toString());

            res.render("paymentsuccess");
        }
        else{throw err}
    }catch (err){
        console.log(err);
    }
}

function emptycart(request, res) {
    var userid= request.session.userid;
    var msg_payload = {"userid": userid};

    mq_client.make_request('emptycart_queue', msg_payload, function(err, results){
        if(err){
            console.log(err);
        }
    });
}
```

The screenshot shows the WebStorm IDE interface with the project 'Ebay' open. The 'server.js' file is the active editor, displaying command-line logs from the terminal. The logs show the application starting on port 3000 and performing several 'setInterval' calls to update bid progress. The logs are color-coded with red for errors and green for informational messages.

```
"C:\Program Files (x86)\JetBrains\WebStorm 2016.2.2\bin\runner.exe" "C:\Program Files\nodejs\node.exe" C:\Users\Shrey\WebstormProjects\Ebay\bin\www
Ebay:server Listening on port 3000 +0ms
Sat, 05 Nov 2016 03:12:43 GMT express deprecated req.param(name): Use req.params, req.body, or req.query instead at Ebay_Client\routes\creditcard.js:10:28
Sat, 05 Nov 2016 03:12:43 GMT express deprecated req.param(name): Use req.params, req.body, or req.query instead at Ebay_Client\routes\creditcard.js:11:32
Sat, 05 Nov 2016 03:12:43 GMT express deprecated req.param(name): Use req.params, req.body, or req.query instead at Ebay_Client\routes\creditcard.js:12:31
[Function: next]
setInterval called
hello
The Bid is still under progress for: 581c08910c68583fd44f0b33f
The Bid is still under progress for: 581c08a90c68583fd44f0b340
The Bid is still under progress for: 581c09ba0c68583fd44f0b341
The Bid is still under progress for: 581c09cc0c68583fd44f0b342
The Bid is still under progress for: 581c09de0c68583fd44f0b343
The Bid is still under progress for: 581c09fa0c68583fd44f0b344
The Bid is still under progress for: 581c0910c68583fd44f0b345
The Bid is still under progress for: 581c0920c68583fd44f0b346
The Bid is still under progress for: 581c09450c68583fd44f0b347
The Bid is still under progress for: 581c09570c68583fd44f0b348
The Bid is still under progress for: 581c09720c68583fd44f0b349
The Bid is still under progress for: 581c09840c68583fd44f0b34a
The Bid is still under progress for: 581c33add8cd94ebc3ce25d
```

Sign Up

The screenshot shows the WebStorm IDE interface with the project 'Ebay' open. The current file is 'routes/login.js'. The code handles user registration and sign-up logic. It uses moment.js to format dates and bcrypt for password hashing. A try-catch block is used to handle errors from the message queue client. The code then renders a template ('login.ejs') or logs an exception.

```
var contact=request.body.contact;
// var encrypt=bcrypt.hashSync(password);

var lastlogin=moment().format('MM DD YYYY, h:mm:ss a');

var msg_payload = { "username": username, "email": email, "password": password, "birthdate": birthdate, "location": location, "contact": contact };

try {
    mq_client.make_request('register_queue', msg_payload, function (err, results) {
        if (err) {
            console.log(err);
        }
        else {
            if (results.code == 200) {
                console.log("Successful SignUp");
            }
            else if (results.code == 0) {
                console.log("Unsuccessful SignUp");
            }
        }
    });
    res.render('login.ejs', {invalid: ""});
} catch (Exception) {
    console.log("Exception in SignUp");
}
```

Login

The screenshot shows the WebStorm IDE interface with the project 'Ebay' open. The current file is 'routes/login.js'. The code implements passport authentication for the 'login' route. It checks for successful login, invalid credentials, or database errors. It also handles the sign-out process. The code uses req, res, and next parameters for middleware.

```
try {
    passport.authenticate('login', function (err, results, info) {
        if (err) {
            console.log(err);
        }
        else {
            if (results.code == 200) {
                console.log("Successful Login");
                console.log("will renderfile");
                console.log(results);
                req.session.username = results.data[0].username;
                req.session.userid = results.data[0]._id;
                console.log(req.session.userid);
                return res.render('homepage.ejs');
            }
            else if (results.code == 400) {
                return res.render('login', {invalid: "Invalid Username or password"});
            }
            else if (results.code == 0) {
                console.log("DB Operation Failed");
            }
        }
    })(req, res, next);
} catch (Exception) {
    console.log(Exception);
}

exports.signout=function(request,res)
```

Entering Amount for Bidding less than the previous bid

Ebay - [C:\Users\Shrey\WebstormProjects\Ebay] - ...\\Ebay_Client\\routes\\bid.js - WebStorm 2016.2.4

File Edit View Navigate Code Refactor Run Tools VCS Window Help

Ebay Ebay_Client routes bid.js

Project Structure

1-Project

2-Structure

3-Favorites

creditcard.js advertisement.js bid.js

server.js

123 var update = {
124 'bidprice': bidprice,
125 'bidderid': request.session.userid
126 };
127 try{
128 var msg_payload = { "conditions": conditions, "update": update};
129
130 mq_client.make_request('bidding_queue', msg_payload, function(err, results){
131 if(err){
132 console.log(err);
133 //throw err;
134 } else {
135 if(results.code == 200){
136 log.info(request.session.username + " | Bid price updated | " + new Date().toString());
137
138 console.log("Bid price updated");
139
140 res.render("allbids");
141 else if (results.code == 0){
142 console.log("Unsuccessful update of bid price");
143 }
144 })} catch(RangeException){
145 console.log("Exception");
146 }
147 biddinghistory();
148 }
149 }
150
151 }
152
153
154
155

Event Log

701:1 CRLF: UTF-8: 4:49 PM 11/5/2016

Ask me anything

Run TODO Terminal

Adding Quantity More than the Available Quantity

The screenshot shows the WebStorm IDE interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "routes". The file "productdetail.js" is currently selected.
- Code Editor:** The main area displays the "productdetail.js" code. The code handles a POST request to add a product to the cart. It includes logic to validate the quantity and handle errors. A yellow highlight covers the line "76 } catch (Exception){".
- Toolbars and Status Bar:** The top bar has standard file menu items like File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help. The status bar at the bottom right shows "76:34 CRLF: UTF-8" and the date/time "11/5/2016 4:55 PM".

```
var price=request.session.price;
var userid=request.session.userid;
var total=total;

try{
var msg_payload = {"productid": productid,"productname": productname,"quantity": quantity,"price": price,"userid": userid,"total": total};

mq_client.make_request('addcart_queue', msg_payload, function(err, results){
if(err){
    console.log(err);
}
else
{
    if(results.code == 200){

        console.log("product added to cart");
        log.info('info', request.session.username + " | Product Added to cart | " + new Date().toString());

        console.log(results);
        res.render("shopping-cart");
    }
    else if (results.code == 0){

