



Simple Online Market Place



Team 09

Prototyping eBay to create a simple market place for users
to sell and buy their products on the web.

Arthi Ravishankar

Mansi Shah

Neeraj Gupta

Sai Mounika Nekkalapudi

Sanstuti Srivastav

Shreyasi Vad

Table of Contents

Abstract.....	5
System Architecture	6
Database	7
Database Scripts	7
<i>Auction</i>	7
<i>Bid</i>	7
<i>Buyer</i>	7
<i>Person</i>	8
<i>Product</i>	8
<i>Seller</i>	8
<i>Shopping Cart</i>	9
<i>Product View</i>	9
Database Schema.....	9
Redis Server	10
Connection Pooling.....	10
Auditing.....	11
Functionalities.....	12
User Sign Up:.....	12
User Login:	13
User Logout:.....	14
User Profile:	15
User Buying history:.....	16
User Selling history:	17
User Bidding history:.....	18
List all categories.....	19
List all sub categories:.....	20
List all Persons.....	21
List all Buyers	22
List all Sellers.....	23
List all Auctions	24
Search for a person based on attributes.....	25

<i>Search based on country “India”.....</i>	25
Search for a product based on attributes	27
Bidding for a product	28
Payment for a product and Shopping Cart	29
Selling a product	30
Update a product.....	31
Scalability, Performance and Reliability:	32
SQL Caching	32
Pagination: Redis Lists.....	32
Redis Hash-Map Searches.....	32
Denormalization	33
Views.....	34
Connection Pooling.....	34
Query Builder.....	34
Dependency Injection using Prepared Statement	35
Testing	36
JMeter: Performance Testing	36
Mocha: Functional Testing.....	40
Test scripts for integration Testing using Browser & Jmeter.....	40
Testing Snapshots using Mocha.....	41
Code Listing	42
App.js	42
Admin.....	43
Auction.....	50
Authentication	52
Bid	54
Buy Product.....	54
Categories	55
Payment.....	56
Person	57
Product	60
Sell.....	65

DB Connection Pooling	66
Pagination and Caching.....	67
Data insertion Scripts.....	70

Prototyping eBay: Online Market Place

(*Enterprise Distributed Systems, Fall 2014*)

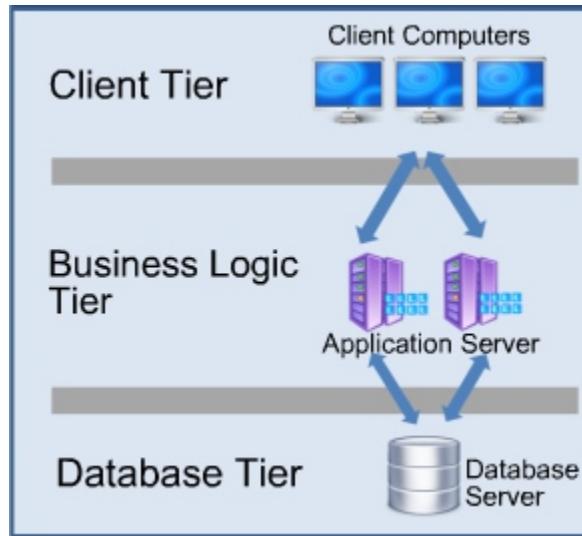
Abstract

Advent of internet has revolutionized many domains, one of which is commerce. E-commerce has seen exponential growth in recent years and has been one of the most impacting outcomes of internet growth. It allows customers to exchange services online, like selling a product or buying a product, irrespective of time and distance. It supports both B2B and B2C commerce.

Keeping in mind the idea behind a simple online market place, the aim was to replicate the basic functionalities of eBay where a user can register itself and can search for the products catalog to buy it. At the same time, a registered user has the capability of selling a product. This sale can be made at a fixed price or can be put up for an auction where other registered users can bid and buy. Thus, a user can be both buyer and seller. The buyer can buy a fixed sale product or bid for an actionable product. The admin is capable of deleting a person. Therefore, a simple online market place serves as a focal point for all buyers to buy products and sellers to sell their products, thereby promoting ecommerce.

System Architecture

Our architecture consists of three tiers, shown below:



Technologies Implemented		
Client Tier	Business Logic Tier	Database Tier
<ul style="list-style-type: none">HTML5jQueryAjax	<ul style="list-style-type: none">Node.js Express Framework	<ul style="list-style-type: none">MySQLRedis Server

Database

Database Scripts

Auction

```
CREATE TABLE `auction` (
  `auction_id` bigint(20) NOT NULL AUTO_INCREMENT,
  `seller_id` bigint(20) DEFAULT NULL,
  `buyer_id` bigint(20) DEFAULT NULL,
  `product_id` bigint(20) DEFAULT NULL,
  `auction_starting_date` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `auction_end_date` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  `auction_starting_price` float(100,2) DEFAULT NULL,
  `current_price` float(100,2) DEFAULT NULL,
  PRIMARY KEY (`auction_id`),
  KEY `product_id_idx` (`product_id`),
  CONSTRAINT `product_id` FOREIGN KEY (`product_id`) REFERENCES `product` (`product_id`) ON DELETE NO ACTION ON
  UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=19581 DEFAULT CHARSET=latin1;
```

Bid

```
CREATE TABLE `bid` (
  `bidid` mediumint(9) NOT NULL AUTO_INCREMENT,
  `auction_id` bigint(20) DEFAULT NULL,
  `bidder_id` bigint(20) DEFAULT NULL,
  `bid_price` float(100,13) DEFAULT NULL,
  `product_id` bigint(20) DEFAULT NULL,
  `bid_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `bidderfirstname` varchar(45) DEFAULT NULL,
  `bidderlastname` varchar(45) DEFAULT NULL,
  `product_title` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`bidid`),
  UNIQUE KEY `bidid_UNIQUE` (`bidid`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1;
```

Buyer

```
CREATE TABLE `buyer` (
  `buyerid` bigint(20) NOT NULL,
  `product_id` bigint(20) DEFAULT NULL,
  `buying_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `buyerfirstname` varchar(256) DEFAULT NULL,
  `buyerlastname` varchar(256) DEFAULT NULL,
  `product_title` varchar(128) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Person

```
CREATE TABLE `person` (
  `memberid` bigint(20) NOT NULL AUTO_INCREMENT,
  `firstname` varchar(256) DEFAULT NULL,
  `lastname` varchar(256) DEFAULT NULL,
  `address` varchar(256) DEFAULT NULL,
  `city` varchar(128) DEFAULT NULL,
  `state` varchar(128) DEFAULT NULL,
  `country` varchar(128) NOT NULL,
  `zipcode` int(11) NOT NULL,
  `emailaddr` varchar(128) NOT NULL,
  `password` varchar(256) NOT NULL,
  `payment_type` varchar(128) DEFAULT NULL,
  `payment_accno` bigint(20) DEFAULT NULL,
  `admin` varchar(5) NOT NULL DEFAULT 'no',
  PRIMARY KEY (`memberid`)
) ENGINE=InnoDB AUTO_INCREMENT=20003 DEFAULT CHARSET=latin1;
```

Product

```
CREATE TABLE `product` (
  `product_id` bigint(20) NOT NULL,
  `product_title` varchar(128) DEFAULT NULL,
  `product_condition` varchar(100) DEFAULT NULL,
  `category` varchar(128) DEFAULT NULL,
  `subcategory` varchar(128) DEFAULT NULL,
  `available_quantity` int(11) DEFAULT NULL,
  `product_description` text,
  `instant_sellingprice` float(100,2) DEFAULT NULL,
  `listing_type` varchar(128) DEFAULT NULL,
  `initial_quantity` int(11) DEFAULT NULL,
  PRIMARY KEY (`product_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Seller

```
CREATE TABLE `seller` (
  `sellerid` bigint(20) DEFAULT NULL,
  `product_id` bigint(20) DEFAULT NULL,
  `selling_time` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  `product_title` varchar(128) DEFAULT NULL,
  `sellerfirstname` varchar(256) DEFAULT NULL,
  `sellerlastname` varchar(256) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

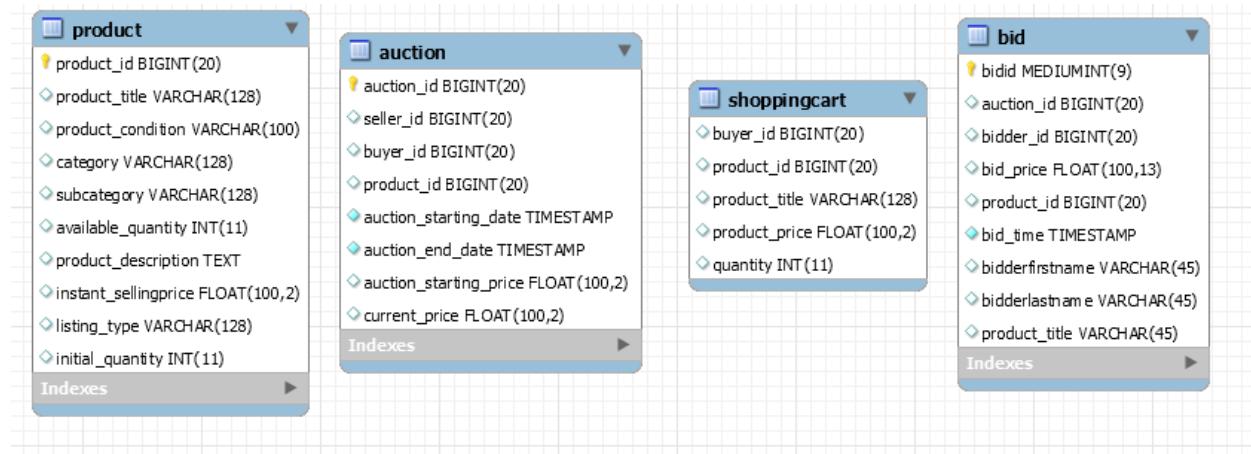
Shopping Cart

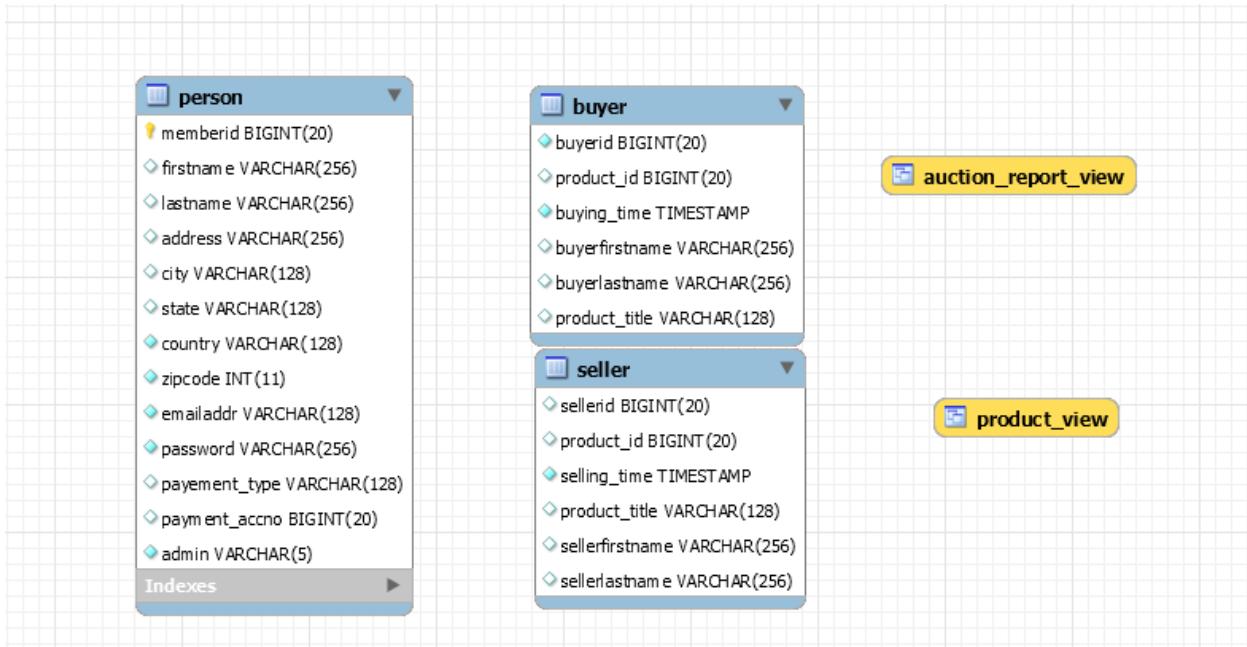
```
CREATE TABLE `shoppingcart` (
  `buyer_id` bigint(20) DEFAULT NULL,
  `product_id` bigint(20) DEFAULT NULL,
  `product_title` varchar(128) DEFAULT NULL,
  `product_price` float(100,2) DEFAULT NULL,
  `quantity` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Product View

```
view
CREATE ALGORITHM=UNDEFINED DEFINER='root'@'localhost' SQL SECURITY DEFINER VIEW `product_view` AS select `p`.`product_id` AS `product_id`, `p`.`product_title` AS `product_title`, `p`.`product_condition` AS `product_condition`, `p`.`category` AS `category`, `p`.`subcategory` AS `subcategory`, `p`.`available_quantity` AS `available_quantity`, `p`.`product_description` AS `product_description`, `p`.`instant_sellingprice` AS `instant_sellingprice`, `p`.`listing_type` AS `listing_type`, `p`.`initial_quantity` AS `initial_quantity`, `a`.`auction_id` AS `auction_id` from (`product` `p` left join `auction` `a` on(`p`.`product_id` = `a`.`product_id`));
```

Database Schema





Redis Server

To improve the efficiency and performance of our system, we have used Redis Server. Our aim was to use data structures provided by the redis server to hold data values for listing and search functionalities. We know that the key-value pair in Redis is not limited to just string values and are capable of holding more complex data structures like JPEG file or empty string.

In our system, we have implemented the following features:

Redis Lists: We have used the ‘Lists’ data structure for all listing operations. The idea is to store all values in the Redis lists, such that when a user requests the list of say categories, we fetch the data from the Redis Lists instead of hitting database with ‘Select’ statements, which is one of the most expensive operations. Also, this avoids using ‘select * from’ statements, which degrades the performance of the system.

Hashes: This is used for search operations in the system; each key is a query which has an associated value as a json string of result. Every time an update is made, the key-value pair is also updated. Thus, in this way the database select hits are reduced , thereby improving the performance of the system.

Connection Pooling

Nodejs provide connection pooling in multiple ways, either you can use library like generic pool or create it directly without using any library. We have implemented non generic pool, wherein we are creating a pool of 10 database connections.

Auditing

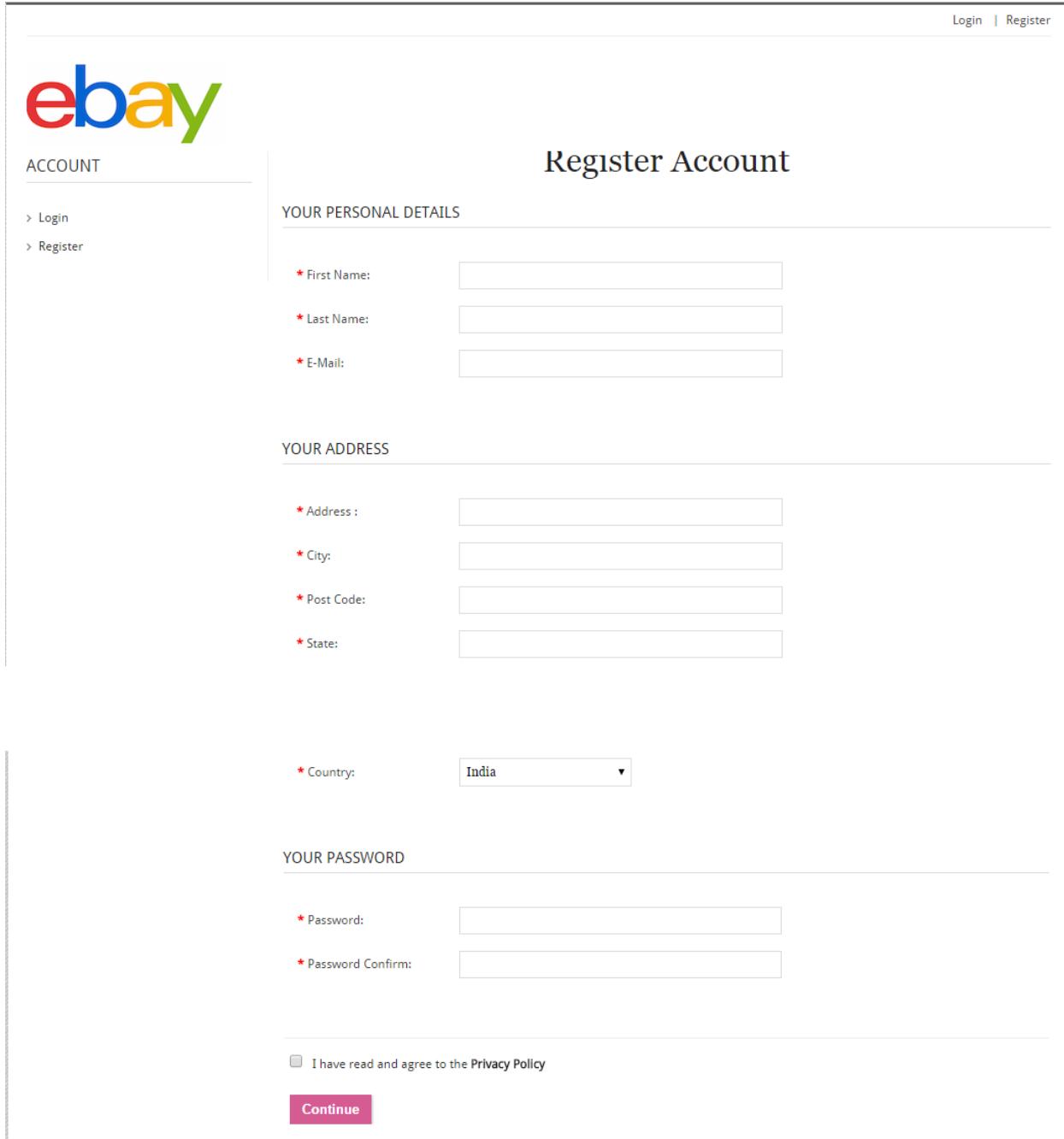
We have enabled auditing for our database for monitoring and recording purposes. We have executed a query so that all the actions performed in the system is stored in a file inside this directory \MYSQL\MySQL Server 5.6\data under the laptop name

```
SET GLOBAL general_log = 'ON';
```

```
SET GLOBAL general_log = 'OFF';
```

Functionalities

User Sign Up:



The image shows the eBay 'Register Account' form. At the top right, there are 'Login' and 'Register' links. The main title 'Register Account' is centered above the personal details section. On the left sidebar, under 'ACCOUNT', there are links for 'Login' and 'Register'. The 'YOUR PERSONAL DETAILS' section contains fields for First Name, Last Name, and E-Mail, each marked with a red asterisk. The 'YOUR ADDRESS' section contains fields for Address, City, Post Code, and State. A dropdown menu for Country shows 'India' selected. The 'YOUR PASSWORD' section contains fields for Password and Password Confirm. At the bottom, there is a checkbox for accepting the Privacy Policy and a 'Continue' button.

ACCOUNT

Login | Register

Register Account

YOUR PERSONAL DETAILS

* First Name:

* Last Name:

* E-Mail:

YOUR ADDRESS

* Address:

* City:

* Post Code:

* State:

* Country:

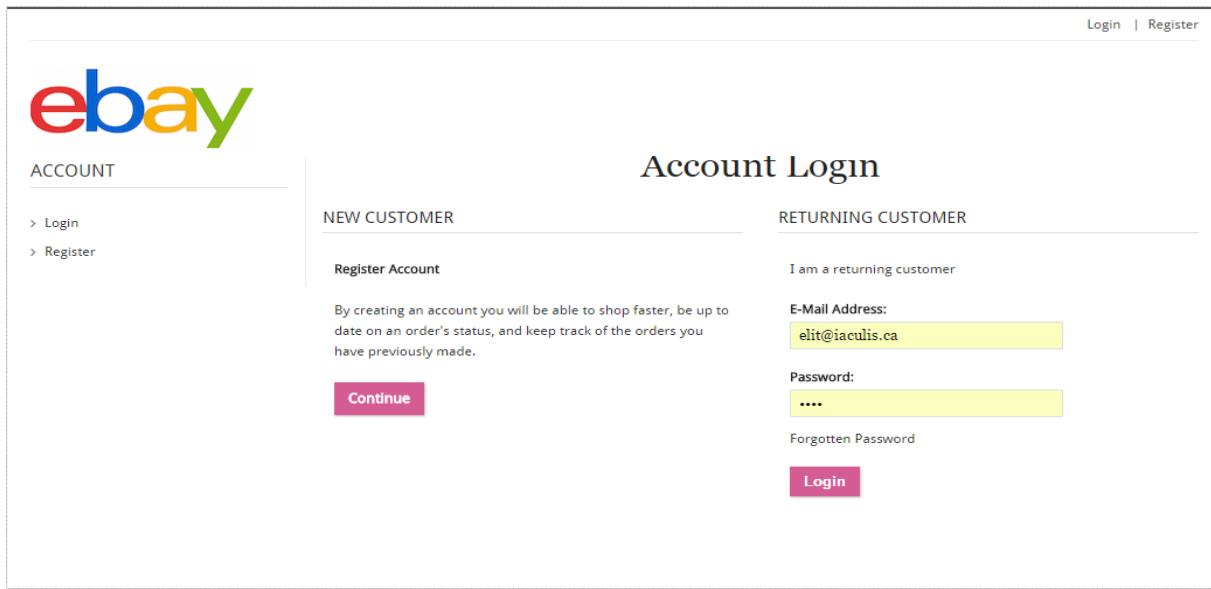
YOUR PASSWORD

* Password:

* Password Confirm:

I have read and agree to the [Privacy Policy](#)

Continue

User Login:

The screenshot shows the eBay Account Login page. At the top right, there are links for "Login" and "Register". The eBay logo is prominently displayed. On the left, there's a sidebar with "ACCOUNT" and links for "Login" and "Register". The main area has two sections: "NEW CUSTOMER" on the left with a "Register Account" link and a note about benefits; and "RETURNING CUSTOMER" on the right with fields for "E-Mail Address" (containing "elite@iaculis.ca") and "Password" (containing "...."), along with links for "Forgotten Password" and a large "Login" button.

ACCOUNT

> Login
> Register

ebay

Account Login

NEW CUSTOMER

Register Account

By creating an account you will be able to shop faster, be up to date on an order's status, and keep track of the orders you have previously made.

Continue

RETURNING CUSTOMER

I am a returning customer

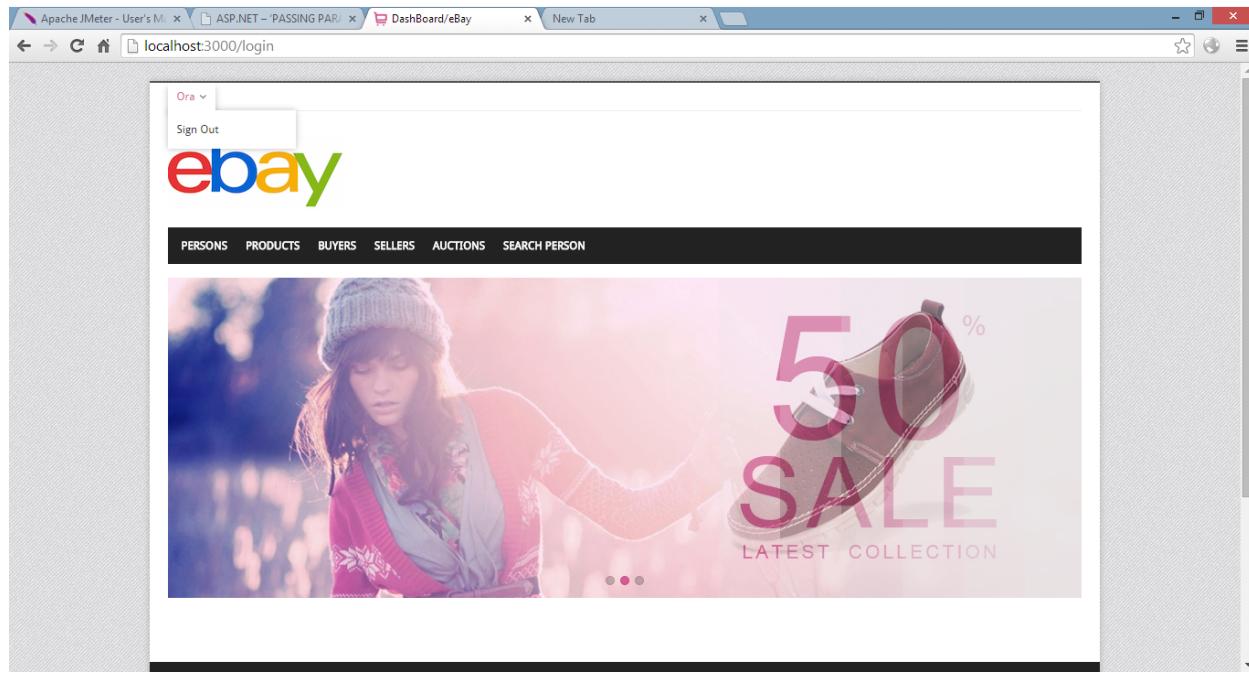
E-Mail Address:
elite@iaculis.ca

Password:
....

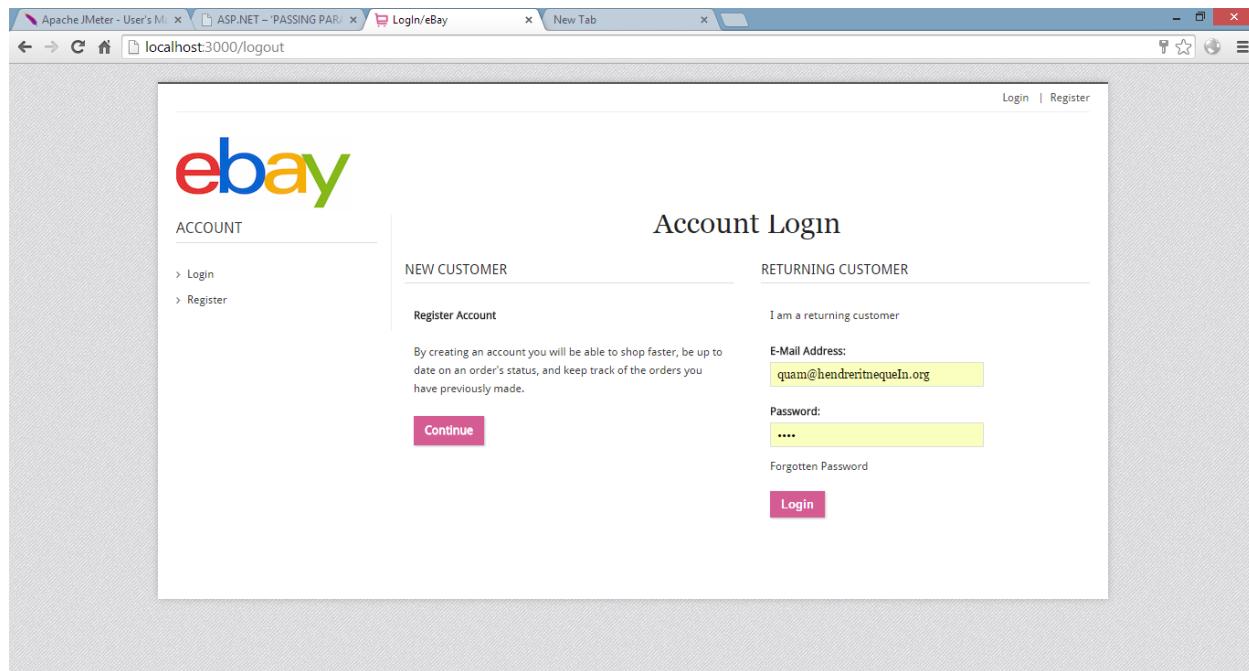
Forgotten Password

Login

User Logout:



After Logout: user redirected to Login:



User Profile:

The screenshot shows the eBay user profile page for 'HAVIVA WARREN'. At the top, there's a search bar and a cart icon. Below the header, there are navigation links for 'CATEGORIES', 'ELECTRONICS', 'ACCESORIES', 'APPAREL', 'FURNITURE', 'BOOKS', 'WATCHES', and 'KITCHEN'. On the left, there's a placeholder profile picture and contact information: Name: Haviva Warren, Address: P.O. Box 490, 8272 Sed, Av., City: Helena, State: Montana, Country: United States, Zip code: 55482. The main content area shows tabs for 'Update Account', 'Bidding History', 'Buying History', and 'Selling History'. At the bottom, there are links for 'INFORMATION', 'CUSTOMER SERVICE', 'EXTRAS', and 'MY ACCOUNT', along with social media icons for Facebook, Twitter, Google+, Pinterest, RSS, and LinkedIn.

Haviva | Sell

My eBay | Checkout

ebay

Search

Cart

CATEGORIES ELECTRONICS ACCESORIES APPAREL FURNITURE BOOKS WATCHES KITCHEN

HAVIVA WARREN

Name: Haviva Warren
Address: P.O. Box 490, 8272 Sed, Av.
City: Helena
State: Montana
Country: United States
Zip code: 55482

Update Account Bidding History Buying History Selling History

INFORMATION

About Us
Delivery Information
Privacy Policy
Elements

CUSTOMER SERVICE

Contact Us
Returns
Site Map

EXTRAS

Brands
Gift Vouchers
Affiliates
Specials

MY ACCOUNT

My Account
Order History
Wish List
Newsletter

eBay © 2014

PayPal American Express 2CO

f t g+ p

User Buying history:

The screenshot shows a user profile on eBay. At the top, there's a navigation bar with 'Nomlanga' and 'Sell' on the left, and 'My eBay | Checkout' on the right. Below the navigation is the eBay logo and a search bar. A shopping cart icon is also present. A black navigation bar below the logo contains links for 'CATEGORIES', 'ELECTRONICS', 'ACCESORIES', 'APPAREL', 'FURNITURE', 'BOOKS', 'WATCHES', and 'KITCHEN'. The main content area displays a user profile for 'NOMLANGA HINTON' with a blue silhouette profile picture. Below the profile picture, the user's name and address are listed: Name: Nomlanga Hinton, Address: 143-6117 Purus Ave, City: Tampa, State: Florida, Country: United States, Zip code: 59700. To the right of the profile information, there's a section titled 'Buying History' with a link to 'Buying History'. Below this, there's a product listing for a 'Kids belt' in the 'dresses' category, priced at \$15. At the bottom of the page, there are four columns: 'INFORMATION' (About Us, Delivery Information, Privacy Policy, Elements), 'CUSTOMER SERVICE' (Contact Us, Returns, Site Map), 'EXTRAS' (Brands, Gift Vouchers, Affiliates, Specials), and 'MY ACCOUNT' (My Account, Order History, Wish List, Newsletter). The footer features payment method icons for PayPal, American Express, 2CO, MasterCard, and Discover, along with the text 'eBay © 2014'. On the far right of the footer, there are social media sharing icons for Facebook, Twitter, Google+, Pinterest, RSS, and LinkedIn.

User Selling history:

Nomlanga Sell My eBay | Checkout

ebay

CATEGORIES ELECTRONICS ACCESORIES APPAREL FURNITURE BOOKS WATCHES KITCHEN



NOMLANGA HINTON

Update Account Bidding History Buying History Selling History

		
ProductTitle: NIB Nike Watch Torque SI WC0067-096 Black /Silver Category: watch \$50	ProductTitle: "Listen to These" Thomas Masson Hardcover Book Antique 1923 Category: books \$50	ProductTitle: Dufonte Lucien Picard Men's Wristwatch w/ Date & Perri Ellis Band ~ 13-B4233 Category: watch \$50

INFORMATION	CUSTOMER SERVICE	EXTRAS	MY ACCOUNT
About Us	Contact Us	Brands	My Account
Delivery Information	Returns	Gift Vouchers	Order History
Privacy Policy	Site Map	Affiliates	Wish List
Elements		Specials	Newsletter

PayPal American Express 2CO MasterCard Discover

eBay © 2014



User Bidding history:

The screenshot shows a user profile page on eBay. At the top, there's a navigation bar with links for 'Nomlanga' and 'Sell', and buttons for 'My eBay' and 'Checkout'. The eBay logo is prominently displayed. A search bar is located above the main content area. To the right, there's a shopping cart icon labeled 'Cart'. Below the header, a black navigation bar contains links for 'CATEGORIES', 'ELECTRONICS', 'ACCESORIES', 'APPAREL', 'FURNITURE', 'BOOKS', 'WATCHES', and 'KITCHEN'. The main content area features a large blue placeholder image for a profile picture, followed by the text 'NOMLANGA HINTON'. Below this, there are four hyperlinks: 'Update Account', 'Bidding History', 'Buying History', and 'Selling History'. A message 'No Results found!' is displayed. On the left side of the main content area, there's a sidebar with the user's contact information: Name: Nomlanga Hinton, Address: 143-6117 Purus Ave, City: Tampa, State: Florida, Country: United States, Zip code: 59700. At the bottom of the page, there's a footer with sections for 'INFORMATION', 'CUSTOMER SERVICE', 'EXTRAS', and 'MY ACCOUNT', each listing various links. The footer also includes payment method icons for PayPal, American Express, 2CO, MasterCard, and Discover, along with the text 'eBay © 2014'. On the far right, there are social media sharing icons for Facebook, Twitter, Google+, Pinterest, and LinkedIn.