        console.log("Unsuccessful cart addition");
    }
}
});catch (Exception){
    console.log("Exception");
}
};
```

Database Collections

User

The screenshot shows the Robomongo interface connected to a MongoDB instance at localhost:27017. The left sidebar shows a tree view of the database structure under 'Ebay'. The 'users' collection is selected. The main pane displays the results of the query `db.getCollection('users').find({})`. The results are listed in a table with columns: Key, Value, and Type. There are three documents returned:

Key	Value	Type
(1) ObjectId("5819ab6f7fd6571904b4d157")	<pre>{ _id: ObjectId("5819ab6f7fd6571904b4d157"), lastlogin: "2016-04-04T12:29:00Z", contact: "12345678901234", location: "San Jose, California", birthdate: "2016-11-14T00:00:00.000Z", password: "\$2a\$10\$jsN1Oz4xFkwKr9a5xcLejA7SVAvk/E91mJwwo5imMfeWT...", email: "shreykbhatt@gmail.com", username: "shreykbhatt", __v: 0}</pre>	Object
(2) ObjectId("581a4f67a9ff770604e69e4c")	<pre>{ _id: ObjectId("581a4f67a9ff770604e69e4c"), lastlogin: "2016-11-03T09:21:00Z", contact: "12345678901234", location: "San Jose, California", birthdate: "2016-11-28T00:00:00.000Z", password: "\$2a\$10\$vn2nINdZ2b9BBdC5DlyHoepNqWOTKdB8xaoGo4q0lBa...", email: "parth@gmail.com", username: "parth", __v: 0}</pre>	Object
(3) ObjectId("581b012498b1724024415ea")	<pre>{ _id: ObjectId("581b012498b1724024415ea"), lastlogin: "2016-11-28T21:32:00Z", contact: "12345678901234", location: "San Jose, California", birthdate: "2016-11-28T00:00:00.000Z", password: "test"}</pre>	Object

Product

The screenshot shows the Robomongo interface connected to a MongoDB instance at localhost:27017. The left sidebar shows a tree view of the database structure under 'Ebay'. The 'products' collection is selected. The main pane displays the results of the query `db.getCollection('products').find({})`. The results are listed in a table with columns: Key, Value, and Type. There are 11 documents returned:

Key	Value	Type
(1) ObjectId("5812d8f133abea220c2cad89")	<pre>{ _id: ObjectId("5812d8f133abea220c2cad89"), shippingfrom: "SF", sellerid: "5812d8f033abea220c2cad88", description: "256 GB 3 GHZ", price: "1500", quantity: "3", productname: "Macbook Pro", __v: 0}</pre>	Object
(2) ObjectId("58145b6349452d263c654e91")	<pre>{ _id: ObjectId("58145b6349452d263c654e91"), shippingfrom: "MA", sellerid: "5812d8f7447d3461e60ced16", description: "BMW", price: "50000", quantity: "3", productname: "Car", __v: 0}</pre>	Object
(3) ObjectId("5817f252924db301434fea5")	<pre>{ _id: ObjectId("5817f252924db301434fea5")}</pre>	Object
(4) ObjectId("581a52e1a9ff770604e69e4d")	<pre>{ _id: ObjectId("581a52e1a9ff770604e69e4d")}</pre>	Object
(5) ObjectId("581a530ca9ff770604e69e4e")	<pre>{ _id: ObjectId("581a530ca9ff770604e69e4e")}</pre>	Object
(6) ObjectId("581a5325a9ff770604e69e4f")	<pre>{ _id: ObjectId("581a5325a9ff770604e69e4f")}</pre>	Object
(7) ObjectId("581a533aa9ff770604e69e50")	<pre>{ _id: ObjectId("581a533aa9ff770604e69e50")}</pre>	Object
(8) ObjectId("581a5335a9ff770604e69e51")	<pre>{ _id: ObjectId("581a5335a9ff770604e69e51")}</pre>	Object
(9) ObjectId("581a5381a9ff770604e69e52")	<pre>{ _id: ObjectId("581a5381a9ff770604e69e52")}</pre>	Object
(10) ObjectId("581a539fa9ff770604e69e53")	<pre>{ _id: ObjectId("581a539fa9ff770604e69e53")}</pre>	Object
(11) ObjectId("581a53d4a9ff770604e69e54")	<pre>{ _id: ObjectId("581a53d4a9ff770604e69e54")}</pre>	Object

Cart

The screenshot shows the Robomongo interface with the 'Ebay' database selected. The left sidebar shows collections: System, Ebay (with Collections: biddinghistories, bids, carts, orders, products, sessions, users), Functions, Users, ebay, and login_demo. The right panel displays the contents of the 'carts' collection. A query bar at the top shows 'db.getCollection("carts").find({})'. The results table has columns: Key, Value, and Type. One document is shown:

Key	Type
ObjectID("581c0a5527b01846d449c942")	Object
_id	ObjectId
total	String
userid	String
price	String
quantity	String
productname	String
__v	Int32

The 'total' field contains '1800', 'userid' contains '5819ab6f7fd6571904b4d157', 'price' contains '900', 'quantity' contains '2', 'productname' contains 'Camera', and '__v' contains '0'.

Order

The screenshot shows the Robomongo interface with the 'Ebay' database selected. The left sidebar shows collections: System, Ebay (with Collections: biddinghistories, bids, carts, orders, products, sessions, users), Functions, Users, ebay, and login_demo. The right panel displays the contents of the 'orders' collection. A query bar at the top shows 'db.getCollection("orders").find({})'. The results table has columns: Key, Value, and Type. Three documents are shown:

Key	Type
(1) ObjectID("58195b5af0d039e0ac8a487f5")	Object
_id	ObjectId
userid	String
date	String
total	String
price	String
quantity	String
productname	String
__v	Int32

Document 1: total=36000, price=3600, quantity=1, productname=BMW, __v=0.

Key	Type
(2) ObjectID("58195bc0fd039e0ac8a487f6")	Object
_id	ObjectId
date	String
total	String
price	String
quantity	String
productname	String
userid	String
__v	Int32

Document 2: total=1500, price=500, quantity=3, productname=Webcam, userid=5812d87447d3461e60ced16e, __v=0.

Key	Type
(3) ObjectID("581d4cff874e172a14baa9fe")	Object
_id	ObjectId
date	String
total	String
price	String
quantity	String
productname	String
userid	String
__v	Int32

Document 3: total=2700, price=900, quantity=3, productname=Mobile, userid=5819ab6f7fd6571904b4d157, __v=0.

Bid

Robomongo 0.9.0-RC10

File View Options Window Help

New Connection (5)

Ebay

Collections (7)

- biddinghistories
- bids
- carts
- orders
- products
- sessions
- users

Functions

Users

ebay

login_demo

localhost:27017 Ebay

db.getCollection('bids').find({})

bids 0.001 sec.

Key	Value	Type
(1) ObjectId("581983209d464b2bb0cf9f38")	{ 11 fields }	Object
_id	ObjectId("581983209d464b2bb0cf9f38")	Objectid
bidderid	5812a8d033abea220c2cad88	String
isBidEnded	true	Boolean
bidendtime	1478067006896	String
bidstarttime	1478066976896	String
bidshippingfrom	NJ	String
bidSellerId	5812d87447d3461e60ced16e	String
bidDescription	Hi	String
bidPrice	950	String
bidProductName	Washing Machine	String
__v	0	Int32
(2) ObjectId("581c08910c68583fd4f0b33f")	{ 11 fields }	Object
_id	ObjectId("581c08910c68583fd4f0b33f")	Objectid
bidderid	null	Null
isBidEnded	false	Boolean
bidendtime	1478577809592	String
bidstarttime	1478232209592	String
bidShippingFrom	NJ	String
bidSellerId	581a4f67a9ff770604e69e4c	String
bidDescription	High Speed	String
bidPrice	30000	String
bidProductName	BMW	String
__v	0	Int32
(3) ObjectId("581c08a90c68583fd4f0b340")	{ 11 fields }	Object
(4) ObjectId("581c08ba0c68583fd4f0b341")	{ 11 fields }	Object
(5) ObjectId("581c08cc0c68583fd4f0b342")	{ 11 fields }	Object

Logs

Ask me anything

12:50 PM 11/6/2016

Connection pooling

```
var mongoose = require('mongoose');

var session = require('client-sessions');
var mongoSessionConnectURL = "mongodb://localhost:27017/Ebay";
var expressSession = require("express-session");
var mongoStore = require("connect-mongo")(expressSession);

mongoose.connect(mongoSessionConnectURL, {server: {poolSize: 10}});

var mongoose = require('mongoose');

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

mongoose.connect(mongoSessionConnectURL, {server: {poolSize: 10}});
mongoose.connection.on('connected', function() {
  console.log('Mongoose connected to ' + mongoSessionConnectURL);
});
mongoose.connection.on('error', function(err) {
  console.log('Mongoose connection error: ' + err);
});
mongoose.connection.on('disconnected', function() {
  console.log('Mongoose disconnected');
});

gracefulShutdown = function(msg, callback) {
  mongoose.connection.close(function() {
    console.log('Mongoose disconnected through ' + msg);
    callback();
  });
};

process.once('SIGUSR2', function() {
  gracefulShutdown('nodemon restart', function() {
    process.kill(process.pid, 'SIGUSR2');
  });
});

process.on('SIGINT', function() {
  gracefulShutdown('app termination', function() {
    process.exit(0);
  });
});
```

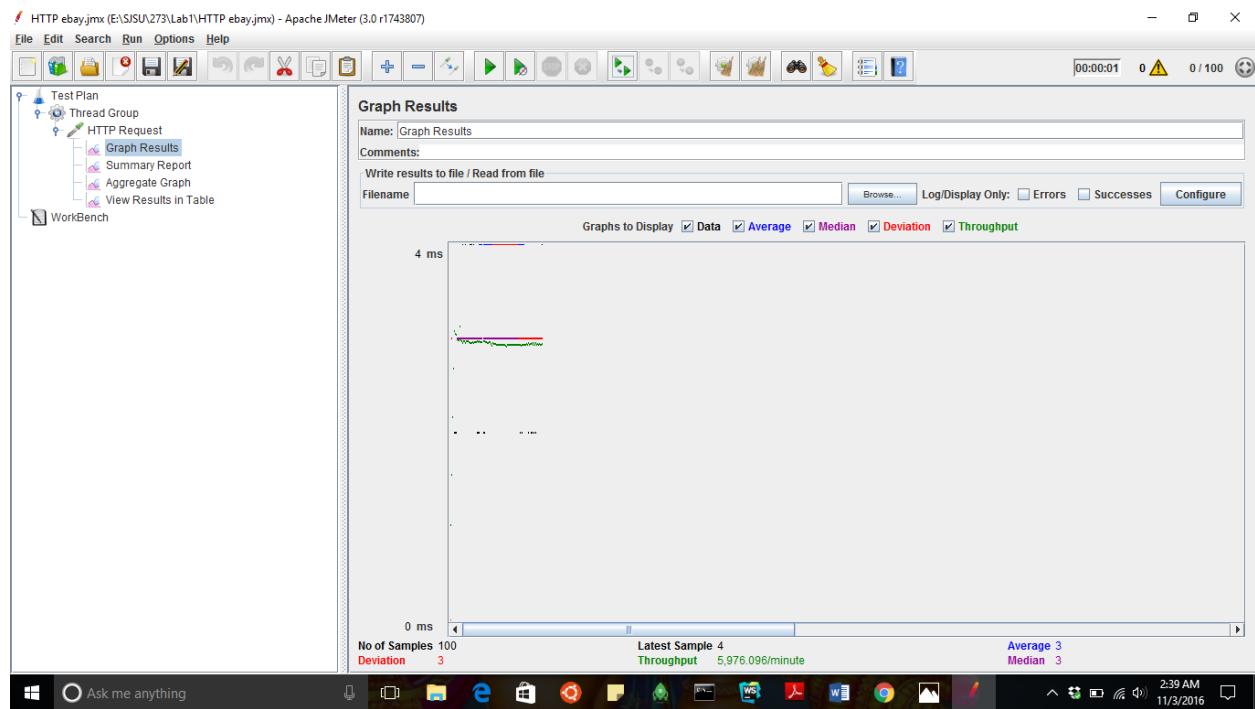
```
});  
});
```

Testing

- **JMeter Screenshots**

1. With Connection Pooling

1) 100 concurrent users



HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Summary Report

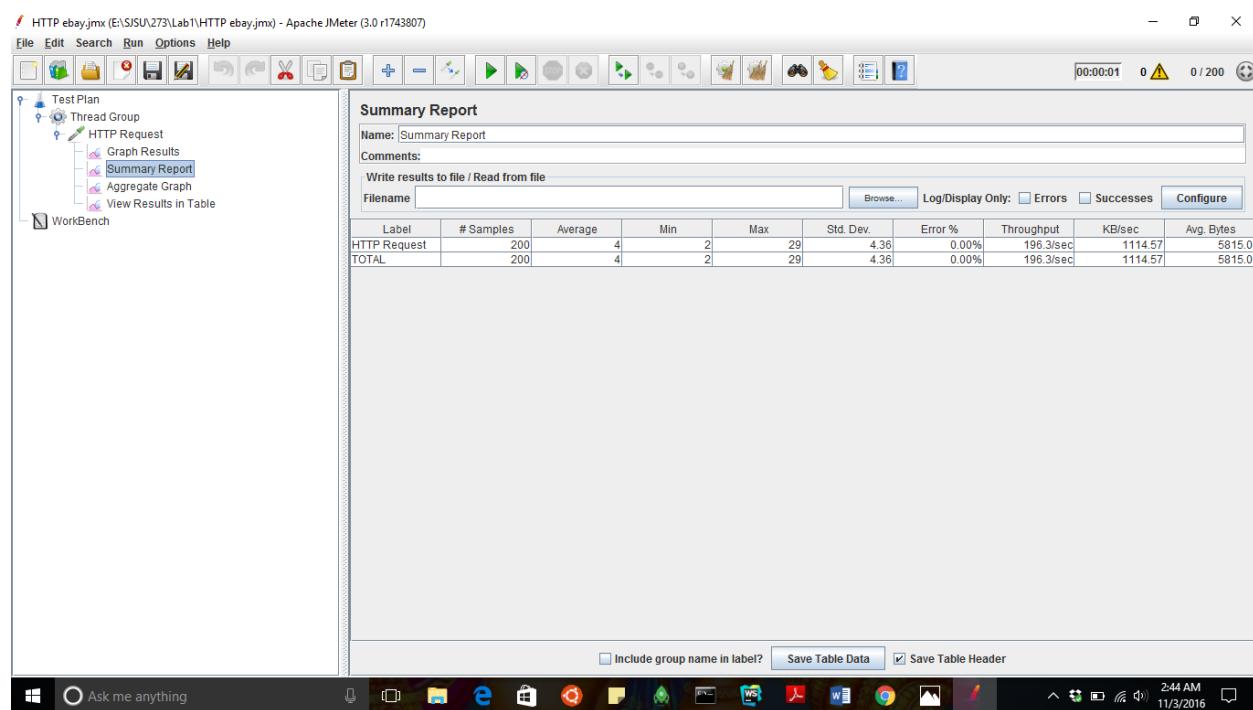
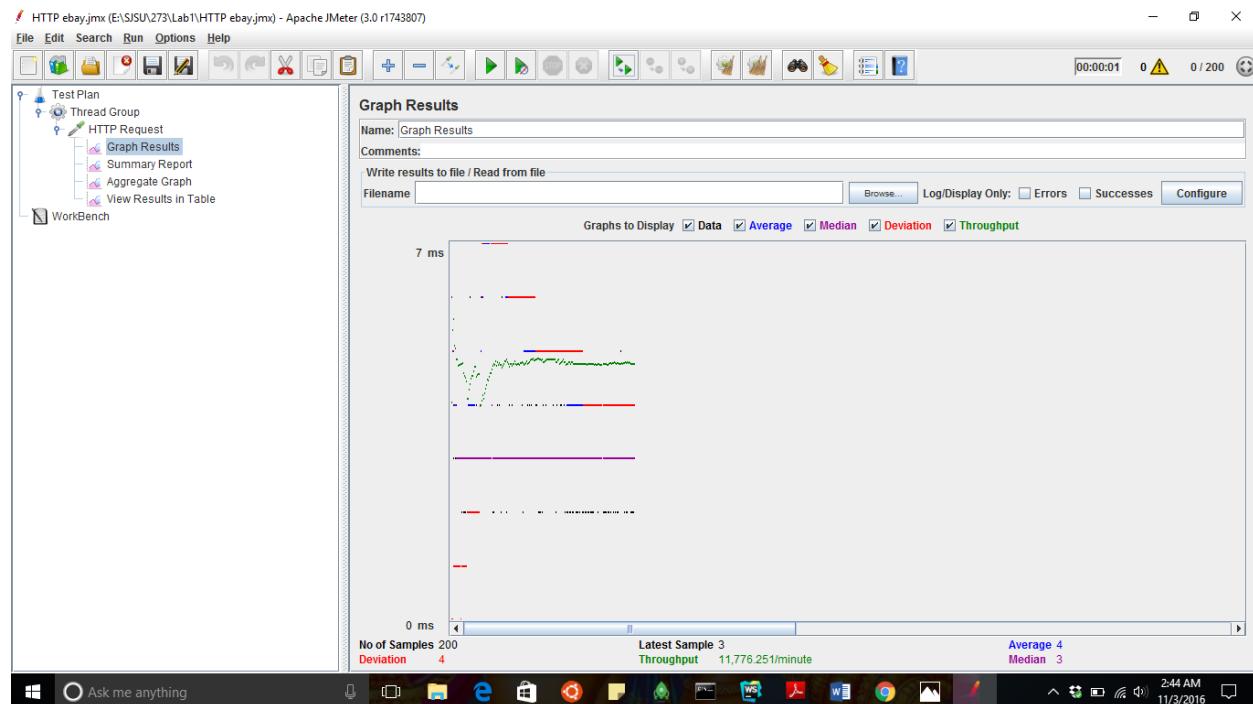
Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	100	3	2	29	3.49	0.00%	99.6/sec	565.61	5815.0
TOTAL	100	3	2	29	3.49	0.00%	99.6/sec	565.61	5815.0

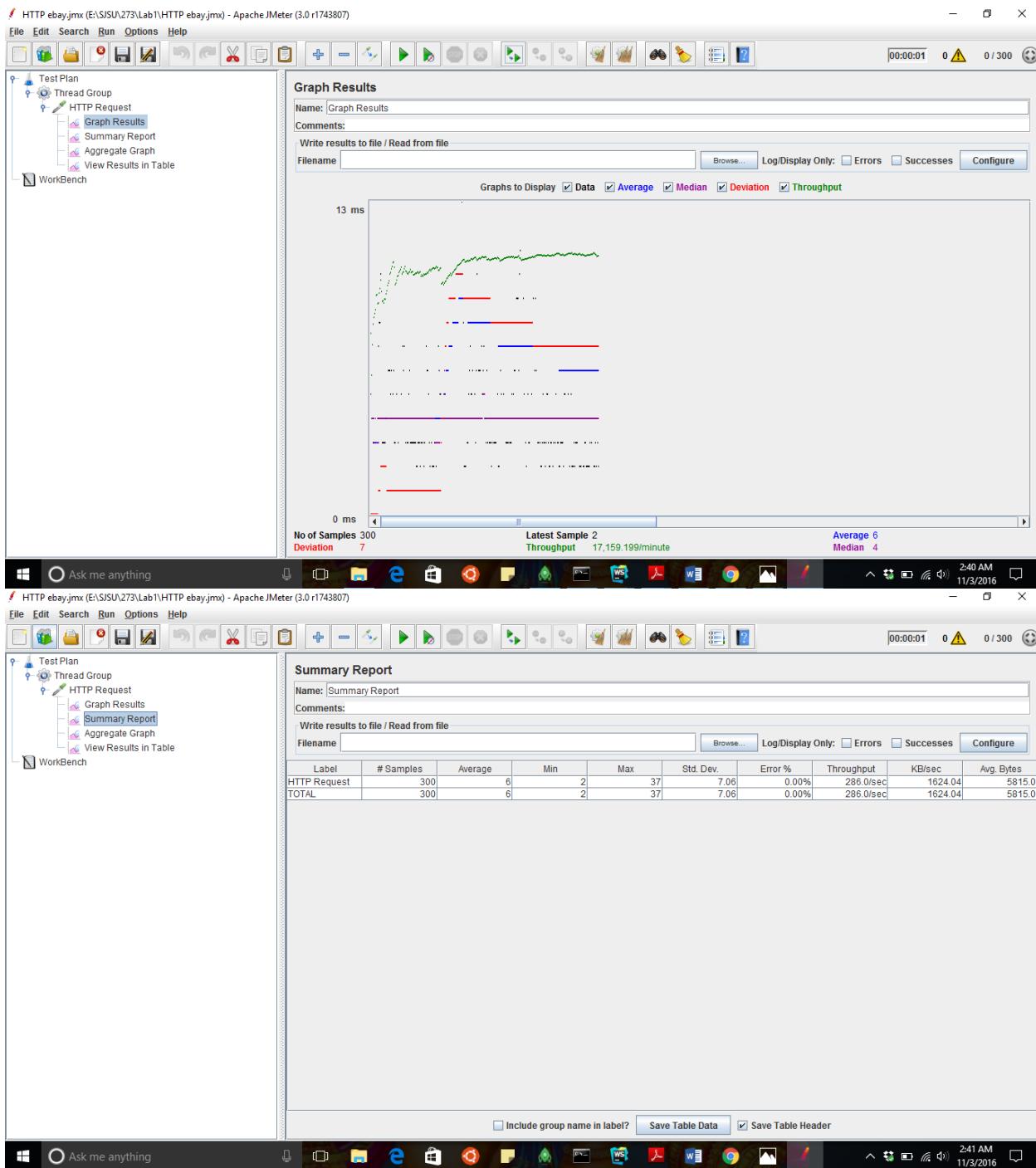
Include group name in label? Save Table Data Save Table Header

Ask me anything 2:40 AM 11/3/2016

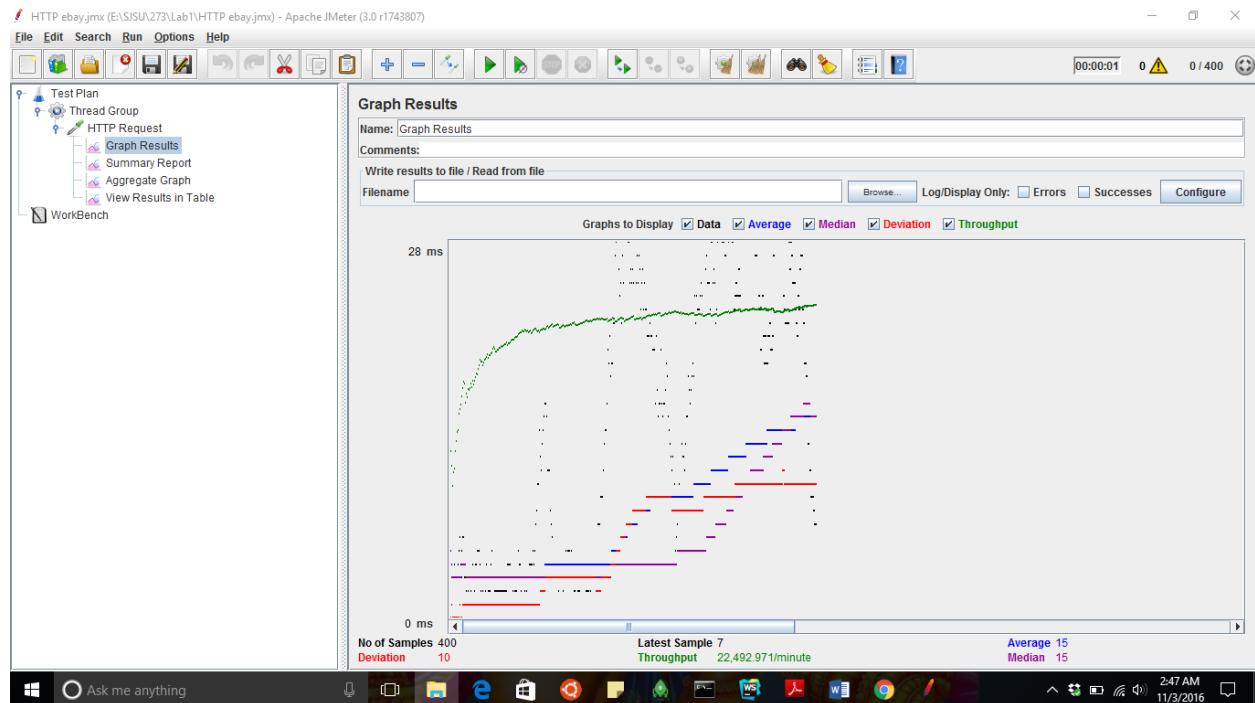
2) 200 concurrent users



3) 300 concurrent users



4) 400 concurrent users



HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Summary Report

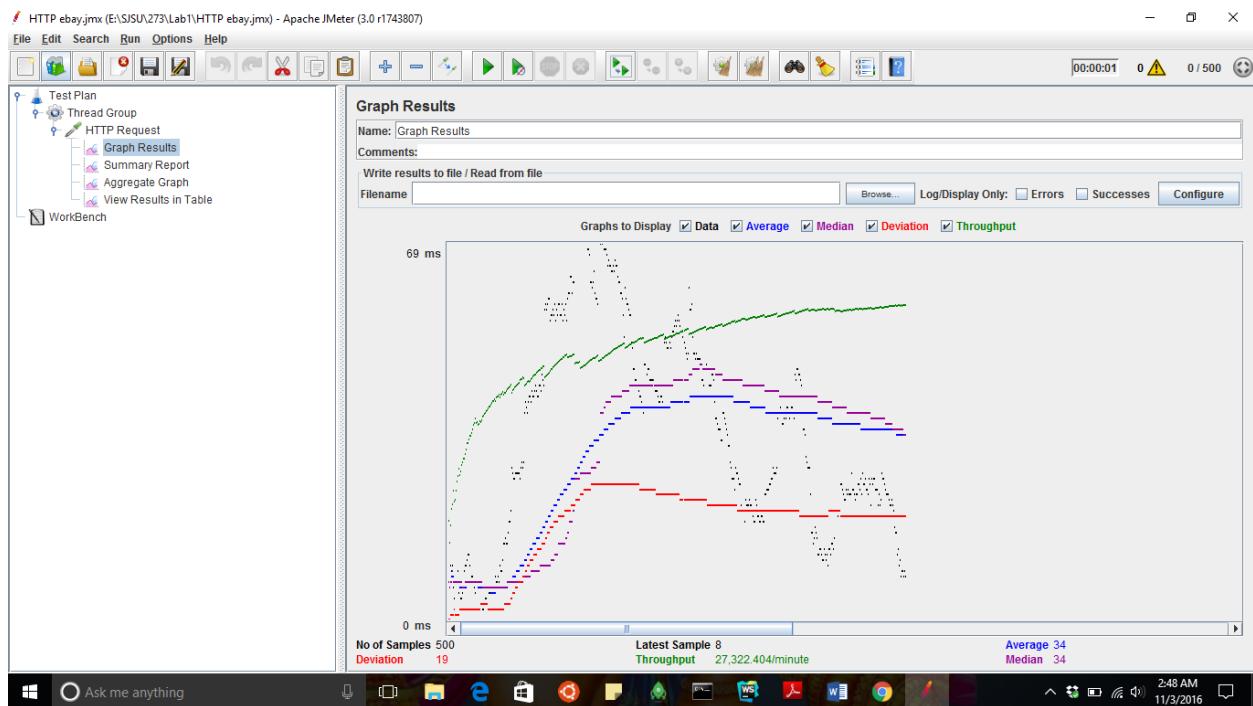
Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

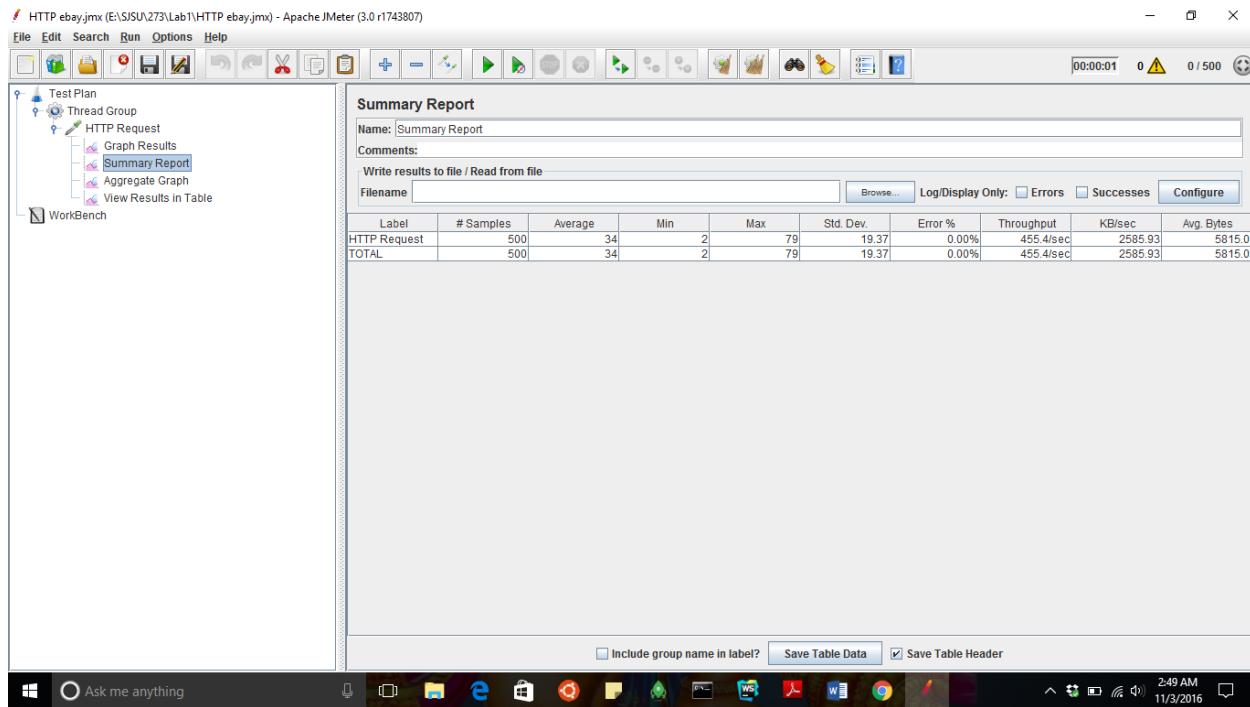
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	400	15	2	38	10.79	0.00%	374.9/sec	2128.85	5815.0
TOTAL	400	15	2	38	10.79	0.00%	374.9/sec	2128.85	5815.0

Include group name in label? Save Table Data Save Table Header

Ask me anything 2:47 AM 11/3/2016

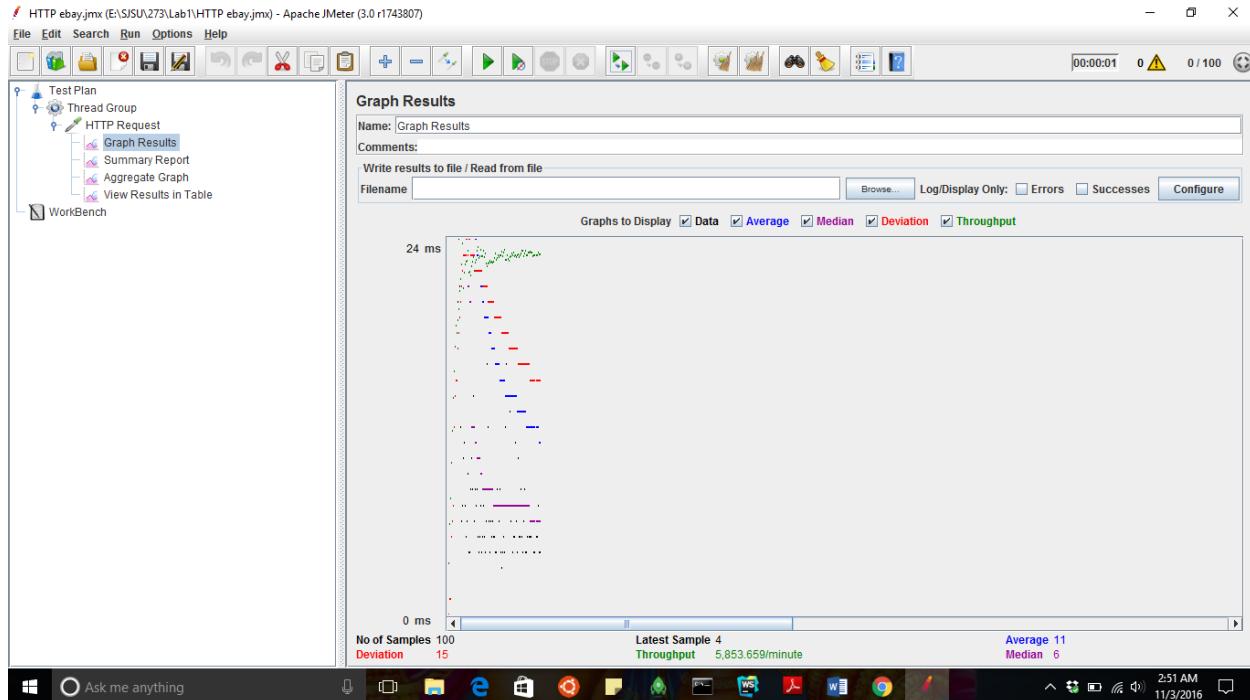
5) 500 concurrent Users





2. Without Connection Pooling

1) 100 concurrent users



HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Summary Report

Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	100	11	3	74	15.34	0.00%	97.6/sec	554.02	5815.0
TOTAL	100	11	3	74	15.34	0.00%	97.6/sec	554.02	5815.0

Include group name in label? Save Table Data Save Table Header