List all categories

The screenshot shows the eBay homepage with a navigation bar at the top. The main content area displays a grid of category links:

CATEGORIES	ELECTRONICS	ACCESORIES	APPAREL	FURNITURE	BOOKS	WATCHES	KITCHEN
Electronics	Accessories		Apparel		Furniture		Books
HeadPhones	Children's Jewellery		Tops & Blouses		Sofas, Loveseats & Chaises		Cookbooks
PCBs	Men's Jewellery		Kleider		Chairs & Seating		Textbooks, Education
Security Cameras	Women's Jewellery		Jackets & Leathers		Furniture Parts & Accessories		Cartoon Character
Lights & Reflectors	Sample Test		Sweaters		Dollhouses		Atlases
Cigarettes	Sample Test11		Skirts		Stylist Stations & Furniture		American Vehicles
Microscopes	Sample Test12		Medieval & Renaissance		Door Knockers		Other Children & Young Adults
Watches	Kitchen						
Wristwatches	Kitchen & Steak Knives						
Wristwatch Bands	Loose Cutlery						
Montres classiques	Egg Cups						
Other	Besteck-Einzelteile						
	Slicers						
	Cooking Utensils						

List all sub categories:

The screenshot shows a web browser window for 'Products/eBay' at 'localhost:3000/searchProduct?subcategory=HeadPhones'. The page title is 'HeadPhones'. The navigation bar includes links for Categories, Electronics, Accessories, Apparel, Furniture, Books, Watches, and Kitchen. A search bar and a cart icon are also present. On the left, a sidebar displays a hierarchical menu under 'CATEGORIES'. Under 'ELECTRONICS', the 'HeadPhones' category is selected, showing sub-categories: PCBs, Security Cameras, Lights & Reflectors, Cigarettes, and Microscopes. Other categories listed include Jewellery, Apparel, Furniture, Books, Watches, and Kitchen. The main content area shows four product listings for headphones:

Product	Description	Price	Action
Bose SoundTrue Around Ear Headphones - Black	FREE SHIPPING	\$50	ADD TO CART
Beats by Dr. Dre Mixr Headband Headphones - Black limited Edition		\$50	ADD TO CART
Red Adjustable Over-Ear Earphone Headset 3.5mm For iPod iPhone MP3 MP4 5G 4Th 5S		\$50	ADD TO CART
(2) iFrogz Earpollution Toxix Stereo Headphones MP3 iPod iPhone Computer EPTXRED		\$50	ADD TO CART

List all Persons

Admin ▾

ebay

Search

PERSONS PRODUCTS BUYERS SELLERS AUCTIONS SEARCH PERSON

PERSONS

Lee Washington Address: P.O. Box 761, 3458 Nib St. Country: United States DELETE	Sawyer Barnes Address: Ap #511-8687 Auctor Road Country: India DELETE	Roth Pugh Address: P.O. Box 460, 4224 Ac Road Country: India DELETE
Nomlanga Hinton Address: 143-6117 Purus Ave Country: United States DELETE	Allistair Wells Address: 397-2286 Orci St. Country: United States DELETE	Rebekah Hull Address: Ap #236-301 Orci St. Country: Australia DELETE

List all Buyers

Admin ▾

ebay

PERSONS PRODUCTS BUYERS SELLERS AUCTIONS SEARCH PERSON

BUYERS



Sawyer Barnes
ipad
2014-11-30T00:33:00.000Z

Nomlanga Hinton
Kids belt
2014-11-30T04:50:09.000Z

DELETE **DELETE**



Nomlanga Hinton
TWO TONE BROWN MICROFIBER DOUBLE (TWO) RECLINER RECLINING SOFA SECTIONAL
2014-12-07T21:33:38.000Z

DELETE

List all Sellers

Admin ▾

ebay

PERSONS PRODUCTS BUYERS SELLERS AUCTIONS SEARCH PERSON

SELLER



Ram Gamble
TWO TONE BROWN MICROFIBER DOUBLE (TWO) RECLINER RECLINING SOFA SECTIONAL
Time: 2014-12-07T01:49:59.000Z

DELETE



Lee Washington
6 Pcs Bridal Wedding Rose Flower Hair Pin Party Prom Hair Styling Accessory
Time: 2014-12-07T01:49:59.000Z

DELETE



Rebekah Hull
Illustrator SIGNED 1949 Religious Poetry Book ~ HE THE MAKER by William B. Ziff
Time: 2014-12-07T01:49:59.000Z



Sawyer Barnes
Theodore Alexander Side Table
Time: 2014-12-07T01:49:59.000Z

Search for a person based on attributes

This feature is specific to admin, where in the admin has special privileges of deleting a person, seller or buyer, from the system in case of any malicious behavior from the person's end like fraudulent sale etc.

Search based on country "India"

The screenshot shows a search form titled "SEARCH PERSON" with fields for First Name, Last Name, City, State, Country (set to India), and ZipCode. Below the form is a "Search" button. At the bottom, there are links for INFORMATION (About Us, Delivery Information, Privacy Policy, Elements), CUSTOMER SERVICE (Contact Us, Returns, Site Map), EXTRAS (Brands, Gift Vouchers, Affiliates, Specials), and MY ACCOUNT (My Account, Order History, Wish List, Newsletter). Logos for PayPal, MasterCard, 2CO, American Express, and Discover are displayed, along with social media icons for Facebook, Twitter, Google+, Pinterest, RSS, and LinkedIn.

Admin

ebay

PERSONS PRODUCTS BUYERS SELLERS AUCTIONS SEARCH PERSON

SEARCH PERSON

First Name :

Last Name :

City:

State:

Country:

ZipCode:

Search

INFORMATION

About Us
Delivery Information
Privacy Policy
Elements

CUSTOMER SERVICE

Contact Us
Returns
Site Map

EXTRAS

Brands
Gift Vouchers
Affiliates
Specials

MY ACCOUNT

My Account
Order History
Wish List
Newsletter

eBay @ 2014

PayPal American Express 2CO American Express Discover

f t g+ p r s v

Admin ▾

ebay

PERSONS PRODUCTS BUYERS SELLERS AUCTIONS SEARCH PERSON

SEARCH PERSON RESULTS



Sawyer Barnes
Ap #511-8687 Auctor Road
India



Roth Pugh
P.O. Box 460, 4224 Ac Road
India



Rae Albert
689-3505 Ut St.
India



Eliana Orr
4872 Diam Road
India



Grant Logan
6757 Ut Street
India



Renee Walker
306-9254 Non Av.
India

Search for a product based on attributes



Bidding for a product

ebay

Search

Cart

CATEGORIES ELECTRONICS ACCESORIES APPAREL FURNITURE BOOKS WATCHES KITCHEN

FREE Delivery



WIDE SYNTHETIC LEATHER HEADBAND HAIR BAND ACCESSORY HAT SIMPLE PLAIN HB1386

Product Title: WIDE SYNTHETIC LEATHER HEADBAND HAIR BAND ACCESSORY HAT SIMPLE PLAIN HB1386

Product Code: 158857

Product Condition: New without tags

Product Description: Check out this cool product. Buy it now or will be sold out soon

Available Quantity: 63

Price: 50 **Place Bid**

Payment for a product and Shopping Cart

Haviva Sell My eBay | Checkout

ebay

ACCOUNT

> Login
> Register
> Forgotten Password

YOUR PAYMENT DETAILS

*Account Number: * Payment Type:

YOUR PASSWORD

* Password:
* Password Confirm:

I have read and agree to the Privacy Policy

Confirm Details

Haviva Sell My eBay | Checkout

ebay

Search

Home ◄ Shopping Cart

SHOPPING CART

IMAGE	PRODUCT NAME	QUANTITY	DEL	UNIT PRICE	TOTAL
	NIB Nike Watch Torque SI WC0067-096 Black /Silver	1	█	\$50	\$50
	TWO TONE BROWN MICROFIBER DOUBLE (TWO) RECLINING SOFA SECTIONAL	1	█	\$50	\$50
Total: \$100					
Checkout					

Selling a product

Homepage | Logout

The image shows a screenshot of the eBay 'Selling a product' interface. At the top, the eBay logo is visible next to the text 'What are you selling'. Below this, a section titled 'DESCRIBE YOUR ITEM' contains fields for Product Title, Condition (set to 'New'), Category (set to 'Accessories'), Sub Category, and a large text area for Description with placeholder text: 'Enter a detailed description of the product you are selling.' A note below the condition field explains the definition of 'New other'. The next section, 'SELECT FORMAT AND PRICE', includes fields for List as (set to 'Auction'), Quantity, Price, Start Date (set to 'mm/dd/yyyy'), and End Date (set to 'mm/dd/yyyy'). At the bottom, there is a checkbox for agreeing to the eBay terms and conditions, followed by a 'Continue' button.

What are you selling

DESCRIBE YOUR ITEM

* Product Title:

* Condition:

* Category:

* Sub Category:

* Description:
Enter a detailed description of the product you are selling.

** New other (see details): A new, unused item with absolutely no signs of wear. The item may be missing the original packaging, or in the original packaging but not sealed. The item may be a factory second or a new, unused item with defects. See the seller's listing for full details and description of any imperfections.

SELECT FORMAT AND PRICE

* List as:

* Quantity:

* Price:

* Start Date(yyyy-mm-dd):

* End Date(yyyy-mm-dd):

I have read and agree to the eBay

Continue

Update a product

Lee ▾

ebay

PERSONS PRODUCTS BUYERS SELLERS AUCTIONS SEARCH PERSON

UPDATE PRODUCT DETAILS

Product Code :

* Product Title :

*Product Condition:

*Category:

*Subcategory:

*Available Quantity:

*Description:

*Price:

* Listing type:

Update

INFORMATION	CUSTOMER SERVICE	EXTRAS	MY ACCOUNT
About Us	Contact Us	Brands	My Account
Delivery Information	Returns	Gift Vouchers	Order History
Privacy Policy	Site Map	Affiliates	Wish List
Elements		Specials	Newsletter

 eBay © 2014 

Scalability, Performance and Reliability:

While developing a web application, many points are to be kept in mind – The data consistency, whether the user is accessing the updated data, which in turn signifies the reliability of a web site. Similarly, while most of the web applications are distributed, they should be capable of handling load of multiple users, keeping in consideration the speed of execution. The success of a web application relies majorly on these three paradigms. We have implemented the following techniques to make our more robust and adaptive:

SQL Caching

Database ‘Select’ is one of the most expensive operations and thus special consideration should be taken to optimize the searches. We have implemented the following techniques in our project:

Pagination: Redis Lists

As discussed before in [Redis Server](#), the listing is being done by saving the data in a result set, typically Redis Lists and shown in pages. In our system, we have implemented ‘pagination’ where each page displays 10 values. Thus, this implementation technique reduces the database hits and reducing usage of ‘Select’ statement. Each hit stored is update when the database is updated. Moreover, since the retrieval is being done from a list instead of database, the retrieval time reduces thereby increasing the efficiency of the system.

Redis Hash-Map Searches

Search module requires ‘Select’ statements which is the most expensive database operation. As discussed before in [Redis Server](#), to reduce cost of system and increase its efficiency we have implemented our searching module using hash maps, where we the keys are simple search queries and its corresponding result value is a json string. This technique reduces the number of database hits, thereby increasing the performance and reducing the cost drastically. When an update in database is encountered, the previous json string also gets updated.

Given below is an example wherein each time a user requests for a list of products, it would first check the redis list by using the key “listallProduct”, if it is present then it will fetch the data from there, otherwise it will call the ‘productlistunavailable’ function (displayed below) which fetches the data from the data base. The list is flushed each time there is an update, thereby ensuring that the data is consistent.

Code Snapshot 1: Function to list all Products

```
function listallProduct(req,res)
{
    var currentpage = req.query.currentpage;
    console.log("page : " + currentpage);
    globals.checkifListAvailable(req,res,"listallProduct",globals.listavailable,productListunavailable,currentpage);

}//getallbuyers
```

Code Snapshot 2: Method if listallProduct is empty; fetches data from database

```
//listallProduct
function productListunavailable(req,res)
{
    var getproducts = " select * from product"

    console.log("Query is:"+getproducts);
    var resultSet="";
    mysqlpool.execQuery(getproducts,'',function(err,results){
        if(err){
            console.log("Error while fetching buyers from db");
            throw err;
        }
        else
        {
            if(results.length > 0){
                console.log("Length"+results.length);
                resultSet = JSON.parse(JSON.stringify(results));
                console.log(resultSet);
                console.log("In adminProduct");

                globals.deletelist("listallProduct");
                globals.setval("listallProduct",resultSet);

                // do pagination
                globals.getval(res,"listallProduct",0,globals.numEntries);

            }//if
            else // no results
            {
                var resultSet=JSON.stringify("No results found");
                console.log(resultSet);
                res.render('adminProduct', { title : 'Admin/Products', error : "No Results found!" ,result: resultSet,user Name:"Admin"});
            }
        }//else
    });
}

}//listallProduct*/
```

Denormalization

The person table is a generic table for each user registered with the system. However, keeping the performance in mind, we have denormalized the person table into buyer and seller table. These tables are still linked to the person table with member id.

Views

In database, a query with joins shows poorer performance as compared to a direct query. Thus, as per our system's data model, auction module is designed as a view to avoid unnecessary number of joins between seller, product and auction table. Similarly, we have implemented product view. This not only saves time and increases throughput but is also less complicated.

Connection Pooling

We have implemented efficient way of managing database connection objects used for multiple connections required at different instance of time. This principle is helping our system in limiting the number of connection objects and reusing the pool of objects. We know that connecting to database is an expensive and slow process, thereby reducing the cost of our system and increasing the speed. Also, we know that it is easier if the database connection is taken care by a single module to maintain it at later stage and debug. Thus, this increases the maintainability and diagnostic efficiency of our system.

Query Builder

Keeping a realistic approach in mind, we have used SQUEL query builder to write queries for search modules. We all know that there are various parameters, based on which a user can search for a product or a person. Thus, its corresponding query cannot be static and the input search parameters are dynamic, decided at the runtime. Keeping this problem in mind and avoiding writing a large number of queries, we have used query builder, which would dynamically create queries at run time, as the user requests. Shown below is the reference code snapshot used in our system:

[Code Snapshot 3: Reference code Snapshot of SQUEL Query Builder](#)

```
var searchproduct = squel.select().from("product_view")

if (search !== '' && search !== undefined)
    searchproduct.where("product_title like '%" + search + "%'")

if (category !== '' && category !== undefined)
    searchproduct.where("category='%" + category + "'")

if (subcategory !== '' && subcategory !== undefined)
    searchproduct.where("subcategory='%" + subcategory + "'")

if (condition !== '' && condition !== undefined)
    searchproduct.where("product_condition='%" + condition + "'")

if (lowestprice !== '' && lowestprice !== undefined)
    searchproduct.where("instant_sellingprice>=" + lowestprice)

if (highestprice !== '' && highestprice !== undefined)
    searchproduct.where("instant_sellingprice<=" + highestprice)

if (listing_type !== '' && listing_type !== undefined)
    searchproduct.where("listing_type='%" + listing_type + "'")
```

Dependency Injection using Prepared Statement

Dependency injection is achieved in the system using prepared statement. Our prepared statements are nothing but parameterized queries which in turn do not rely on the client side sanitation and validation. Thus, we are avoiding building SQL code strings. So in totality we are doing the validation code on the server side and making it thin client and a fat server as per the set standards.

Code Snapshot 4: Reference snapshot for Dependency Injection from our code

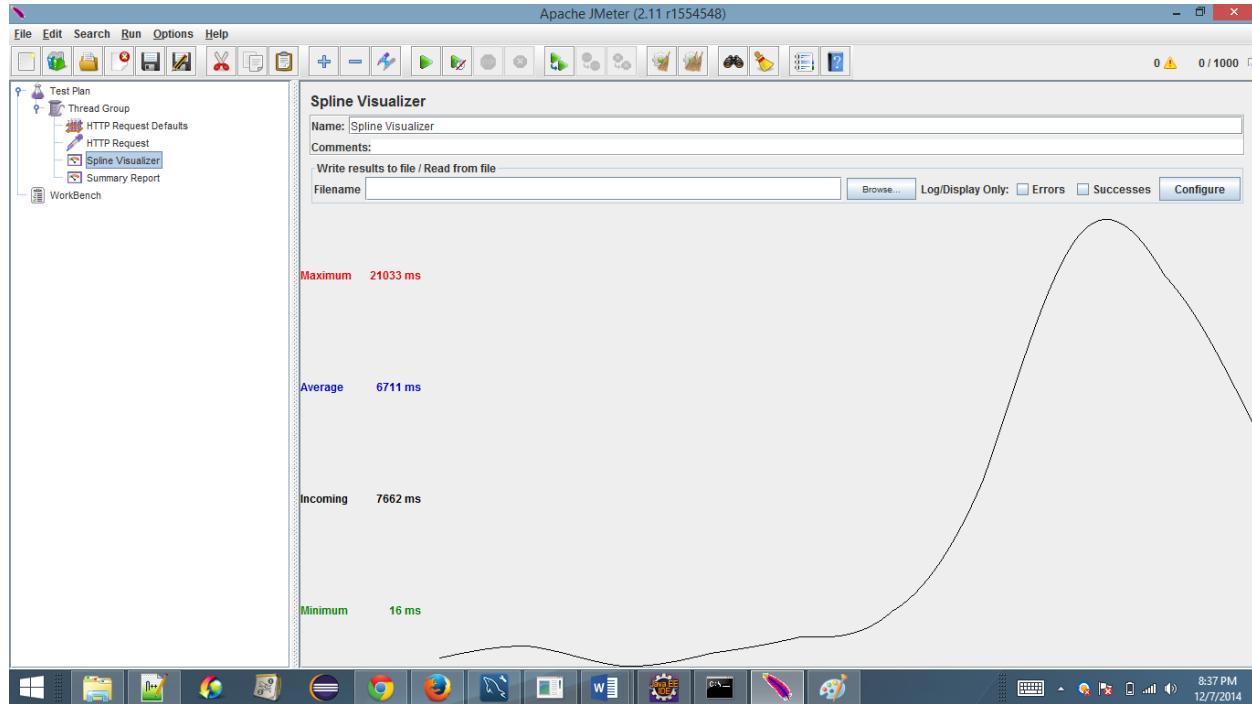
```
exports.userLogin = function (req,res){
    var user = req.param("inputEmail");
    var pwd = req.param("inputPwd");
    var cols = [];
    if(user !== null){
        globals.queryString = 'SELECT * FROM ?? WHERE ?? = ?';
        console.log("Query is : " + globals.queryString);

        // Connection Poling Call
        mysqlpool.execQuery(globals.queryString,['person','emailaddr',user],function(err,results){
            if(err){
                throw err;
            } else {
                if(results.length > 0){
                    var resultStr = JSON.parse(JSON.stringify(results));
                    if(resultStr[0].password === pwd){
                        globals.Auth = true;
                        globals.usrFName = resultStr[0].firstname;
                        globals.usrLName = resultStr[0].lastname;
                        globals.address = resultStr[0].address;
                        globals.city = resultStr[0].city;
                        globals.state = resultStr[0].state;
                        globals.country = resultStr[0].country;
                        globals.zipcode = resultStr[0].zipcode;
                        globals.email = resultStr[0].emailaddr;
                        globals.lastLogin = resultStr[0].lastlogin;
                        globals.user_Id = resultStr[0].memberid;
                        var admin = resultStr[0].admin;
                        if(admin == 'no')
                            res.render('userHome', { title: 'DashBoard/eBay', userName : globals.usrFName, result
                        else
                            res.render('admin', { title: 'DashBoard/eBay', userName : globals.usrFName, result : , });
                    }
                }
            }
        });
    }
}
```

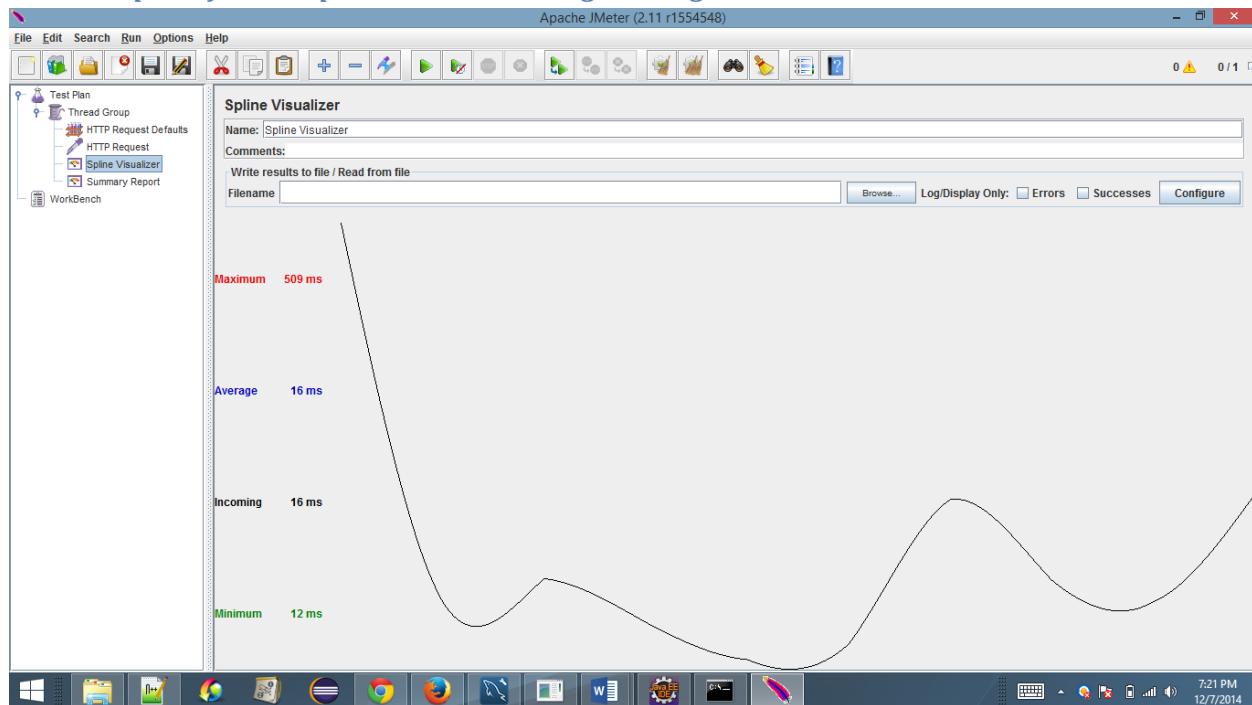
Testing

JMeter: Performance Testing

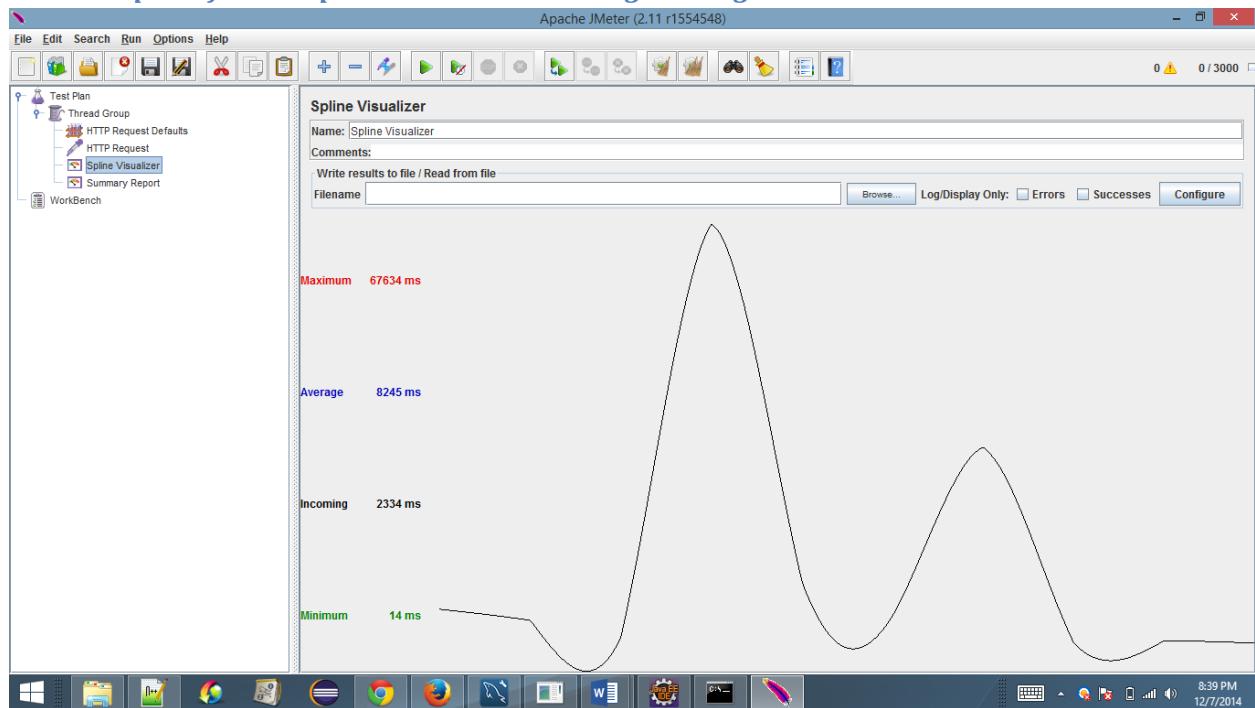
3000 Samples of search product without caching: Average is 6711ms



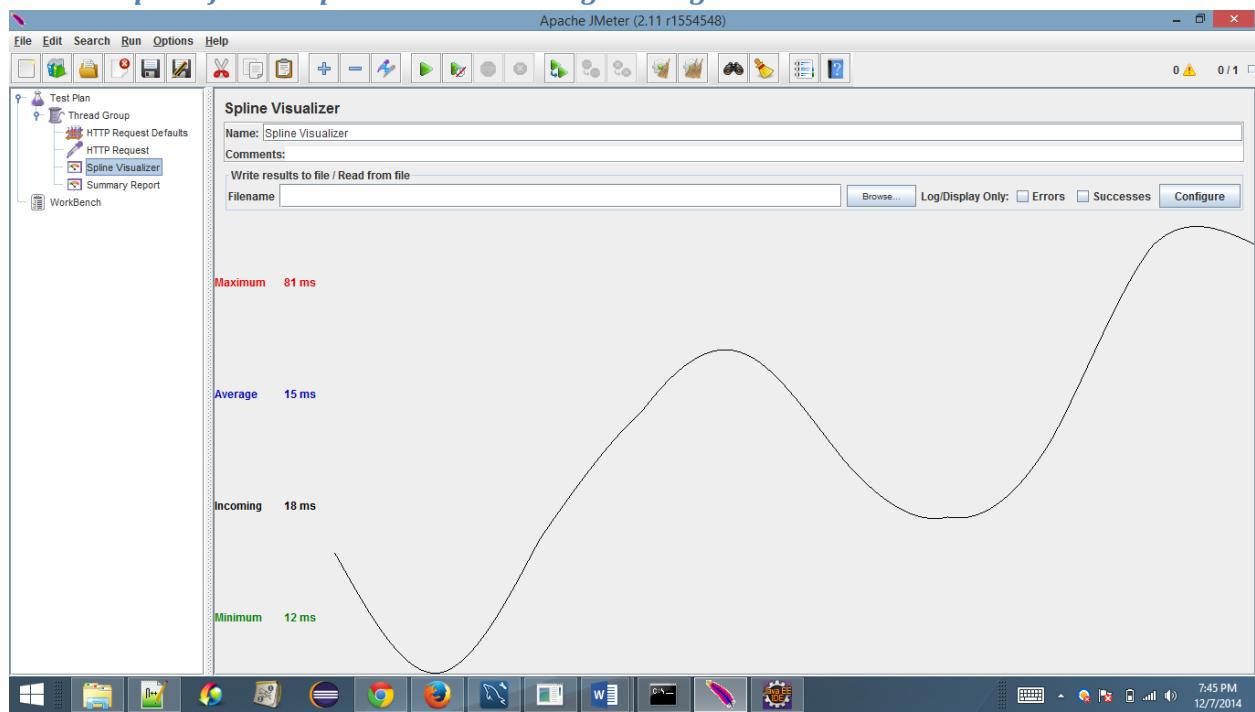
3000 Samples of search product with caching: Average is 16ms

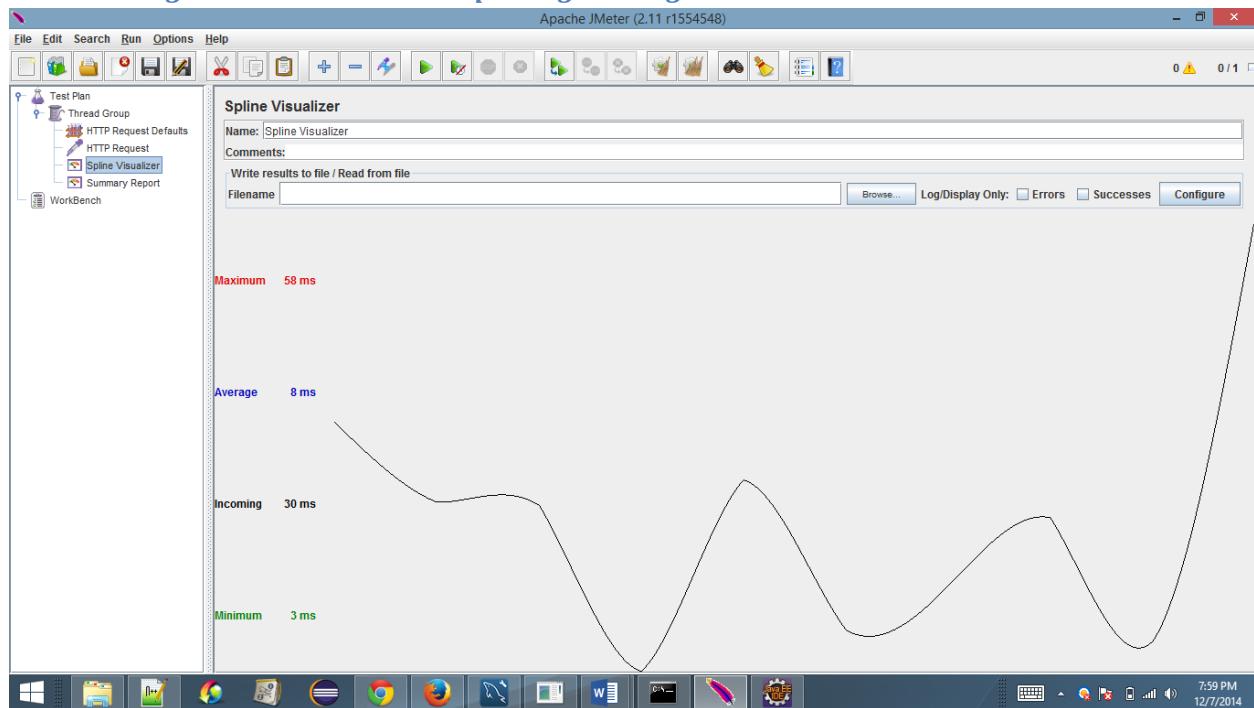
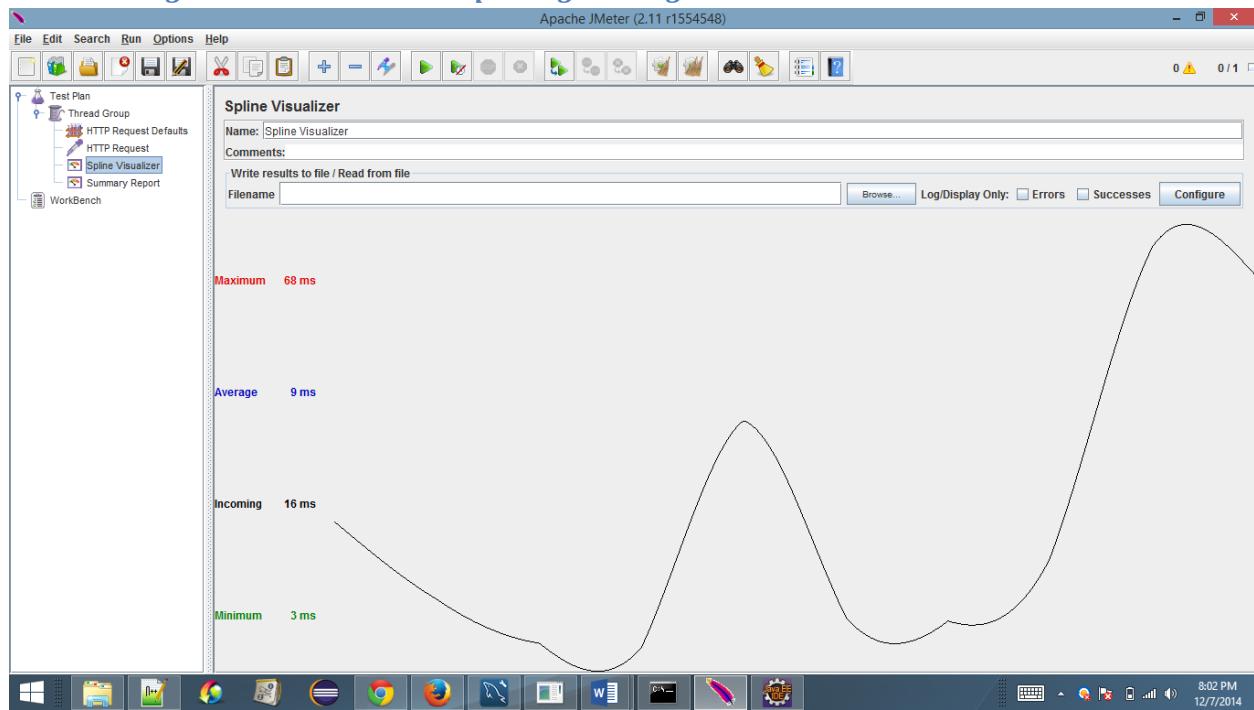