Ask me anything 2:51 AM 11/3/2016

2) 200 concurrent users

HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Graph Results

Name: Graph Results
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Graphs to Display Data Average Median Deviation Throughput

61 ms
0 ms
No of Samples 200 Deviation 25 Latest Sample 3 Throughput 11,214.953/minute Average 15 Median 4

Ask me anything 2:52 AM 11/3/2016

HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan

Thread Group

HTTP Request

Graph Results

Summary Report

Aggregate Graph

View Results in Table

WorkBench

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes Configure

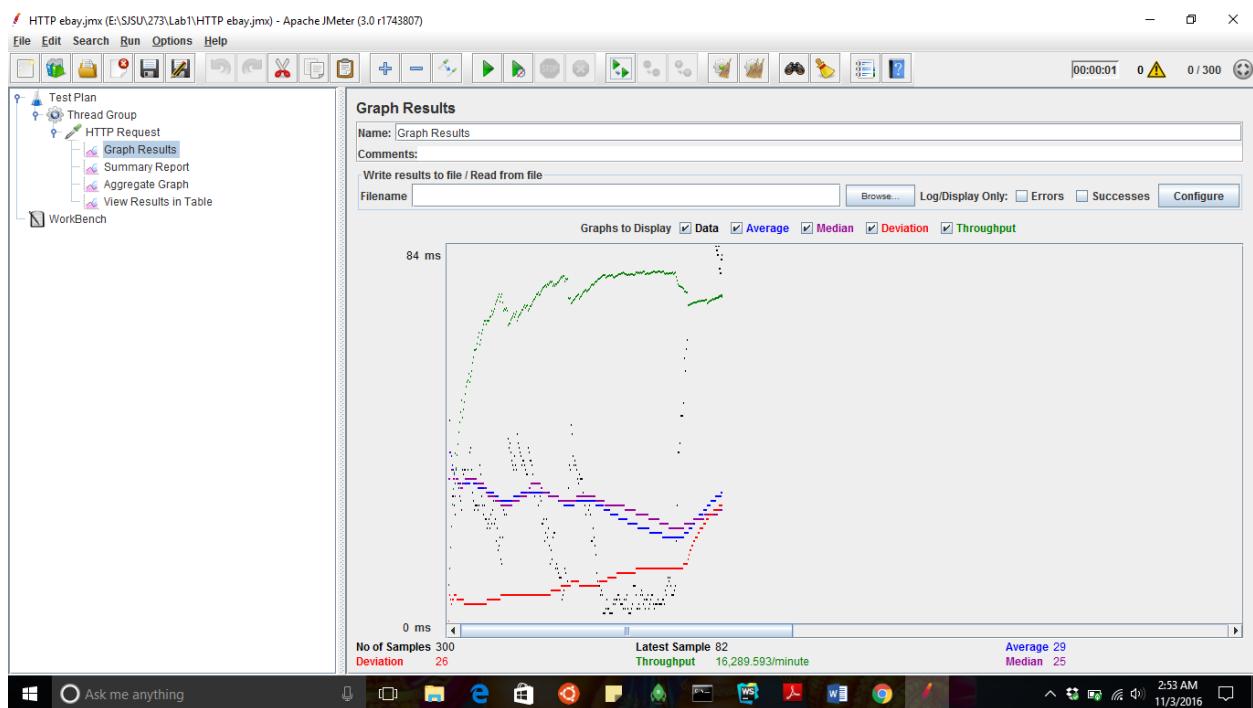
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	200	15	2	115	25.60	0.00%	186.9/sec	1061.44	5815.0
TOTAL	200	15	2	115	25.60	0.00%	186.9/sec	1061.44	5815.0

Include group name in label? Save Table Data Save Table Header

Ask me anything 2:52 AM 11/3/2016

00:00:01 0 0 / 200

3) 300 concurrent users



HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan

- Thread Group
 - HTTP Request
 - Graph Results
 - Summary Report
 - Aggregate Graph
 - View Results in Table
- WorkBench

Summary Report

Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	300	29	2	101	26.47	0.00%	271.5/sec	1541.73	5815.0
TOTAL	300	29	2	101	26.47	0.00%	271.5/sec	1541.73	5815.0

Include group name in label? Save Table Data Save Table Header

Ask me anything 2:53 AM 11/3/2016

4) 400 concurrent users

HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan

- Thread Group
 - HTTP Request
 - Graph Results
 - Summary Report
 - Aggregate Graph
 - View Results in Table
- WorkBench

Graph Results