5000 Samples of search product without caching: Average is 8245ms

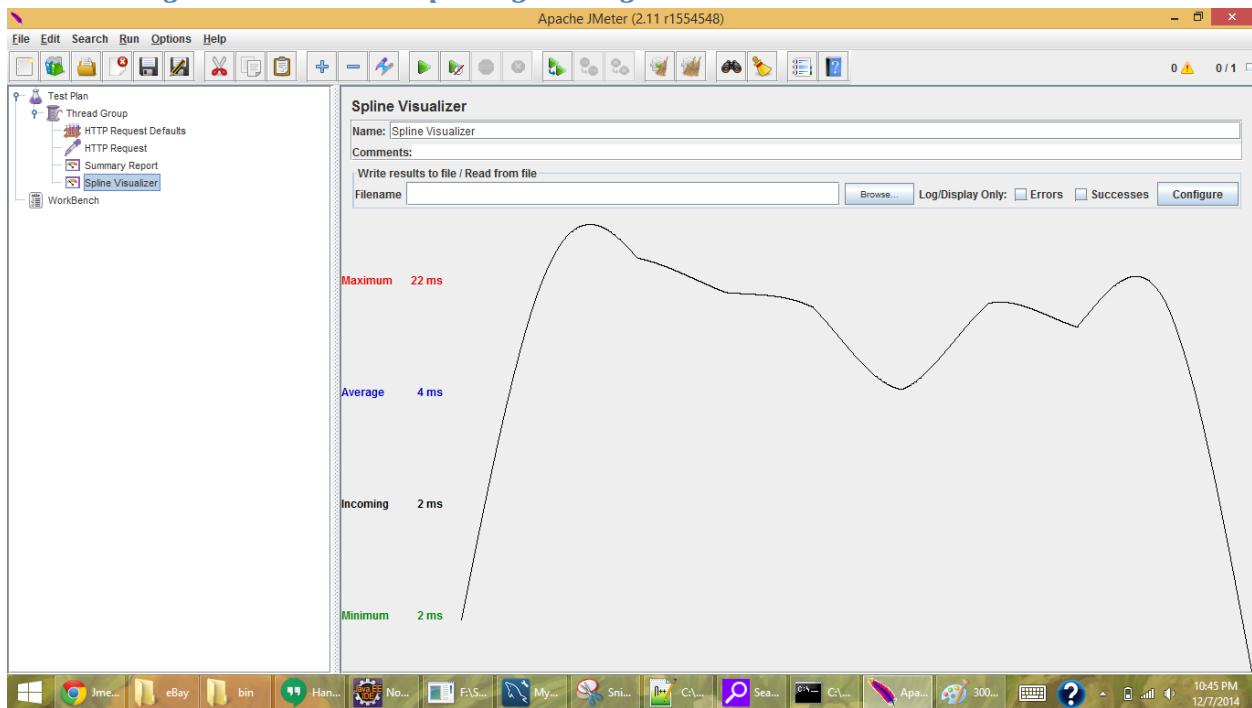


5000 Samples of search product with caching: Average is 15ms

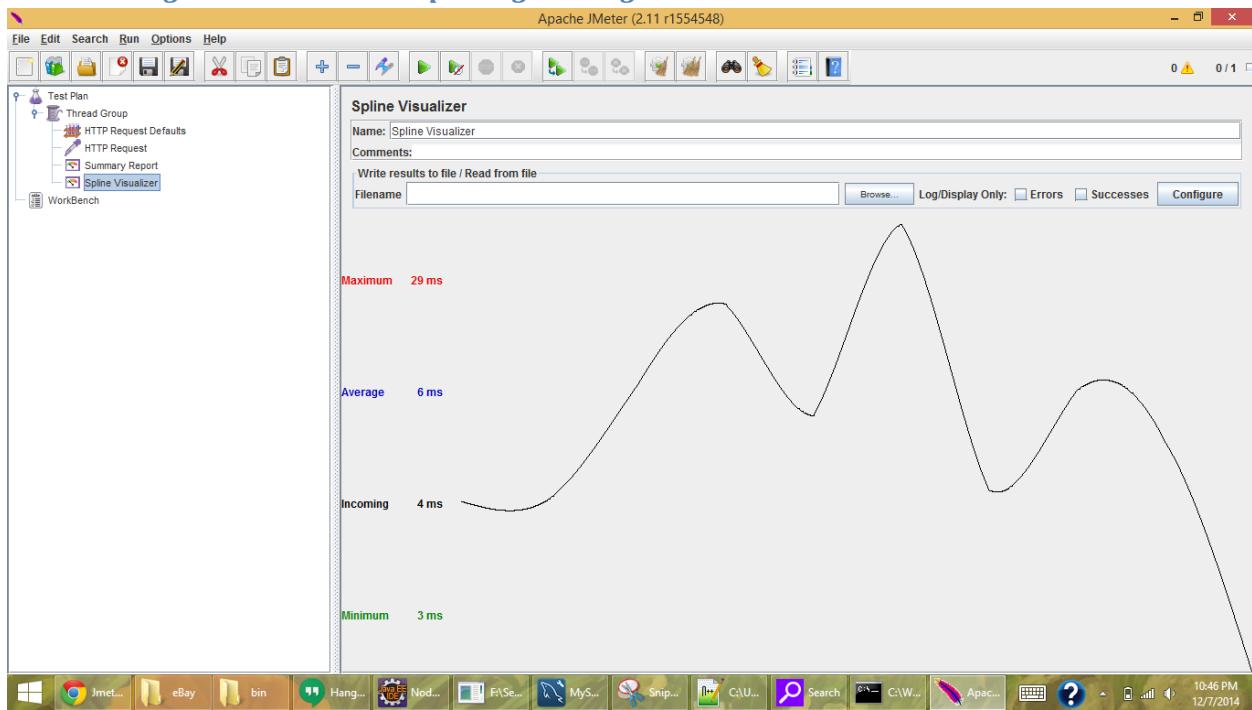


3000 user login without connection pooling: Average is 8ms***5000 user login without connection pooling: Average is 9ms***

3000 user login with connection pooling: Average is 4ms



5000 user login with connection pooling: Average is 6ms

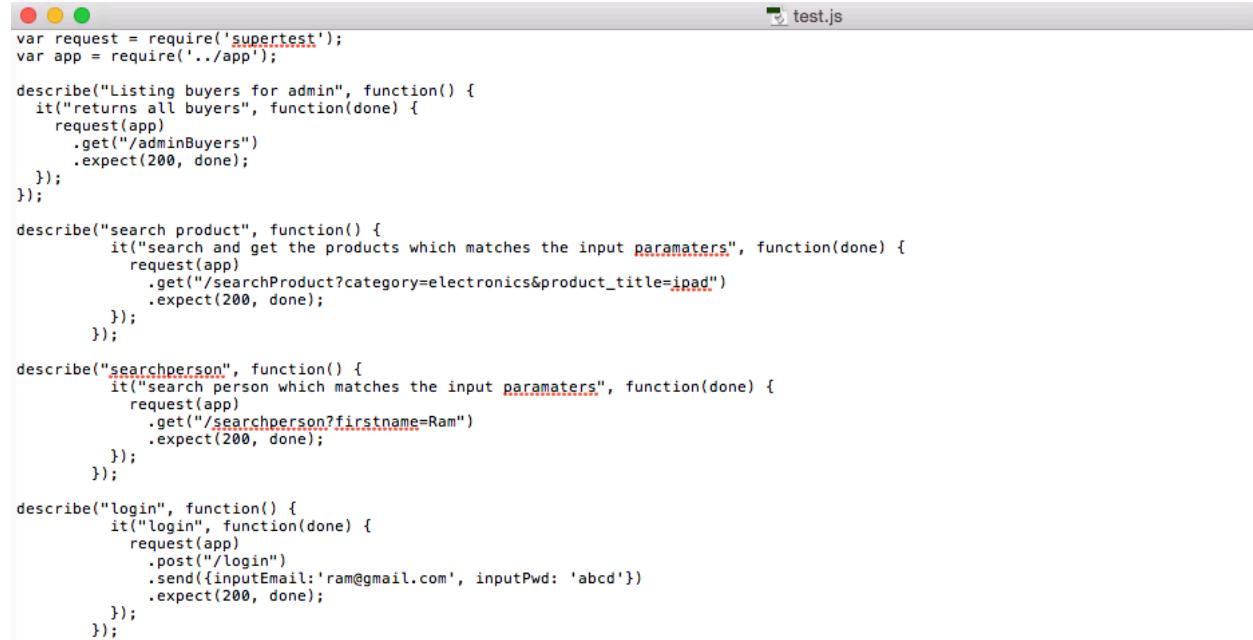


Mocha: Functional Testing

We have achieved functional testing using Mocha & Supertest i.e. libraries for testing. Following are the steps for testing using mocha

- Run Cygwin 64 Batch File
 - Route to Cygdrive containing the ebay testcases i.e. the test.js file
 - Run the following mocha command which is a shell script
- ```
$./node.modules/mocha/bin/mocha_test
```

## Test scripts for integration Testing using Browser & Jmeter



```
test.js
var request = require('supertest');
var app = require('../app');

describe("Listing buyers for admin", function() {
 it("returns all buyers", function(done) {
 request(app)
 .get("/adminBuyers")
 .expect(200, done);
 });
});

describe("search product", function() {
 it("search and get the products which matches the input parameters", function(done) {
 request(app)
 .get('/searchProduct?category=electronics&product_title=ipad')
 .expect(200, done);
 });
});

describe("searchperson", function() {
 it("search person which matches the input parameters", function(done) {
 request(app)
 .get('/searchperson?firstname=Ram')
 .expect(200, done);
 });
});

describe("login", function() {
 it("login", function(done) {
 request(app)
 .post("/login")
 .send({inputEmail:'ram@gmail.com', inputPwd: 'abcd'})
 .expect(200, done);
 });
});
```

## Testing Snapshots using Mocha

```

ar@Arthis /cygdrive/c/Users/arathi/workspace/eBay
$./node_modules/mocha/bin/mocha test

 Express server listening on port 3000
Query is: select buyerid,buyerfirstname,buyerlastname,product_title,buying_time
from buyer
Query is >>> select buyerid,buyerfirstname,buyerlastname,product_title,buying_
time from buyer
length2
[{ buyerid: 3,
 buyerfirstname: 'Sawyer',
 buyerlastname: 'Barnes',
 product_title: 'ipad',
 buying_time: '2014-11-30T00:33:00.000Z' },
 { buyerid: 5,
 buyerfirstname: 'Nomlanga',
 buyerlastname: 'Hinton',
 product_title: 'Kids belt',
 buying_time: '2014-11-30T04:50:09.000Z' }]
In adminBuyer
GET /adminBuyers 200 77ms - 9.17kb
Going to release DB connection to the Pool
Query is:SELECT * FROM product WHERE <category='electronics'>
Query is >>>SELECT * FROM product WHERE <category='electronics'>
length3702
GET /searchProduct?category=electronics&product_title=ipad 200 1128ms - 4.96mb
Going to release DB connection to the Pool
Query is:SELECT memberid, firstname, lastname, emailaddr FROM person WHERE <fir
stname='Ram'>
value----- [{"memberid":1,"firstname":"Ram","lastname":"Gamble","emailaddr":"
"ram@gmail.com"}]
cached
GET /searchperson?firstname=Ram 200 7ms - 104b
Query is : SELECT * FROM ?? WHERE ?? = ?
Query is >>>SELECT * FROM ?? WHERE ?? = ?
POST /login 200 71ms - 7.43kb
Going to release DB connection to the Pool
.

 4 passing (is)

ar@Arthis /cygdrive/c/Users/arathi/workspace/eBay
$

```

## Code Listing

### App.js

```
var express = require('express')
, routes = require('./routes')
, user = require('./routes/user')
, http = require('http')
, path = require('path')
, login = require('./routes/authentication')
, index = require('./routes/index')
, Bid = require('./routes/Bid')
, Buy = require('./routes/Buy')
, Sell = require('./routes/Sell')
, Auction = require('./routes/Auction')
, Categories = require('./routes/Categories')
, Person = require('./routes/Person')
, Product = require('./routes/Product')
, Payment = require('./routes/Payment')
, admin = require('./routes/admin');

var app = express();

// all environments
app.set('port', process.env.PORT || 3000);
app.set('views', __dirname + '/views');
app.set('view engine', 'ejs');
app.use(express.favicon());
app.use(express.logger('dev'));
app.use(express.bodyParser());
app.use(express.methodOverride());
app.use(app.router);
app.use(express.static(path.join(__dirname, 'public')));

// development only
if ('development' === app.get('env')) {
 app.use(express.errorHandler());
}

// LogIn & SignUp
app.get('/', index.index);
app.get('/login', login.signIn);
app.post('/login', login.userLogin);
app.get('/logout', login.logout);
app.get('/signup', login.signUp);
app.post('/signup', login.registerUser);

// Post Login Methods
app.post('/buy', Buy.buyProduct);
app.get('/sell', Sell.index);
app.post('/sell', Sell.sellProduct);
app.post('/bid', Bid.bidProduct);
app.get('/categories', Categories.fetchCategories);
app.get('/subcategories', Categories.fetchSubcategories);

app.get('/searchProduct', Product.searchProduct);
app.get('/fetchProductDetail', Product.fetchProductDetail);

//Admin functionalities
app.get('/adminProducts', admin.listallProduct);
app.get('/adminPersons', admin.listallPerson);
app.get('/adminBuyers', admin.listallbuyers);
app.get('/adminSellers', admin.listallsellers);
app.get('/adminAuctions', admin.listallauctions);
app.get('/delete', admin.deletefromtable);
app.get('/adminBuyersOfProduct', admin.listbuyersofProduct);
app.get('/adminSellersOfProduct', admin.listsellersofProduct);
app.post('/updateProduct', admin.updateProduct);
app.get('/searchPerson', admin.searchPerson);
app.get('/adminSearchPerson', admin.adminSearchPerson)

//User account details
app.post('/updatePerson', Person.updatePerson);
app.get('/sellingHistory', Person.sellingHistory);
app.get('/buyingHistory', Person.buyingHistory);
app.get('/biddingHistory', Person.biddingHistory);
app.get('/fetchPersonDetails', Person.fetchPersonDetails);

//Shopping Cart
app.get('/fetchfromcart', Product.fetchfromCart);
app.get('/addtocart', Product.addToCart);
app.get('/buynow', Product.buyNow);
app.get('/deletefromCart', Product.deletefromCart);

// payment detail mapping
app.get('/displayPayment', Payment.displayPayment);
app.post('/paymentDetails', Payment.paymentDetails);
```

## Admin

### *Admin Search Person*

```
|function adminSearchPerson(req,res)
{
 res.render('adminSearchPerson', { title : 'Admin/AdminSearchPerson', userName:"Admin"});
}
```

### *Auction list Unavailable*

```
|function auctionsListUnavailable(req,res)
{
 var getallAuction="select * from auction_report_view;";

 console.log("Query is:"+getallAuction);
 mysqlpool.execQuery(getallAuction,'',function(err,results){
 if(err){
 console.log("Error while fetching data");
 throw err;
 }
 else
 {
 if(results.length > 0){
 console.log("length"+results.length);
 json=JSON.stringify(results);
 resultSet = JSON.parse(JSON.stringify(results));
 console.log(resultSet);
 globals.deleteList("listallauctions");
 globals.setVal("listallauctions",results);

 // do pagination
 globals.getVal(res,"listallauctions",0,globals.numEntries);
 console.log("In adminAuctions");
 }//if
 else // no results
 {
 var resultSet=JSON.stringify("No results found");
 console.log(resultSet);
 res.render('adminAuctions', { title : 'Admin/Auctions', error : "No Results found!" ,result: resultSet, userName:"Admin"});
 }
 }//else
 });
}
```