Name: Graph Results
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Graphs to Display Data Average Median Deviation Throughput

138 ms
0 ms
No of Samples 400
Deviation 45
Latest Sample 10
Throughput 22,813.688/minute
Average 52
Median 39

Ask me anything 2:54 AM 11/3/2016

HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan

- Thread Group
 - HTTP Request
 - Graph Results
 - Summary Report
 - Aggregate Graph
 - View Results in Table
- WorkBench

Summary Report

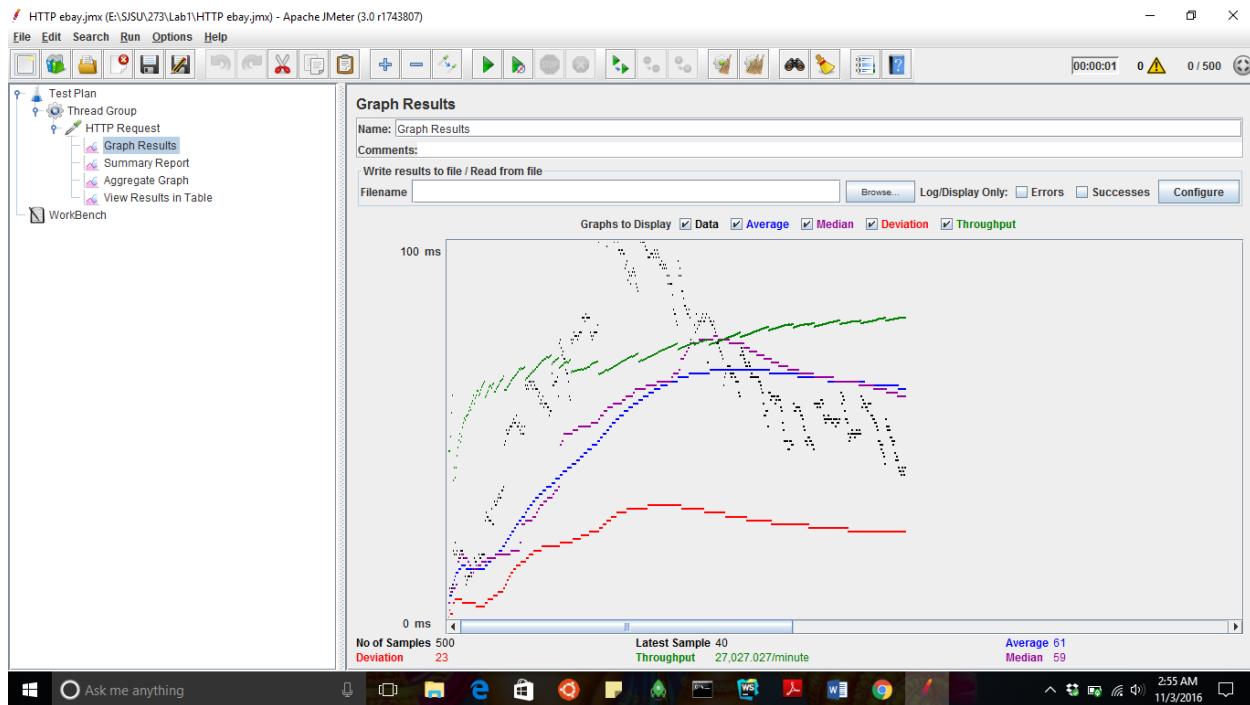
Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

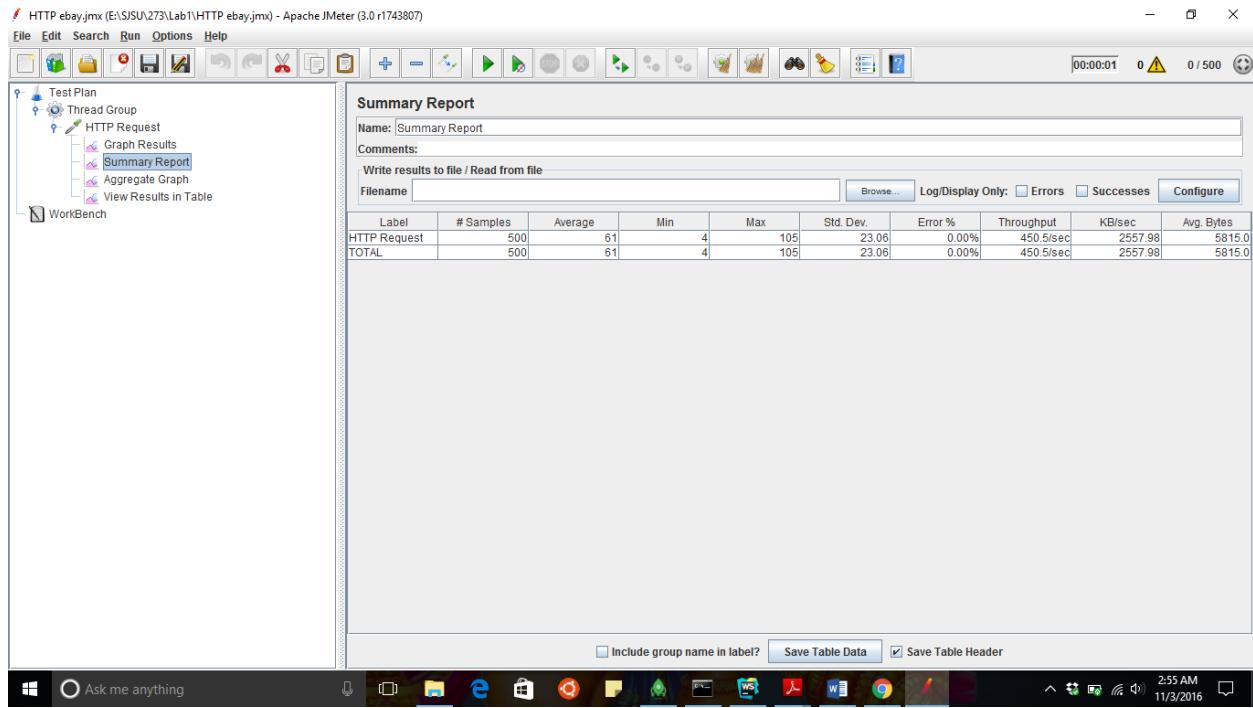
Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	400	52	2	150	45.49	0.00%	380.2/sec	2159.21	5815.0
TOTAL	400	52	2	150	45.49	0.00%	380.2/sec	2159.21	5815.0

Include group name in label? Save Table Data Save Table Header

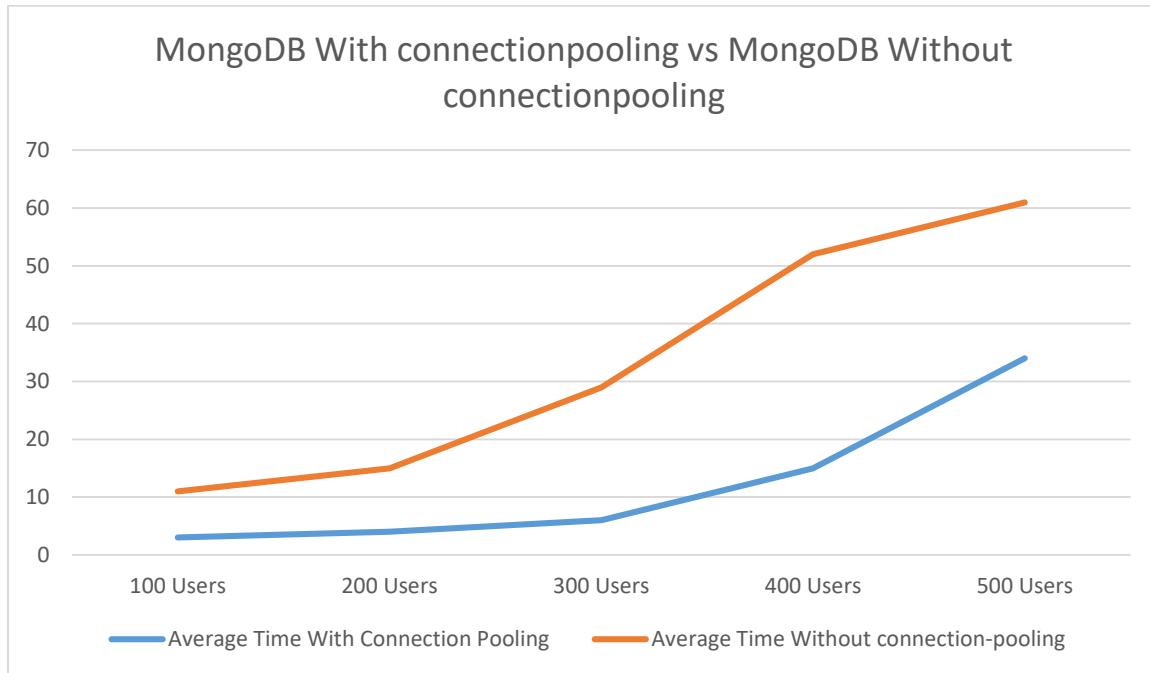
Ask me anything 2:55 AM 11/3/2016

5) 500 concurrent users





• Analysis Graph



Explanation

It can be observed from the graphs shown above that if the connection pooling is implemented in the application then the average time for request and response decreases to a considerable amount. When multiple concurrent users are requesting for resource; then connection pooling will be very useful because it reduces the number of times new connections are created. Therefore, the process of getting a connection also becomes faster.

In my application I have made a pool which allocates connections to database. At present, I have kept the pool size 10. When a new connection is needed it will simply borrow the connection from the pool and release as soon as the work gets completed.

- **Mocha Testing:- test.js File**