### Delete Product , Person, Buyer and Seller

```
function deletefromtable(req,res){

 var id = req.query.id;
 var tableName = req.query.tableName;
 console.log("Id" +id);
 console.log("TableName" +tableName);
 var deleteData = '';
 if(tableName == "product")
 {
 deleteData = "delete from product where product_id="+id;
 }else if(tableName == "seller")
 {
 deleteData = "delete from buyer where sellerid="+id;
 }else if(tableName == "buyer")
 {
 deleteData = "delete from buyer where buyerid="+id;
 }

 mysqlpool.execQuery(deleteData,'',function(err,results){
 if(err){
 console.log("Error while deleteting Product in db");
 throw err;
 }else
 {
 if(tableName == "product")
 {
 listallProduct(req,res);
 }else if(tableName == "buyer")
 {
 listallbuyers(req,res);
 }else if(tableName == "seller")
 {
 listallsellers(req,res);
 }
 }
 });
}
```

### List All Product

```
function listallProduct(req,res)
{
 var currentpage = req.query.currentpage;
 console.log("page : " + currentpage);
 globals.checkifListAvailable(req,res,"listallProduct",globals.listavailable,productListunavailable,currentpage);
}
//getallbuyers
//list all person
```

### *List All Auctions*

```

function listallauctions(req,res)
{
 var currentpage = req.query.currentpage;
 globals.checkifListAvailable(req,res,"listallauctions",globals.listavailable,auctionsListunavailable,currentPage);

}//getallperson

```

### *List All Buyers*

```

function listallbuyers(req,res)
{
 var getallbuyers = "select buyerid,buyerfirstname,buyerlastname,product_title,buying_time from buyer"
 console.log("Query is:"+getallbuyers);
 var resultSet="";
 mysqlpool.execQuery(getallbuyers,'',function(err,results){
 if(err){
 console.log("Error while fetching buyers from db");
 throw err;
 }
 else
 {
 if(results.length > 0){
 console.log("length"+results.length);
 json=JSON.stringify(results);
 resultSet = JSON.parse(JSON.stringify(results));
 console.log(resultSet);
 console.log("In adminBuyer");

 res.render('adminBuyer', { title: 'Admin/Buyer', error : "", result : resultSet,userName:"Admin"});

 }//if
 else // no results
 {
 var resultSet=JSON.stringify("No results found");
 console.log(resultSet);
 res.render('adminBuyer', { title : 'Admin/Buyer', error : "No Results found!" ,result: resultSet,userName:"Admin"});
 }
 }//else
 });
}

//list all buyers
//list all sellers

```

### *List All Person*

```

function listallPerson(req,res)
{
 var currentpage = req.query.currentpage;
 globals.checkifListAvailable(req,res,"listallperson",globals.listavailable,personListunavailable,currentpage);

}//getallperson

//list all buyers

```

### *List All Sellers*

```

function listallsellers(req,res)
{
 var currentpage = req.query.currentpage;
 globals.checkifListAvailable(req,res,"listallsellers",globals.listavailable,sellersListunavailable,currentpage);

}//getallperson

```

### *List Buyers of a Product*

```

function listbuyersofProduct(req,res)
{
 var product_id = req.query.product_id;
 var getallbuyersofproduct = "select buyerfirstname,buyerlastname,product_title,buying_time from buyer where product_id =" + product_id;

 console.log("Query is:" + getallbuyersofproduct);

 mysqlpool.execSQL(getallbuyersofproduct,'',function(err,results){
 if(err){
 console.log("Error while fetching buyers from db");
 throw err;
 }
 else
 {
 if(results.length > 0){
 console.log("length" + results.length);
 json=JSON.stringify(results);
 resultSet = JSON.parse(JSON.stringify(results));
 console.log(resultSet);
 console.log("In admin buyer of a product");
 res.render('adminBuyersOfProduct', { title: 'Admin/adminBuyersOfProduct', error: "", result: resultSet,userName:"Admin"});
 } //if
 else // no results
 {
 var resultSet=JSON.stringify("No results found");
 console.log(resultSet);
 res.render('adminBuyersOfProduct', { title: 'Admin/adminBuyersOfProduct', error: "No Results Found!",result: resultSet,userName:"Admin"});
 }
 } //else
 });
}

//listbuyersofProduct

```

### *List Sellers of Product*

```

function listsellersofProduct(req,res)
{
 var product_id = req.query.product_id;
 var getallsellersofproduct = "select sellerfirstname,sellerlastname,product_title, selling_time from seller where product_id =" + product_id;

 console.log("Query is:" + getallsellersofproduct);

 mysqlpool.execSQL(getallsellersofproduct,'',function(err,results){
 if(err){
 console.log("Error while fetching sellers from db");
 throw err;
 }
 else
 {
 if(results.length > 0){
 console.log("length" + results.length);
 json=JSON.stringify(results);
 resultSet = JSON.parse(JSON.stringify(results));
 console.log(resultSet);
 console.log("In admin seller of a product");
 res.render('adminSellersOfProduct', { title: 'Admin/adminSellersOfProduct', error: "", result: resultSet,userName:"Admin"});
 } //if
 else // no results
 {
 var resultSet=JSON.stringify("No results found");
 console.log(resultSet);
 res.render('adminSellersOfProduct', { title: 'Admin/adminSellersOfProduct', error: "No Results Found!",result: resultSet,userName:"Admin"});
 }
 } //else
 });
}

//listsellersofProduct

```

### *Person List Unavailable*

```

function personListunavailable(req,res)
{
 var getallperson="select memberid,firstname,lastname,address,city,state,country,zipcode,emailaddr from person limit 30";
 console.log("Query is:"+getallperson);
 var resultSet="";
 mysqlpool.execQuery(getallperson,'',function(err,results){
 if(err){
 console.log("Error while fetching Person from db");
 throw err;
 }
 else
 {
 if(results.length > 0){
 console.log("length"+results.length);
 json=JSON.stringify(results);
 resultSet = JSON.parse(JSON.stringify(results));
 console.log(resultSet);
 console.log("In adminPerson");
 globals.deletelist("listallperson");
 globals.setval("listallperson",resultSet);

 // do pagination
 globals.getval(res,"listallperson",0,globals.numEntries);
 } //if
 else // no results
 {
 var resultSet=JSON.stringify("No results found");
 console.log(resultSet);
 res.render('adminPerson', { title : 'Admin/Person', error : "No Results found!" ,result: resultSet,userName:"Admin"});
 }
 } //else
 });
}

//list all person

```

### *Product List Unavailable*

```

function productListunavailable(req,res)
{
 var getproducts = " select * from product"

 console.log("Query is:"+getproducts);
 var resultSet="";
 mysqlpool.execQuery(getproducts,'',function(err,results){
 if(err){
 console.log("Error while fetching buyers from db");
 throw err;
 }
 else
 {
 if(results.length > 0){
 console.log("length"+results.length);
 resultSet = JSON.parse(JSON.stringify(results));
 console.log(resultSet);
 console.log("In adminProduct");

 globals.deletelist("listallProduct");
 globals.setval("listallProduct",resultSet);

 // do pagination
 globals.getval(res,"listallProduct",0,globals.numEntries);

 } //if
 else // no results
 {
 var resultSet=JSON.stringify("No results found");
 console.log(resultSet);
 res.render('adminProduct', { title : 'Admin/Products', error : "No Results found!" ,result: resultSet});
 }
 } //else
 });
}

```

## Search Person

```

function searchPerson(req,res)
{
 // check if user is admin
 var firstname=req.param("firstname");
 var lastname=req.param("lastname");
 var city=req.param("city");
 var state=req.param("state");
 var country=req.param("country");
 var zipcode=req.param("zipcode");

 var searchperson=squel.select()
 .field("firstname")
 .field("lastname")
 .field("address")
 .field("country")
 .from("person")

 if(firstname!="" && firstname!=undefined)
 searchperson.where("firstname='"+firstname+"'")
 if(lastname!="" && lastname != undefined)
 searchperson.where("lastname='"+lastname+"'")
 if(zipcode!="" && zipcode!= undefined)
 searchperson.where("zipcode='"+zipcode+"'")
 if(city!="" && city != undefined)
 searchperson.where("city='"+city+"'")
 if(state!="" && state!=undefined)
 searchperson.where("state='"+state+"'")
 if(country!="" && country!=undefined)
 searchperson.where("country='"+country+"'")

 searchperson+="\0";

 console.log("Query is:"+searchperson);
 var data= false;
 globals.exportclient().hmget("searchPerson", searchperson,function(error, result) {
 if (error) console.log('Error: ' + error);
 else {
 console.log('value----- ' + result);
 data=result;
 }
 }

 if(data==false && data != undefined)
 {
 //data cached already
 console.log("cached");
 res.render('searchPersonResults', { title: 'Admin/SearchPersonResults', error : "", result : data,userName:"Admin"});
 //res.jsonp(data);
 }
 else{

 mysqlpool.execQuery(searchperson,'',function(err,results){
 if(err){
 console.log("Error while fetching data");
 throw err;
 }
 else
 {

 var json="";
 if(results.length > 0){
 json=JSON.stringify(results);
 var resultSet = JSON.parse(JSON.stringify(results));
 console.log(resultSet);

 globals.exportclient().hdel("searchPerson",searchperson);
 globals.exportclient().hset("searchPerson",searchperson,json);
 res.render('searchPersonResults', { title: 'Admin/SearchPersonResults', error : "", result : resultSet,userName:"Admin"});

 }/if
 else // no results
 {
 var resultSet=JSON.stringify("No results found");
 console.log(resultSet);
 res.render('searchPersonResults', { title : 'Admin/SearchPersonResults', error : "No Results found!",result:resultSet});
 }
 }/else
 });
 }/else // data not cached
});/else hmget
);
}

```

### *Seller List Unavailable*

```

function sellersListunavailable(req,res)
{
 var getallbuyers = "select sellerid,sellerfirstname, sellerlastname,product_title,selling_time from seller";
 console.log("Query is:"+getallbuyers);
 var resultSet="";
 mysqlpool.execQuery(getallbuyers,'',function(err,results){
 if(err){
 console.log("Error while fetching sellers from db");
 throw err;
 }
 else
 {
 if(results.length > 0){
 console.log("length"+results.length);
 json=JSON.stringify(results);
 resultSet = JSON.parse(JSON.stringify(results));
 console.log(resultSet);
 console.log("In adminSeller");
 globals.deletelist("listallsellers");
 globals.setval("listallsellers",resultSet);

 // do pagination
 globals.getval(res,"listallsellers",0,globals.numEntries);

 } //if
 else // no results
 {
 var resultSet=JSON.stringify("No results found");
 console.log(resultSet);
 res.render('adminSeller', { title : 'Admin/Seller', error : "No Results found!" ,result: resultSet, userName:"Admin"});
 }
 } //else
 });
}

}//listallsellers

```

### *Update Product*

```

function updateProduct(req, res) {
 var updateproduct = "update product set product_title ='"
 + req.param("product_title") + "' ,product_condition ='"
 + req.param("product_condition") + "' ,category='"
 + req.param("category") + "' ,subcategory ='"
 + req.param("subcategory") + "' ,available_quantity ='"
 + req.param("available_quantity") + "' ,product_description='"
 + req.param("product_description") + "' ,instant_sellingprice='"
 + req.param("instant_sellingprice") + "' ,listing_type='"
 + req.param("listing_type") + "' " + " where product_id="
 + req.param("product_id");

 console.log("Query is:" + updateproduct);

 mysqlpool.execQuery(updateproduct, '', function(err, results) {
 if (err) {
 console.log("Error while updating Product into db");
 throw err;
 } else
 {
 listallProduct(req,res);
 }
 });
}

}//updateProduct

```

## Auction

```

function auction() {
 console.log("inside auction");

 // check for the auctions that has end just now, not the ones that has been closed already
 // live auctions wont have buyer_id in them
 var checkAuctionTime="select auction_id from auction where TIMESTAMPDIFF(MINUTE,now(), auction_end_date)<=0 and buyer_id is null";
 console.log("Query is:"+checkAuctionTime);

 mysqlpool.execQuery(checkAuctionTime,'',function(err,results){
 if(err){
 console.log("Error while fetching data");
 throw err;
 }
 else
 {
 // any auction has end
 var json="";
 if(results.length > 0){

 console.log("length"+results.length);

 //for each auction that has end, do the following
 for(var i=0;i<results.length;i++)
 {
 // to check highest bid for an auction
 var auctid = results[i].auction_id;
 var fetchLastBidder="select * from bid where auction_id='"+results[i].auction_id+"' and bid_price=(select MAX(bid_price) from bid
 console.log("Query is:"+fetchLastBidder);

 mysqlpool.execQuery(fetchLastBidder,'',function(err,bidresults){
 if(err){
 console.log("Error while fetching data");
 throw err;
 }
 else
 {
 //update in buyer table,
 if(bidresults.length > 0){
 var buyproduct="insert into buyer (product_id,buyerid,buying_time,buyerfirstname,buyerlastname,product_title) values
 bidresults[0].product_id+"','"+bidresults[0].bidder_id+"','"+now()+"','"+bidresults[0].bidderfirstname+"','"++
 bidresults[0].bidderlastname+"','"+bidresults[0].product_title+"')";
 console.log("Query is:"+buyproduct);

 mysqlpool.execQuery(buyproduct,'',function(err,buyresults){
 if(err){
 console.log("Error while inserting into db");
 throw err;
 }
 });
 }
 // },buyproduct);

 // update auction table for setting the buyer id
 var updateAuction = "update auction set buyer_id="+bidresults[0].bidder_id+ " where auction_id="+auctid;
 console.log("Query is:" + updateAuction);

 mysqlpool.execQuery(updateAuction,'',function(err,update$results){
 if(err){
 console.log("Error while inserting into db");
 throw err;
 }
 });
 //},updateAuction);

 // update product table to reduce quantity // for auction we consider that we are selling all the quantities of a p
 var updateProduct = "update product set available_quantity=available_quantity-1 where product_id="+bidresults[0].pr
 console.log("Query is:" + updateProduct);

 mysqlpool.execQuery(updateProduct,'',function(err,updateprod$results){
 if(err){
 console.log("Error while inserting into db");
 throw err;
 }
 });
 }
 });
 }
 }
 }
 });
}

```

*Check Auction*

```
function checkAuction()
{
 setInterval(auction,60000); //1000ms = 1 second ; 60,000 ms = 1 minute
 //auction();
}
```

*Get Auction*

```
function getAuction(req,res)
{
 var getallAuction="select * from auction_report_view;";

 console.log("Query is:"+getallAuction);
 mysqlpool.execQuery(getallAuction,'',function(err,results){
 if(err){
 console.log("Error while fetching data");
 throw err;
 }
 else
 {

 var json="";
 if(results.length > 0){

 console.log("length"+results.length);
 json=JSON.stringify(results);
 console.log(json);
 res.jsonp(json);

 } //if
 } //else
 },getallAuction);
}
```

## Authentication

### Login

```

exports.userLogin = function (req,res){
 var user = req.param("inputEmail");
 var pwd = req.param("inputPwd");
 var cols = [];
 if(user !== null){
 globals.queryString = 'SELECT * FROM ?? WHERE ?? = ?';
 console.log("Query is : " + globals.queryString);

 // Connection Poling Call
 mysqlpool.execQuery(globals.queryString,['person','emailaddr',user],function(err,results){
 if(err){
 throw err;
 } else {
 if(results.length > 0){
 var resultStr = JSON.parse(JSON.stringify(results));
 if(resultStr[0].password === pwd){
 globals.Auth = true;
 globals usrFName = resultStr[0].firstname;
 globals usrLName = resultStr[0].lastname;
 globals.address = resultStr[0].address;
 globals.city = resultStr[0].city;
 globals.state = resultStr[0].state;
 globals.country = resultStr[0].country;
 globals.zipcode = resultStr[0].zipcode;
 globals.email = resultStr[0].emailaddr;
 globals.lastLogin = resultStr[0].lastlogin;
 globals.user_Id = resultStr[0].memberid;
 var admin = resultStr[0].admin;
 if(admin == 'no'){
 res.render('userHome', { title: 'DashBoard/eBay', userName : globals.usrFName, result : globals.resultSet, userId:globc
 else
 res.render('admin', { title: 'DashBoard/eBay', userName : globals.usrFName, result : globals.resultSet});
 } else {
 console.log("Invalid Login password");
 res.render('signin', { title : 'LogIn/eBay', error : 'Invalid Email Id or Password.'});
 }
 } else {
 console.log("Invalid Email Id");
 res.render('signin', { title : 'LogIn/eBay', error : 'Invalid Email Id or Password.'});
 }
 }
 }
 });
 }
}
```

```

Logout

```

exports.logout = function(req , res){
    globals usrName = '';
    globals.lastLogin = '';
    globals.user_Id = '';
    globals.queryString = '';
    globals.Auth = false;
    console.log("Logout Session From Site");
    res.render('signin', { title : 'LogIn/eBay', error : ''});
};

```

Register User/Sign Up

```

exports.registerUser = function (req,res){
    globals.usrName = req.param("firstname");
    var lname = req.param("lastname");
    globals.email = req.param("email");
    var pwd = req.param("password");
    var addr = req.param("address");
    var city = req.param("city");
    var zip = req.param("zipcode");
    var state = req.param("state");
    var country = req.param("country_id");
    var acc_no = 12456789;

    globals.queryString = "select ?? from ?? where emailaddr = ?";
    mysqlpool.execQuery(globals.queryString,['emailaddr','person',globals.email],function(err,results){
        console.log("Result : " + results);
        if(results.length === 0){
            //globals.queryString = "insert into person (firstName,lastName,emailaddr,password,address,city,zipcode,state,country,lastlogin,payemer
            var posts = {firstName : globals.usrName, lastName : lname, emailaddr : globals.email, password : pwd, address : addr, city : city, zip
            globals.queryString = "insert into person SET ?";
            console.log("SignUp Query is:" + globals.queryString);
            mysqlpool.execQuery(globals.queryString, posts, function(err,results){
                if(err){
                    throw err;
                } else {
                    var cols = ['memberid'];
                    globals.queryString = "select ?? from ?? where emailaddr = ?";
                    console.log("Inner Query :: " + globals.queryString);
                    mysqlpool.execQuery(globals.queryString,[cols,'person',globals.email],function(err,results){
                        if(err){
                            throw err;
                        } else {
                            if(results.length > 0){
                                var resultStr = JSON.parse(JSON.stringify(results));
                                globals.usrFName = resultStr[0].firstname;
                                globals.usrLName = resultStr[0].lastname;
                                globals.address = resultStr[0].address;
                                globals.country = resultStr[0].country;
                                globals.zipcode = resultStr[0].zipcode;
                                globals.user_Id = resultStr[0].memberid;
                                globals.lastLogin = resultStr[0].lastLogin;
                                res.render('userHome', { title: 'DashBoard/eBay',userName : globals.usrName, result: resultSet, userId:globals.user
                            } else {
                                console.log('Cannot Create User');
                                res.render('signup', { title : 'Register/eBay', error : 'Error Occured Cannot Create User.'});
                            }
                        }
                    });
                }
            });
        } else {
            console.log("User Already Exist...");
            res.render('signup', { title : 'Register/eBay', error : 'User Already Exists'});
        }
    });
}

```

Bid

```

function bidProduct(req, res) {
    // add an entry in bid table
    var dt = new Date();
    var bidid = dt.getTime();
    console.log("Auc Id : " + req.param("auction_id"));
    var insertBid = "insert into bid (auction_id,buyer_id,product_id,bid_price,bid_time) values ("+req.param("auction_id")+",";
    globals.user_id + "," + req.param("product_id") + "," + req.param("bid_price") + ",now())";

    console.log("Query is:" + insertBid);

    mysqlpool.execQuery(insertBid, '', function(err, results){
        if(err){
            console.log("Error while inserting into db");
            throw err;
        } else {
            res.end();
        }
    });

    //update current price in product table
    var updateProduct = "update product set instant_sellingprice=" + req.param("bid_price") + " where product_id=" + req.param("product_id");
    console.log("Query is:" + updateProduct);

    mysqlpool.execQuery(updateProduct, '', function(err, results){
        if(err){
            console.log("Error while inserting into db");
            throw err;
        } else {
            res.end();
        }
    });
}
//bidproduct

```

Buy Product

```

function buyProduct(req, res) {
    // add an entry in buyers table

    var buyproduct="insert into buyer (product_id,buyerid,buying_time) values ("+ req.param("product_id") + "," +req.param("buyerid") +",now())";
    console.log("Query is:" +buyproduct);

    mysqlpool.execQuery(buyproduct, '', function(err, results){
        if(err){
            console.log("Error while inserting into db");
            throw err;
        }
    });

    // reduce the quantity and add payment details in product table
    var updateProduct = "update product set available_quantity=available_quantity-" +req.param("quantity") + " where product_id=" +req.param("product_id");

    console.log("Query is:" + updateProduct);

    mysqlpool.execQuery(updateProduct, '', function(err, results){
        if(err){
            console.log("Error while inserting into db");
            throw err;
        }
    });
}

// buy product

```

Categories

```
function fetchCategories(req,res)
{
    var getCategories="select distinct(category) from product";
    console.log("Query is:"+getCategories);
    mysqlpool.execQuery(getCategories,'',function(err,results){
        if(err){
            console.log("Error while fetching data");
            throw err;
        }
        else
        {

            var json="";
            if(results.length > 0){

                console.log("length"+results.length);
                json=JSON.stringify(results);
                console.log(json);
                res.jsonp(json);

            } //if
        } //else
    });
}

//fetchCategories

function fetchSubcategories(req,res)
{
    var getSubCategories="select distinct(subcategory) from product where category = '"+req.param("category")+"'";
    console.log("Query is:"+getSubCategories);
    mysqlpool.execQuery(getSubCategories,'',function(err,results){
        if(err){
            console.log("Error while fetching data");
            throw err;
        }
        else
        {

            var json="";
            if(results.length > 0){

                console.log("length"+results.length);
                json=JSON.stringify(results);
                console.log(json);
                res.jsonp(json);

            } //if
        } //else
    });
}

//fetchSubcategories
```

Payment

Payment Details

```

exports.paymentDetails = function (req,res){
    console.log("Inside paymentDetails function...");
    var pwd = req.param("password");
    var payment_type = req.param("payment_type");
    var acc_no = req.param("payment_accno");

    //console.log('R : ' + globals.cartRes);
    //console.log("S Len : " + globals.cartRes.length);
    // add an entry in buyers table
    for(var i=0; i < globals.cartRes.length; i++){
        console.log('i : ' + globals.cartRes[i].product_id);
        var buyproduct = "insert into buyer (product_id,buyerid,buying_time,buyerfirstname,buyerlastname,product_t";
        console.log("Query is:" + buyproduct);
        mysqlpool.execQuery(buyproduct,'',function(err,results){
            if(err){
                console.log("Error while inserting into db");
                res.render('error', {title : 'Error/eBay' , });
            } else {
                console.log("Successfully inserted into db");
            }
            res.end();
        });

        // reduce the quantity and add payment details in product table
        var updateProduct = "update product set available_quantity=available_quantity-" + globals.cartRes[i].quant;
        console.log("Query is:" + updateProduct);

        console.log("before break");
        mysqlpool.execQuery(updateProduct,'',function(err,results){
            console.log("Result : " + results);
            if(err){
                console.log('Error');
            }
        });
    }
}

```

Display Payments

```

exports.displayPayment = function(req,res){
    console.log('In Payment Details');
    var cartVal = "select product_id,product_title,product_price,quantity,product_price*quantity AS 'Sum' from sl";
    console.log("Query is:"+cartVal);
    mysqlpool.execQuery(cartVal,'',function(err,results){
        if(err){
            console.log("Error while fetching from db for cart");
            throw err;
        }else{
            if(results.length > 0){
                globals.cartRes = JSON.parse(JSON.stringify(results));
                console.log("length"+globals.cartRes.length);
                console.log(globals.cartRes);
                res.render('paymentDetails', {title : 'payment/eBay' , userName : globals usrFName });
            }
        }
    });
}

```

Person

Buying History

```
//given a person id give both what he has bought..

function buyingHistory(req,res)
{
    var buyinghistory = "select a.product_title,c.instant_sellingprice, c.category from buyer a , person b, product c where a.buyerid="+globals.us
    console.log("Query is:"+buyinghistory);

    mysqlpool.execQuery(buyinghistory,'',function(err,results){
        if(err){
            console.log("Error while fetching buyer from db");
            throw err;
        }
        else
        {
            if(results.length > 0){
                console.log("Length"+results.length);
                json=JSON.stringify(results);
                resultSet = JSON.parse(JSON.stringify(results));
                console.log(resultSet);
                console.log("In buyingHistory");
                res.render('buyingHistory', { title: 'Ebay/BuyingHistory' , error : "" , result : resultSet,firstName:globals.usrFName, address:globa
            });//if
            else // no results
            {
                var resultSet=JSON.stringify("No results found");
                console.log(resultSet);
                res.render('buyingHistory', { title : 'Ebay/BuyingHistory' , error : "No Results Found!" ,address:globals.address,result : resultSet
            }
        };//else
    });
}

//buyinghistory
... ...
... ...
... ...
```

Delete Person

```

function deletePerson(req,res)
{
    var deleteperson="delete from person where memberid="+req.param("memberid");

    console.log("Query is:"+deleteperson);

    mysqlpool.execQuery(deleteperson,'',function(err,results){
        if(err){
            console.log("Error while deleting Person in db");
            throw err;
        }
    });

}//deleteperson

```

Fetch Person Detail

```

//Fetch person details
function fetchPersonDetails(req,res)
{
    var isUpdate = req.query.isUpdate;
    var personDetails="select * from person where memberid="+globals.user_Id;

    console.log("Query is:"+personDetails);
    mysqlpool.execQuery(personDetails,'',function(err,results){
        if(err){
            console.log("Error while fetching from db for cart");
            throw err;
        }else{
            if(results.length > 0){
                console.log("length"+results.length);
                var resultSet = JSON.parse(JSON.stringify(results));
                globals usrFName = resultSet[0].firstname;
                globals usrLName = resultSet[0].lastname;
                globals address = resultSet[0].address;
                globals city = resultSet[0].city;
                globals state = resultSet[0].state;
                globals country = resultSet[0].country;
                globals zipcode = resultSet[0].zipcode;
                console.log(resultSet);
                if(isUpdate == "yes")
                {
                    res.render('updateProfile', { title: 'Ebay/UpdateProfiel', error : "", firstName:globals.usrFName, lastName:globals.usrLName });
                }else
                {
                    res.render('userProfile', { title: 'Ebay/PersonDetails', error : "", firstName:globals.usrFName, lastName:globals.usrLName });
                }
            } //if
            else // no results
            {
                var resultSet=JSON.stringify("No results found");
                console.log(resultSet);
                if(isUpdate == "yes")
                {
                    res.render('updateProfile', { title: 'Ebay/UpdateProfiel', error : "", firstName:globals.usrFName, lastName:globals.usrLName });
                }else
                {
                    res.render('userProfile', { title: 'Ebay/PersonDetails', error : "", firstName:globals.usrFName, lastName:globals.usrLName });
                }
            }
        }
    });
}

```

Selling History

```

function sellingHistory(req,res)
{
    var sellinghistory = "select a.product_title,c.category,c.instant_sellingprice from seller a , person b, product c where a.sellerid='"+globals.u
    console.log("Query is:"+sellinghistory);

    mysqlpool.execQuery(sellinghistory,'',function(err,results){
        if(err){
            console.log("Error while fetching sellers from db");
            throw err;
        }
        else
        {
            if(results.length > 0){
                console.log("Length"+results.length);
                json=JSON.stringify(results);
                resultSet = JSON.parse(JSON.stringify(results));
                console.log(resultSet);
                console.log("In sellinghistory");
                res.render('sellingHistory', { title: 'Ebay/SellingHistory', error : "", result : resultSet,firstName:globals.usrfName, address:glc
            });//if
            else // no results
            {
                var resultSet=JSON.stringify("No results found");
                console.log(resultSet);
                res.render('sellingHistory', { title : 'Ebay/SellingHistory', error : "No Results found!" ,result : resultSet,firstName:globals.usr
            }
        };//else
    });
}

}//sellingHistory

```

Update Person

```

//function updatePerson(firstname,lastname,address, city, state, country,zipcode, emailaddr, password, payment_type, payment_accno,memberid)

function updatePerson(req,res)
{
    var updateperson="update person set firstname ='"+req.param("firstname")+"',lastname ='"+req.param("lastname")+"',address='"+
    req.param("address")+"',city ='"+ req.param("city")+"',state ='"+req.param("state")+"',country='"+req.param("country")+
    "',zipcode='"+req.param("zipcode")+"' where memberid='"+globals.user_Id;

    console.log("Query is:"+updateperson);

    mysqlpool.execQuery(updateperson,'',function(err,results){
        if(err){
            console.log("Error while inserting update Person into db");
            throw err;
        }
        else
        {
            fetchPersonDetails(req,res);
        }
    });
}

}//updatePerson

```

Product

Add to Cart

```
exports.addtoCart = function (req,res) {
  var pid = req.query.id;
  var qty = req.query.qty;
  var title = req.query.title;
  var price = req.query.price;
  if(qty == undefined)
    qty = 1;
  console.log("id : " + pid);

  var checkshoppingCart="select quantity from shoppingcart where product_id= "+ pid + " and buyer_id=" + globals.user_Id;
  console.log("Query is:"+checkshoppingCart);
  mysqlpool.execQuery(checkshoppingCart,'',function(err,results){
    if(err){
      console.log("Error while inserting into db");
      throw err;
    } else {
      if(results.length > 0){
        var p_qty = results[0].quantity + qty;
        var updateQuantity="UPDATE shoppingcart SET quantity=" + p_qty + " WHERE product_id=" + pid + " and buyer_id=" + globals.user_Id;
        console.log("Query is:"+updateQuantity);
        mysqlpool.execQuery(updateQuantity,'',function(err,results){
          if(err){
            console.log("Error while inserting into db");
            throw err;
          } else {
            res.redirect("/fetchfromCart");
          }
        });
      } else {
        var addtocart="insert into shoppingcart (buyer_id,product_id,product_title,product_price,quantity) values ("+
          globals.user_Id + "," + pid + "," + title + "," + price + "," + qty + ")";
        console.log("Query is:"+addtocart);
        mysqlpool.execQuery(addtocart,'',function(err,results){
          if(err){
            console.log("Error while inserting into db");
            throw err;
          } else {
            res.redirect("/fetchfromCart");
          }
        });
      }
    }
  });
}
```

Buy Now

```
exports.buynow = function (req,res) {
  var pid = req.query.id;
  var qty = req.query.qty;
  var title = req.query.title;
  var price = req.query.price;
  var sum = 0;

  if(qty == undefined){
    qty = 1;
    sum = qty * price;
  } else {
    sum = qty * price;
  }

  var json = "[{" + "product_id :" + pid + "," + "product_title :" + title + "," + "quantity :" + qty + "," + "price :" + price + "," + "total :" + sum + "}]";
  globals.cartRes = JSON.parse(JSON.stringify(eval("(" + json + ")")));
  res.render('paymentDetails', {title : 'payment/eBay', userName : globals usrFName});
};
```

Delete from Cart

```
//Delete from cart
exports.deletefromCart = function (req,res) {
    var deletefromCart="delete from shoppingcart where buyer_id='"+globals.user_Id+"' and product_id='"+req.query.id+
    console.log("Query is:"+deletefromCart);

    mysqlpool.execQuery(deletefromCart,'',function(err,results){
        if(err){
            console.log("Error while deleting Product from Shopping Cart");
            throw err;
        }
        else
        {
            res.redirect("/fetchfromCart");
            //res.render('shoppingCart', { title : 'shoppingCart', userName : globals usrFName, error : "No result found" })
        }
    });
};
```

Delete Product

```
//delete product
// product_id from front end
function deleteProduct(req,res){
    var deleteproduct="delete from product where product_id="+req.param("productid");
    console.log("Query is:"+deleteproduct);

    mysqlpool.execQuery(deleteproduct,'',function(err,results){
        if(err){
            console.log("Error while deleting Product in db");
            throw err;
        }
    });
};//deleteProduct
```

Fetch from Cart

```
exports.fetchfromCart = function (req,res) {
    var fetchfromcart="select product_id,product_title,product_price,quantity,product_price*quantity AS 'Sum' from shoppingcart where buyer_id='"+globals.user_Id+"';

    console.log("Query is:"+fetchfromcart);
    mysqlpool.execQuery(fetchfromcart,'',function(err,results){
        if(err){
            console.log("Error while fetching from db for cart");
            throw err;
        }
        else{
            if(results.length > 0){
                var total = 0;
                var len = results.length;
                console.log("Length "+results.length);
                resultSet = JSON.parse(JSON.stringify(results));
                while(len > 0){
                    total += resultSet[len-1].Sum;
                    len--;
                }
                console.log(resultSet);
                res.render('shoppingCart', { title: 'shoppingCart', userName : globals usrFName, error : "", result : resultSet, tot : total});
            } //if
            else // no results
            {
                var resultSet = JSON.stringify("No results found");
                console.log(resultSet);
                res.render('shoppingCart', { title : 'shoppingCart', userName : globals usrFName, error : "No results found", result : resultSet, t
            }
        } //else
    });
};
```

Fetch Product Detail