```
var express = require('express');
var request = require('request');
var assert = require('assert');
var http = require('http');
var mocha = require('mocha');

it('Test Login with Right Credentials', function(done) {
  request.post(
    'http://localhost:3000/checkLogin',
    { form: { username: 'shreykbhatt', password:'test' } },
    function (error, response, body) {
      assert.equal(200, response.statusCode);
      done();
    }
  );
});

it('Test the My advertisement page', function (done) {
  http.get('http://localhost:3000/myadvertisement', function (res) {
    assert.equal(200, res.statusCode);
    done();
  });
});

it('Test get home page', function (done) {
  http.get('http://localhost:3000/homepage', function (res) {
```

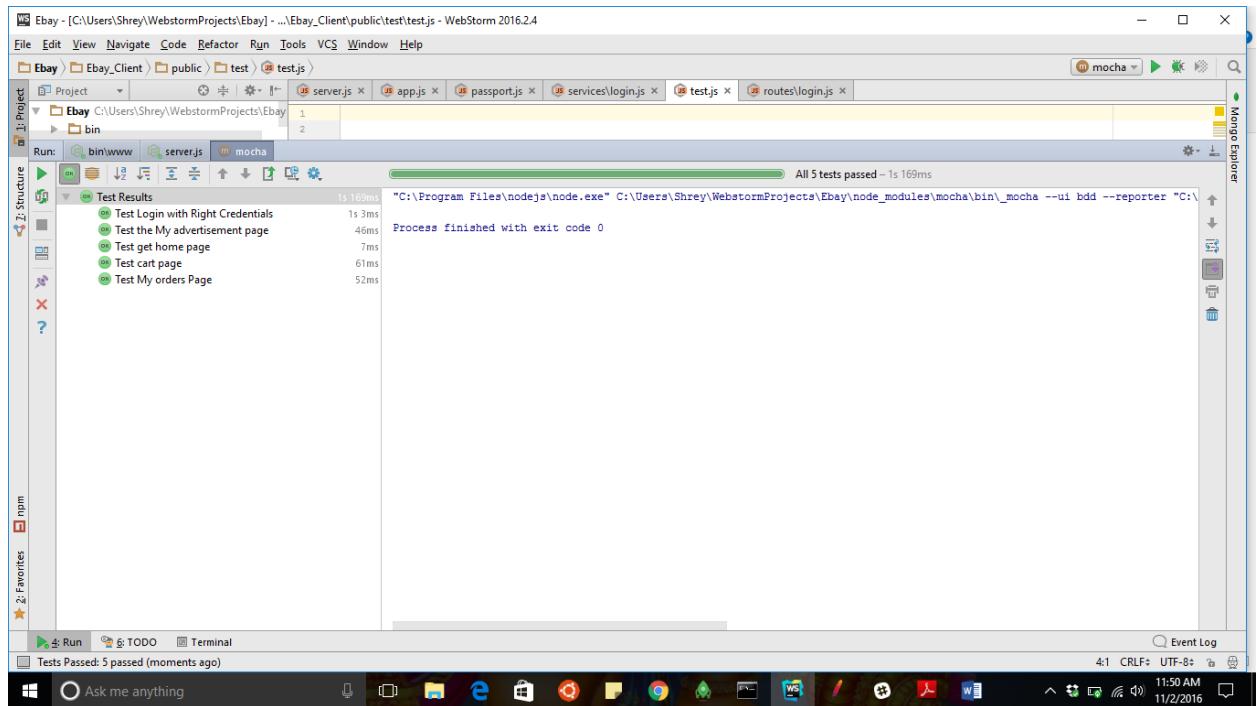
```

        assert.equal(200, res.statusCode);
        done();
    });
});

it('Test cart page', function (done) {
    http.get('http://localhost:3000/cart', function (res) {
        assert.equal(200, res.statusCode);
        done();
    });
});