```
//Fetch Product Detail
exports.fetchProductDetail = function(req,res){
    //<a href="fetchProductDetail?product_id=<%= resp.product_id%>&product_title=<%= resp.product_title%>&product_
    var product_id=req.query.product_id;
    var product_title=req.query.product_title;
    var product_condition=req.query.product_condition;
    var category=req.query.category;
    var subcategory=req.query.subcategory;
    var available_quantity=req.query.available_quantity;
    var product_description=req.query.product_description;
    var instant_sellingprice=req.query.instant_sellingprice;
    var listing_type=req.query.listing_type;

    var isUpdate = req.query.isUpdate;

    if(isUpdate === "yes")
    {
        res.render('adminUpdateProduct', { title: 'Admin/UpdateProduct', userName : globals.usrFName, product_id:product_id });
    }
    else
    {
        res.render('productDetail', { title: 'ProductDetail/eBay', userName : globals.usrFName, product_id:product_id });
    }
};
```

Fetch top products

```
//Fetch top 20 products..
function fetchtopfewProduct(req, res) {
    var gettopfewproducts = " select product_id,product_title,instant_Sellingprice,listing_type from product limit 20";

    console.log("Query is:" + gettopfewproducts);
    var resultSet = "";
    mysqlpool.execQuery(gettopfewproducts, "", function(err, results) {
        if (err) {
            console.log("Error while fetching buyers from db");
            throw err;
        } else {
            if (results.length > 0) {
                console.log("length" + results.length);
                resultSet = JSON.parse(JSON.stringify(results));
                console.log(resultSet);
                res.render('product', {
                    title : 'Products',
                    userName : globals.usrName,
                    result : resultSet
                });
            }
        }
    });
    else // no results
    {
        var resultSet = JSON.stringify("No results found");
        console.log(resultSet);
        res.render('product', {
            title : 'Products',
            error : "No results found",
            result : resultSet
        });
    }
}
};//fetchtopfewProduct
```

Get Product Details

```

function getProductdetails(req, res) {
    var gettopfewproducts = " select product_id,product_title,instant_Sellingprice,listing_type from product limit 20";

    console.log("Query is:" + gettopfewproducts);
    var resultSet = "";
    mysqlpool.execQuery(gettopfewproducts, "", function(err, results) {
        if (err) {
            console.log("Error while fetching buyers from db");
            throw err;
        } else {

            if (results.length > 0) {
                console.log("length" + results.length);
                resultSet = JSON.parse(JSON.stringify(results));
                console.log(resultSet);
                res.render('product', {
                    title : 'Products',
                    userName : globals.userName,
                    result : resultSet
                });
            }
            //if
        } // no results
    } {
        var resultSet = JSON.stringify("No results found");
        console.log(resultSet);
        res.render('product', {
            title : 'Products',
            error : "No results found",
            result : resultSet
        });
    }
    //else
});
}

} //getProductdetails

```

List Auctionable Products

```

function listauctioableProduct(req, res) {
    var getauctionableproducts = " select product_id,product_title,instant_Sellingprice from product where listing_type='auction'"

    console.log("Query is:" + getauctionableproducts);
    var resultSet = "";
    mysqlpool.execQuery(getauctionableproducts, "", function(err, results) {
        if (err) {
            console.log("Error while fetching buyers from db");
            throw err;
        } else {

            if (results.length > 0) {
                console.log("length" + results.length);
                resultSet = JSON.parse(JSON.stringify(results));
                console.log(resultSet);
                res.render('product', {
                    title : 'Products',
                    userName : globals.userName,
                    result : resultSet
                });
            }
            //if
        } // no results
    } {
        var resultSet = JSON.stringify("No results found");
        console.log(resultSet);
        res.render('product', {
            title : 'Products',
            error : "No results found",
            result : resultSet
        });
    }
    //else
});
}

} //listauctioableProduct

```

Search Products

```
var searchproduct = squel.select().from("product_view")

if (search !== '' && search !== undefined)
    searchproduct.where("product_title like '%" + search + "%'")
if (category !== '' && category !== undefined)
    searchproduct.where("category='" + category + "'")
if (subcategory !== '' && subcategory !== undefined)
    searchproduct.where("subcategory='" + subcategory + "'")
if (condition !== '' && condition !== undefined)
    searchproduct.where("product_condition='" + condition + "'")
if (lowestprice !== '' && lowestprice !== undefined)
    searchproduct.where("instant_sellingprice>=" + lowestprice)
if (highestprice !== '' && highestprice !== undefined)
    searchproduct.where("instant_sellingprice<=" + highestprice)
if (listing_type !== '' && listing_type !== undefined)
    searchproduct.where("listing_type='" + listing_type + "'")

exports.userLogin = function (req,res){
    var user = req.param("inputEmail");
    var pwd = req.param("inputPwd");
    var cols = [];
    if(user !== null){
        globals.queryString = 'SELECT * FROM ?? WHERE ?? = ?';
        console.log("Query is : " + globals.queryString);

        // Connection Poling Call
        mysqlpool.execQuery(globals.queryString,['person','emailaddr',user],function(err,results){
            if(err){
                throw err;
            } else {
                if(results.length > 0){
                    var resultStr = JSON.parse(JSON.stringify(results));
                    if(resultStr[0].password === pwd){
                        globals.Auth = true;
                        globals.usrFName = resultStr[0].firstname;
                        globals.usrLName = resultStr[0].lastname;
                        globals.address = resultStr[0].address;
                        globals.city = resultStr[0].city;
                        globals.state = resultStr[0].state;
                        globals.country = resultStr[0].country;
                        globals.zipcode = resultStr[0].zipcode;
                        globals.email = resultStr[0].emailaddr;
                        globals.lastLogin = resultStr[0].lastlogin;
                        globals.user_Id = resultStr[0].memberid;
                        var admin = resultStr[0].admin;
                        if(admin == 'no')
                            res.render('userHome', { title: 'DashBoard/eBay', userName : globals.usrFName, result : resultStr });
                        else
                            res.render('admin', { title: 'DashBoard/eBay', userName : globals.usrFName, result : resultStr });
                    }
                }
            }
        })
    }
}
```

Sell

```
//(parameters needed from front end)
//sellproduct(producttitle, condition, category, subcategory, quantity, description, sellingprice, listing_type, memberid, startdate,
//enddate, startingprice

// conditions to check in front end - starting and end date should be in the format 'yyyy-mm-dd hh:mm:ss' and (end date > start date)

function sellProduct(req, res) {
    // add an entry in Product table
    //initially initial_qauntity and available_quantity will be same

    //var d = new Date();
    //var product_id=d.getTime();

    var createproduct="insert into product (product_id,product_title,product_condition,category,subcategory,available_quantity," +
        "initial_quantity,product_description,instant_sellingprice,listing_type) values ("+product_id + ",""+req.param("producttitle")+
        ",""+req.param("condition")+"','"+req.param("category")+"','"+req.param("subcategory")+"','"+req.param("quantity") +
        ",""+req.param("quantity")+"','"+req.param("description")+"','"+req.param("sellingprice")+"','"+req.param("listing_type")+"')";

    console.log("Query is:"+createproduct);

    mysqlpool.execQuery(createproduct,'',function(err,results){
        if(err){
            console.log("Error while inserting into db");
            throw err;
        }
    });

    // add an entry in seller table

    var createSeller = "insert into seller (sellerid,product_id,selling_time, product_title,sellerfirstname, sellerlastname) values(" + globals.user
    console.log("Query is:" + createSeller);

    mysqlpool.execQuery(createSeller,'',function(err,results){
        if(err){
            console.log("Error while inserting into db");
            throw err;
        }
    });

    // if listing type is auction then add an entry in auction table
    //initially starting price and current price will be same
    console.log(req.param("listing_type").toLowerCase());
    if(req.param("listing_type") === "auction"){
        var dt = new Date();
        var auction_id=dt.getTime();

        var createAuction = "insert into auction (seller_id,product_id,auction_starting_date,auction_end_date,auction_starting_price,current_price) =
            req.param("startdate")+"','"+req.param("enddate")+"','"+req.param("sellingprice")+"','"+req.param("sellingprice")+"')";

        console.log("Query is:" + createAuction);

        mysqlpool.execQuery(createAuction,'',function(err,results){
            if(err){
                console.log("Error while inserting into db");
                throw err;
            }
        });

        } //if auction
        res.render('sellingHistory', { title: 'eBay',userName : globals.usrFName});
    }

    // sell product

    exports.sellProduct = sellProduct;

    exports.index = function(req,res){
        res.render('sell', {title : 'sell/eBay'});
    };
}
```

DB Connection Pooling

```
var mysql_pool = require('mysql');
var pool = mysql_pool.createPool({
  host      : 'localhost',
  user      : 'root',
  password  : 'Neeru',
  port      : '3306',
  database  : 'ebay',
  connectionLimit : '10'
});

exports.pool = pool;

/**
 * New node file
 * To take care of obtaining DB connection from the connection pool and executing queries.
 * Also logs SQL exceptions, results and releases the DB connection to the pool.
 */
function execQuery (sql, params, callback) {
  var connPool = require('./mysqlConn').pool;
  connPool.getConnection(function (err, connection) {
    if (err) {
      console.log('MySQL connection error: ' + err);
      callback(err, true);
      return;
    }
    console.log("Query is >>>>" +sql);
    var qResult = connection.query(sql, params, callback);
    qResult.on('error', function(err) {
      console.log('MySQL query error: ' + err);
      callback(err, true);
    });
    qResult.on('result', function(rows) {
      console.log('Got result from DB');
      callback(false, rows);
    });
    qResult.on('end', function() {
      console.log('Going to release DB connection to the Pool');
      connection.release();
    });
  });
}

exports.execQuery=execQuery;
```

Pagination and Caching

Check Available List

```
exports.checkifListAvailable = function (req,res,listname,isPresentCallback, loadCacheCallback,currentpage) {
    client.on("error", function (err) {
        console.log("Error in Redis client " + err);
    });

    var length = 0;

    client.llen(listname,function(err,data) {
        length=data;
        console.log(" length : "+length);
        if(length > 0) {
            console.log("page no : " + currentpage)
            isPresentCallback(res,length,currentpage,listname);
        }else {
            loadCacheCallback(req,res);
        }
    });
};
```

Delete List

```
exports.deletelist = function (name) {

    client.on("error", function (err) {
        console.log("Error in Redis client " + err);
    });

    console.log("deleting list ");
    client.del(name);

};
```

Set Value

```
exports.setval = function (name,results) {
    client.on("error", function (err) {
        console.log("Error in Redis client " + err);
    });

    for ( i=0 ; i<results.length; i++)
        client.rpush(name, JSON.stringify(results[i]));

};
```

Get Value

```

if(name == "listallProduct")
{
    console.log("In adminProduct");

    res.render('adminProduct', { title: 'Admin/Products', error : "", result : resultSet, userName:'Admin', currentPage:currentPage});
}else if(name == "listallperson")
{
    console.log("In adminPerson");
    res.render('adminPerson', { title: 'Admin/Person', error : "", result : resultSet,userName:'Admin',currentPage:currentPage});
}else if(name == "listallsellers")
{
    console.log("In adminSeller");
    res.render('adminSeller', { title: 'Admin/Seller', error : "", result : resultSet, userName:'Admin',currentPage:currentPage});
}else if(name == "listallauctions")
{
    console.log("In adminAuction");
    res.render('adminAuctions', { title: 'Admin/Auctions', error : "", result : resultSet, userName:'Admin',currentPage:currentPage});
}

}//if
else // no results
{
    var resultSet=JSON.stringify("No results found");
    console.log(resultSet);
    if(name == "listallProduct")
    {
        console.log("In adminProduct");
        res.render('adminProduct', { title : 'Admin/Products', error : "No Results found!",result: resultSet,userName:'Admin',curr
    }else if(name == "listallperson")
    {
        console.log("In adminPerson");
        res.render('adminPerson', { title : 'Admin/Person', error : "No Results found!",result: resultSet,userName:'Admin',current
    }else if(name == "listallsellers")
    {
        console.log("In adminSeller");
        res.render('adminSeller', { title : 'Admin/Seller', error : "No Results found!",result: resultSet, userName:'Admin',curren
    }else if(name == "listallauctions")
    {
        console.log("In adminAuction");
        res.render('adminAuctions', { title : 'Admin/Auctions', error : "No Results found!",result: resultSet, userName:'Admin',cu

exports.getval = function (res,name,startindex,endindex,currentPage) {

    client.on("error", function (err) {
        console.log("Error in Redis client " + err);
    });

    client.lrange(name, startindex, endindex, function (error, results) {
        if (error)
        {
            console.log("Er : " + error);
            throw error;
        }else
        {
            console.log('Name:' + name);
            if(results.length > 0){
                console.log("length "+results.length);
                var resultSet = results;//JSON.parse(JSON.stringify(results));
                resultSet = [];
                for(i=0;i<results.length;i++) {
                    resultSet.push(JSON.parse(results[i]));
                }
            }
        }
    });
}

```

List Available

```
exports.listavailable = function (res,length,currentPage,listname)
{
    //var listLength = length
    var startindex=0;

    var pages = 0
    var lastPageEntries = length % numEntries;
    if(length < numEntries )      // length < num of entries per page , only 1 page
    {
        console.log("only 1 page");
        pages = 1;
        endindex=length;
    }
    else {

        //to calculate num of pages
        if(lastPageEntries > 0 )           // last page entry >0 but less than 10.
            pages = length/numEntries + 1;
        else                                // page contains exactly 10 items.
            pages = length/numEntries;

        // to caculate start and end index
        startindex=((currentPage-1)*(numEntries));
        if(currentPage==pages) // last page
        {
            console.log(" last page ");
            endindex= lastPageEntries;
        }
        else{
            console.log(" each page ");
            endindex= startindex + (numEntries -1 ) ;
        }
    }

    console.log("start index : "+startindex+" end index : "+endindex);
    exports.getval(res,listname,startindex,endindex,currentPage);
}
```

Data insertion Scripts

```
/**  
 * New node file  
 */  
var eis = require("eis");  
  
var mysqlpool = require('./dbConnection/mysqlQuery');  
  
function mysql_real_escape_string (str) {  
    return str.replace(/[\0\x08\x09\x1a\n\r"\\%]/g, function (char) {  
        switch (char) {  
            case "\0":  
                return "\\0";  
            case "\x08":  
                return "\\b";  
            case "\x09":  
                return "\\t";  
            case "\x1a":  
                return "\\z";  
            case "\n":  
                return "\\n";  
            case "\r":  
                return "\\r";  
            case "\":  
            case "\'":  
            case "\\":  
            case "%":  
                return "\\"+char; // prepends a backslash to backslash, percent,  
                                // and double/single quotes  
        }  
    });  
}  
  
function insertdata() {  
    var productdata = "select * from product";  
    console.log("Query is:"+productdata);  
    mysqlpool.execQuery(productdata,'',function(err,prodresults){  
        if(err){  
            console.log("Error while fetching from db");  
            throw err;  
        }  
        else {  
            console.log("prodresults length"+prodresults.length);  
            var persondata= " select * from person";  
            mysqlpool.execQuery(persondata,'',function(err,personresults){  
                if(err){  
                    console.log("Error while fetching from db");  
                    throw err;  
                }  
                else {  
                    var j=0;  
                    for(var i=0;i<prodresults.length;i++)  
  
                        var insertseller="insert into seller  
(product_id,sellerid,selling_time,sellerfirstname,sellerlastname,product_title) values (" +  
                        prodresults[i].product_id+" ,"+personresults[j].member_id+" ,now()," +personresults[j].firstname  
+" ,"+  
                        personresults[j].lastname+" ,"+mysql_real_escape_string(prodresults[i].product_title)+"")";  
                        console.log("Query is:"+insertseller);  
  
                        mysqlpool.execQuery(insertseller,'',function(err,sellerres){  
                            if(err){  
                                console.log("Error while inserting into db");  
                                throw err;  
                            }  
                        });  
                    if(prodresults[i].listing_type=='auction')  
                }  
            })  
        }  
    })  
}
```

```
if(prodresults[i].listing_type=='auction')
{
    var createAuction = "insert into auction
seller_id,product_id,auction_starting_date,auction_end_date,auction_starting_price,current_price) values ('+
personresults[j].memberid','"+prodresults[i].product_id+", now(),'2014-12-9
':30:21',' +prodresults[i].instant_sellingprice','+prodresults[i].instant_sellingprice')";
    console.log("Query is:" + createAuction);
    mysqlpool.executeQuery(createAuction,'',function(err,results){
        if(err){
            console.log("Error while inserting into db");
            throw err;
        }
    });
}

j++;
if(j==personresults.length) j=0;
}//for product
}//else person
});//product
});//product

//bidproduct

exports.insertdata = insertdata;
```