it('Test My orders Page', function (done) {
    http.get('http://localhost:3000/myorders', function (res) {
        assert.equal(200, res.statusCode);
        done();
    });
});

```



New Tab Test Results — mocha ...

file:///E/SJSU/273/Lab2/Test%20Results%20-%20mocha.html

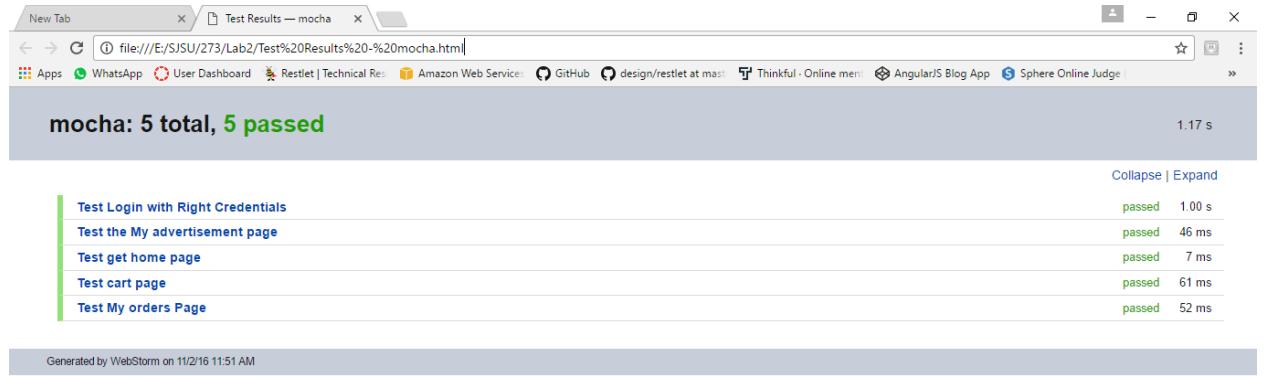
Apps WhatsApp User Dashboard Restlet | Technical Res Amazon Web Service GitHub design/restlet at mast Thinkful - Online ment AngularJS Blog App Sphere Online Judge

mocha: 5 total, 5 passed 1.17 s

Collapse | Expand

Test Login with Right Credentials	passed	1.00 s
Test the My advertisement page	passed	46 ms
Test get home page	passed	7 ms
Test cart page	passed	61 ms
Test My orders Page	passed	52 ms

Generated by WebStorm on 11/2/16 11:51 AM

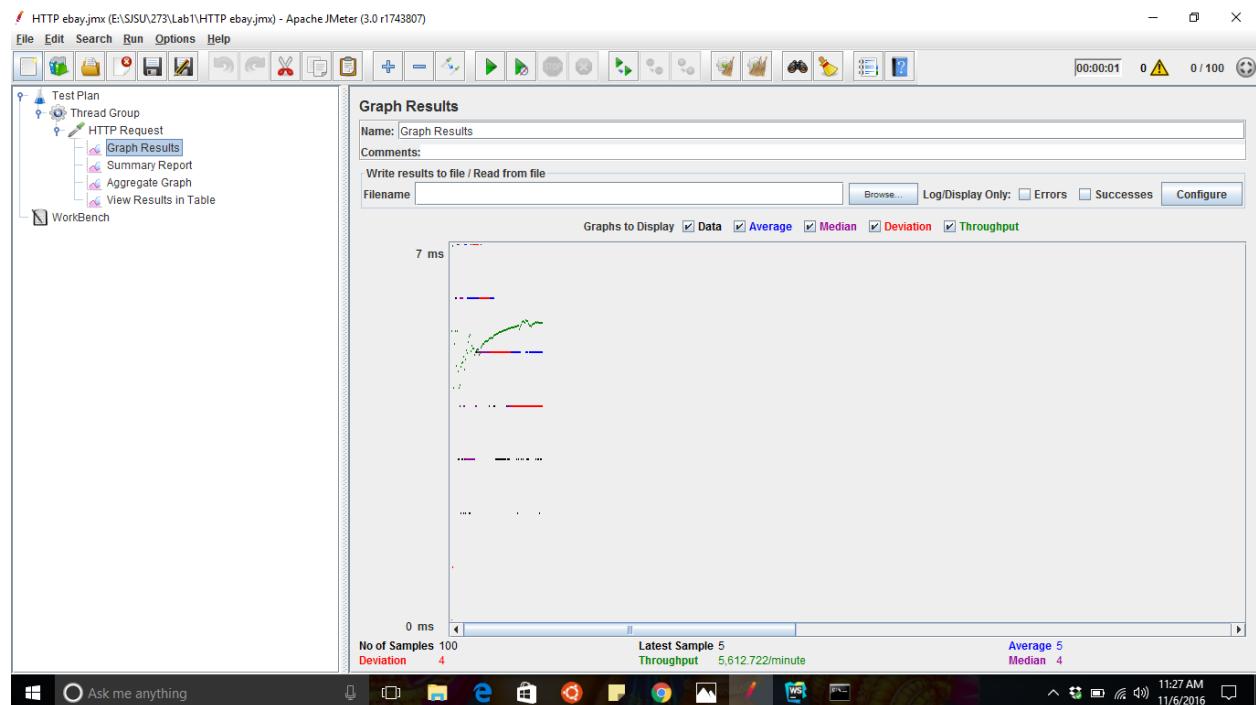


Part 2 : Answers

1.Explain what performance change RabbitMQ provides? Elaborate on the results of throughput with and without using RabbitMQ. If you find any increase/decrease in the throughput, explain the reason for the same.

1.Without RabbitMQ

1)100 concurrent users



HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Summary Report

Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	100	5	2	41	4.36	0.00%	93.5/sec	531.31	5816.0
TOTAL	100	5	2	41	4.36	0.00%	93.5/sec	531.31	5816.0

Include group name in label? Save Table Data Save Table Header

Ask me anything 11:27 AM 11/6/2016

2) 200 concurrent users

HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Graph Results

Name: Graph Results
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Graphs to Display Data Average Median Deviation Throughput

0 ms 14 ms

No of Samples 200 Deviation 8 Latest Sample 44 Throughput 10,398.614/min Average 6 Median 3

Ask me anything 7:19 PM 11/6/2016

HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan

Thread Group

HTTP Request

Graph Results

Summary Report

Aggregate Graph

View Results in Table

WorkBench

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

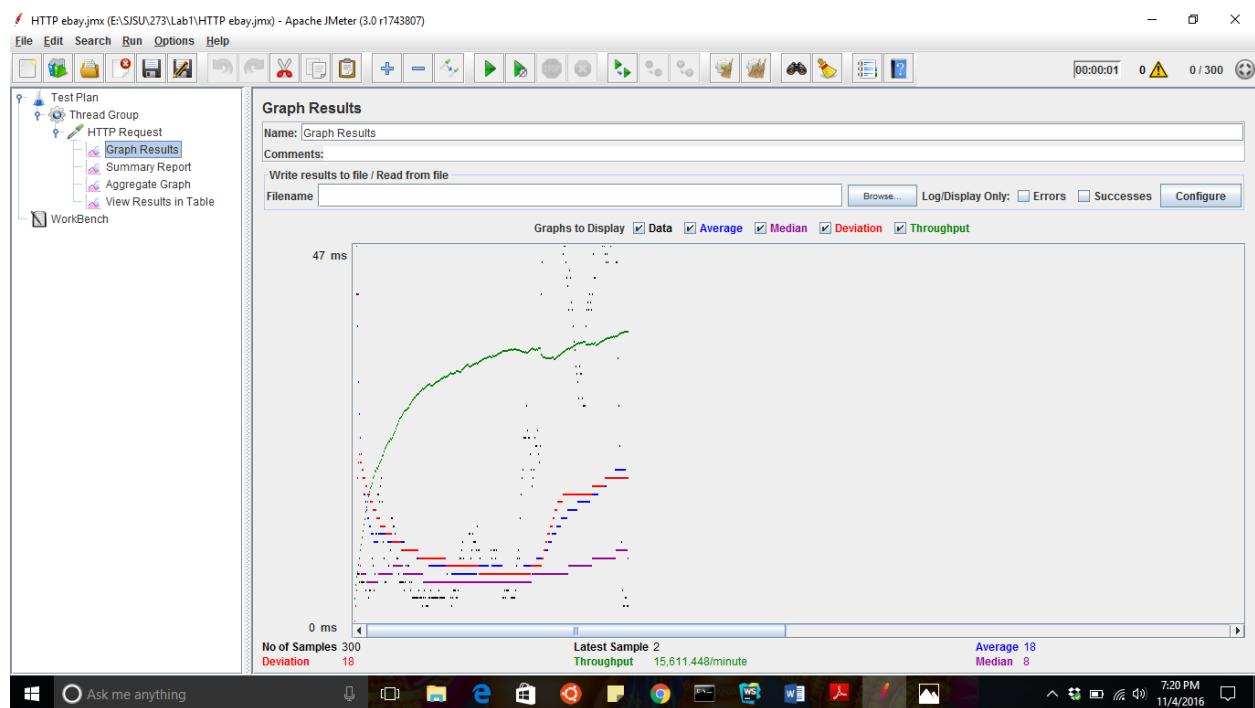
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	200	6	2	53	8.45	0.00%	173.3/sec	984.35	5816.0
TOTAL	200	6	2	53	8.45	0.00%	173.3/sec	984.35	5816.0

Include group name in label Save Table Data Save Table Header

Ask me anything 7:19 PM 11/4/2016

3) 300 concurrent users



HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

00:00:01 0 0 / 300

Test Plan

- Thread Group
 - HTTP Request
 - Graph Results
 - Summary Report
 - Aggregate Graph
 - View Results in Table

Summary Report

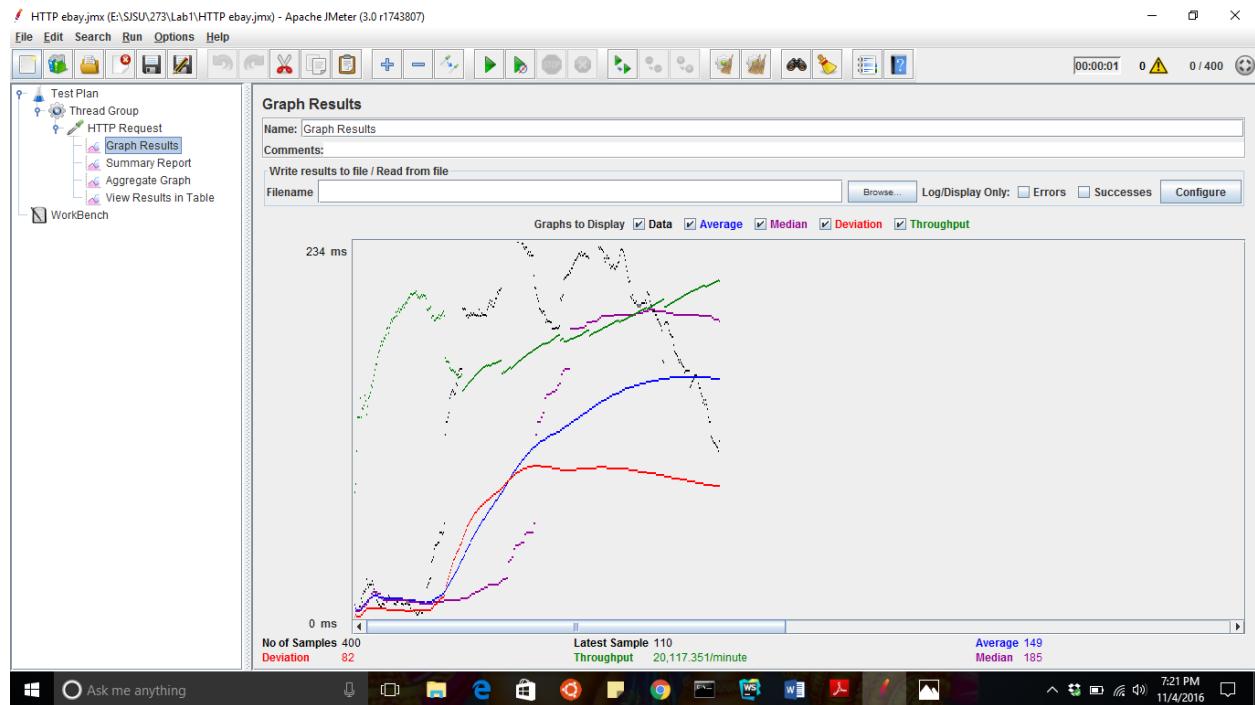
Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	300	18	2	64	18.52	0.00%	260.2/sec	1477.80	5816.0
TOTAL	300	18	2	64	18.52	0.00%	260.2/sec	1477.80	5816.0

Include group name in label Save Table Data Save Table Header

Ask me anything 7:20 PM 11/4/2016

4) 400 concurrent users



HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Summary Report

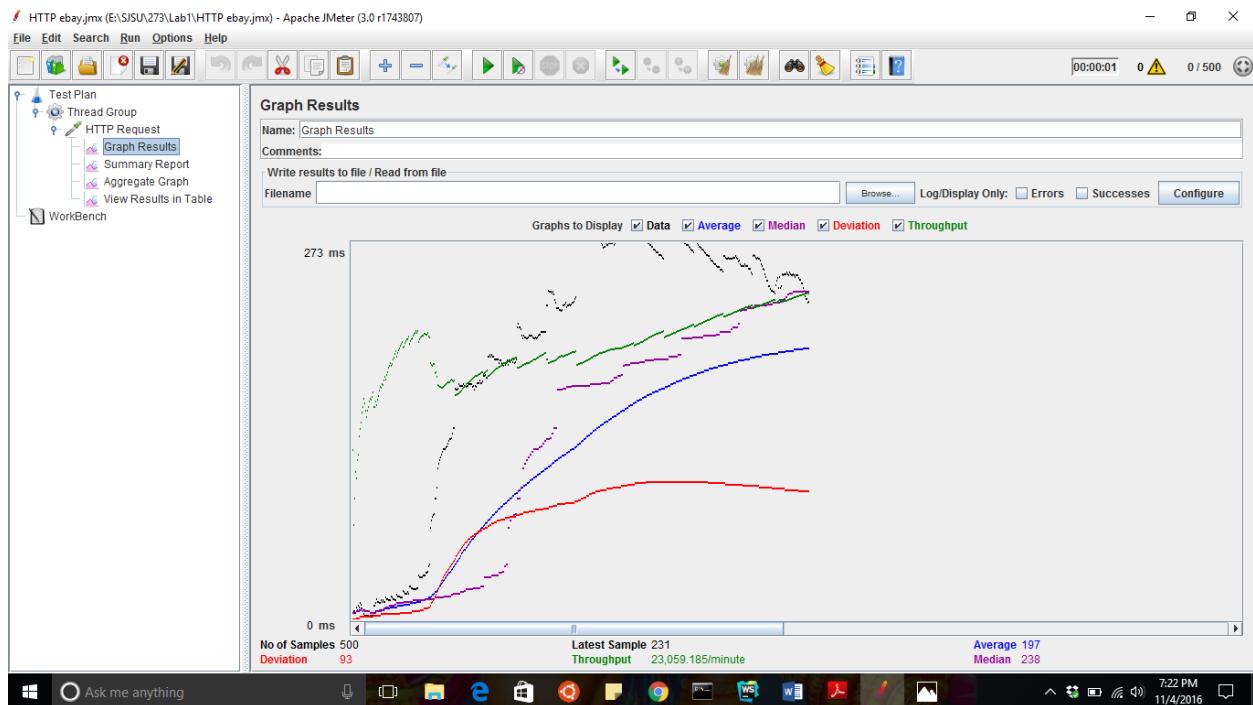
Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	400	149	2	251	82.43	0.00%	335.3/sec	1904.34	5816.0
TOTAL	400	149	2	251	82.43	0.00%	335.3/sec	1904.34	5816.0

Include group name in label? Save Table Data Save Table Header

7:20 PM 11/4/2016

5) 500 concurrent users



HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Summary Report

Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	500	197	2	294	93.16	0.00%	384.3/sec	2182.82	5816.0
TOTAL	500	197	2	294	93.16	0.00%	384.3/sec	2182.82	5816.0

Include group name in label? Save Table Data Save Table Header

Ask me anything 7:22 PM 11/4/2016

00:00:01 0 / 500

2. With RabbitMQ

1) 100 concurrent users

HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Graph Results

Name: Graph Results
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Graphs to Display Data Average Median Deviation Throughput

4 ms

0 ms

No of Samples 100 Deviation 3 Latest Sample 4 Throughput 5,976.096/minute

Average 3 Median 3

00:00:01 0 / 100

Ask me anything 2:39 AM 11/3/2016

HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Summary Report

Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	100	3	2	29	3.49	0.00%	99.6/sec	565.61	5815.0
TOTAL	100	3	2	29	3.49	0.00%	99.6/sec	565.61	5815.0

Include group name in label? Save Table Header

Ask me anything 2:40 AM 11/3/2016

2) 200 concurrent users

HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Graph Results

Name: Graph Results
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes

Graphs to Display Data Average Median Deviation Throughput

7 ms
0 ms

No of Samples 200 Deviation 4 Latest Sample 3 Throughput 11,776.251/minute Average 4 Median 3

Ask me anything 2:44 AM 11/3/2016

HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Summary Report

Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	200	4	2	29	4.36	0.00%	196.3/sec	1114.57	5815.0
TOTAL	200	4	2	29	4.36	0.00%	196.3/sec	1114.57	5815.0

Include group name in label? Save Table Data Save Table Header

Ask me anything 2:44 AM 11/3/2016

3) 300 concurrent users

HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group HTTP Request Graph Results Summary Report Aggregate Graph View Results in Table WorkBench

Graph Results

Name: Graph Results
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Graphs to Display Data Average Median Deviation Throughput

13 ms
0 ms
No of Samples 300
Latest Sample 2
Throughput 17,159.199/min
Average 6
Median 4

Ask me anything 2:40 AM 11/3/2016

HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan

Thread Group

HTTP Request

Graph Results

Summary Report

Aggregate Graph

View Results in Table

WorkBench

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	300	6	2	37	7.06	0.00%	286.0/sec	1624.04	5815.0
TOTAL	300	6	2	37	7.06	0.00%	286.0/sec	1624.04	5815.0

Include group name in label? Save Table Data Save Table Header

Ask me anything 2:41 AM 11/3/2016

4) 400 concurrent users

HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan

Thread Group

HTTP Request

Graph Results

Summary Report

Aggregate Graph

View Results in Table

WorkBench

Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: Errors Successes Configure

Graphs to Display: Data Average Median Deviation Throughput

28 ms

0 ms

No of Samples 400

Latest Sample 7

Throughput 22,492.971/min

Average 15

Median 15

Ask me anything 2:47 AM 11/3/2016

HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan

- Thread Group
 - HTTP Request
 - Graph Results
 - Summary Report
 - Aggregate Graph
 - View Results in Table
- WorkBench

Summary Report

Name: Summary Report
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	400	15	2	38	10.79	0.00%	374.9/sec	2128.85	5815.0
TOTAL	400	15	2	38	10.79	0.00%	374.9/sec	2128.85	5815.0

Include group name in label? Save Table Data Save Table Header

Ask me anything 2:47 AM 11/3/2016

5) 500 concurrent Users

HTTP ebay.jmx (E:\SISU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan

- Thread Group
 - HTTP Request
 - Graph Results
 - Summary Report
 - Aggregate Graph
 - View Results in Table
- WorkBench

Graph Results

Name: Graph Results
Comments:
Write results to file / Read from file
Filename Browse... Log/Display Only: Errors Successes Configure

Graphs to Display: Data Average Median Deviation Throughput

69 ms

0 ms

No of Samples 500 Deviation 19 Latest Sample 8 Throughput 27,322.404/minute Average 34 Median 34

Ask me anything 2:48 AM 11/3/2016

HTTP ebay.jmx (E:\SJSU\273\Lab1\HTTP ebay.jmx) - Apache JMeter (3.0 r1743807)

File Edit Search Run Options Help

Test Plan Thread Group
HTTP Request
Graph Results
Summary Report
Aggregate Graph
View Results in Table

WorkBench

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

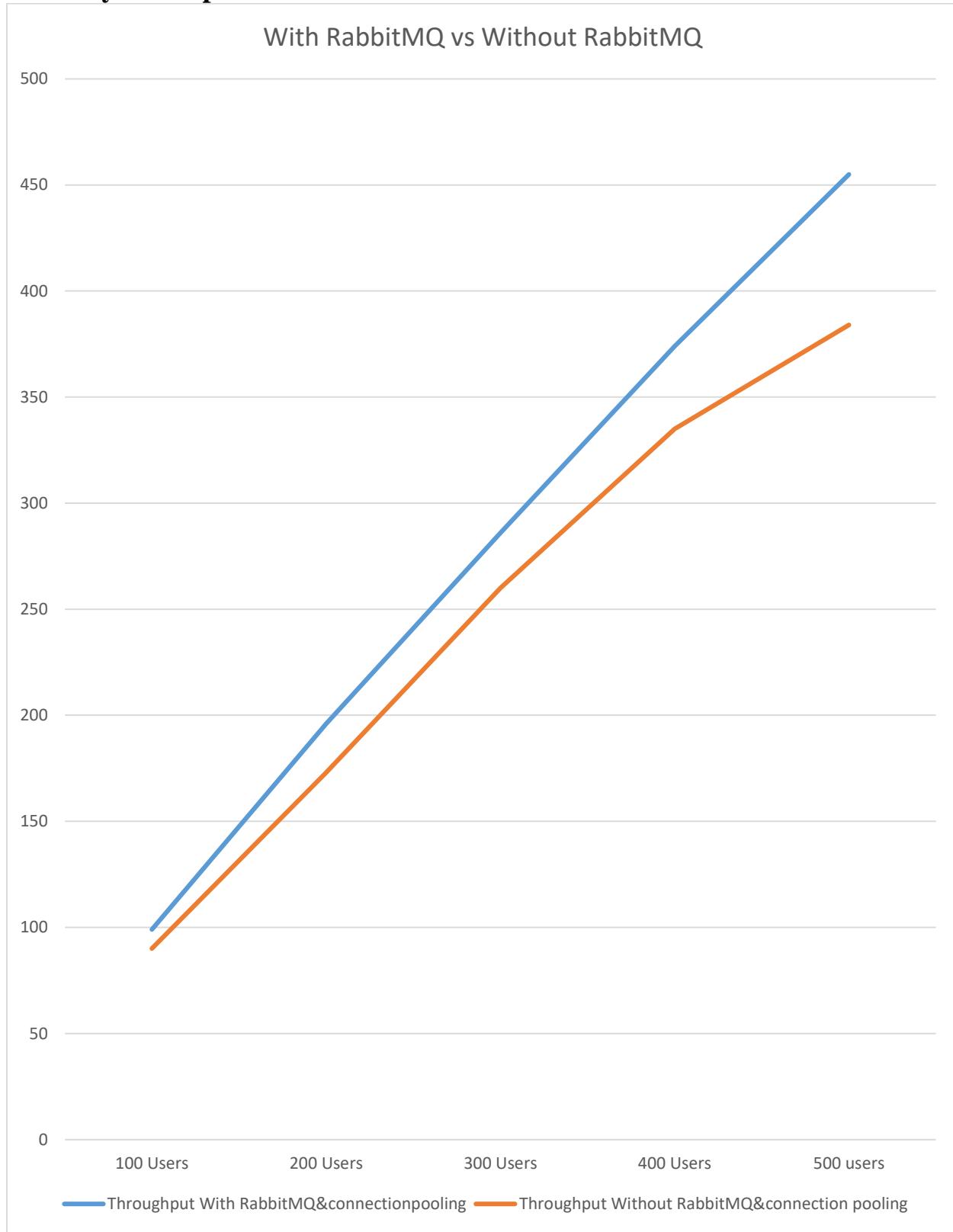
Filename Browse... Log/Display Only: Errors Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
HTTP Request	500	34	2	79	19.37	0.00%	455.4/sec	2585.93	5815.0
TOTAL	500	34	2	79	19.37	0.00%	455.4/sec	2585.93	5815.0

Include group name in label? Save Table Header

Ask me anything 2:49 AM 11/3/2016

- **Analysis Graph**



Explanation

- Y axis shows throughput per second
- X axis shows concurrent users

- As seen in the above graph if we use RabbitMQ Message Broker then the throughput to process the requests increases than the throughput without using RabbitMQ.
- And as the users increases Rabbit message queue gives much better performance.
- The RabbitMQ processes the requests in asynchronous manner.
- The requests are buffered heavily in message queue of the RabbitMQ. So the processing time for each request reduces and thus the throughput increases with RabbitMQ than what it was without RabbitMQ.

2. Compare passport authentication process with the authentication process used in Lab1.

- I used bcrypt for authentication process in Lab1.
- For using bcrypt we need to use hashing algorithm for hashing the passwords depending on the synchronized or asynchronous access.
- After deciding on a stack, we need to build out user management and authentication. this approach is prone to error, and requires more maintenance than passport authentication.
- Passport authentication helps us authenticating with different methods.
- We can use passport.authenticate() and specify which strategy to use. authenticate() function is used for standard connect middleware which can be used as route middleware in Express applications.

3. If given an option to implement MySQL and MongoDB both in your application, specify which data of the applications will you store in MongoDB and MySQL respectively

- MongoDB provides fast lookup compared to MySQL. Also MongoDB allows to create flexible Schemas.
- If my application has to deal with data which needs flexible schema management and the size of the data is comparatively large for the processing, then I will use MongoDB over MySQL.
- But one disadvantage of MongoDB is that there are no joins available like MySQL. So if my application needs to deal with the data which requires ACID properties or if my application has high transactional data then I will prefer MySQL over MongoDB.