(DHANKAWADI, PUNE -43)



PROJECT REPORT ON

“PAYBACK CARD MANAGEMENT SYSTEM”

SUBMITTED BY

Siddhesh Malegaonkar (3941)

Ridhim Rastogi (3958)

Nitesh Sarode (3960)

(Third Year Information Technology)

UNDER THE GUIDANCE OF

Prof. S. A. Jakhate

Submitted To

**Pune Institute of Computer Technology**

**For Academic Year 2015-16**

|  |  |
| --- | --- |
| **TITLE** | **PAGENO** |
| **1. Introduction** | **3** |
| **2. Scope and limitations** | **3** |
| **3.Flow Diagram** | **4** |
| **4. Schema Diagram** | **5** |
| **5. coding -Procedure and Mapreduce (description)** | **9** |
| **6. coding design(snapshot)** | **10** |
| **7.User Interface Snapshots** | **13** |
| **8.Conclusion** | **21** |

# INTRODUCTION

Each one of us shops on a daily basis. It may include things such as food, drinks, clothing that are essential for our survival to luxuries such as mobile, laptops, washing machine. Our payback card system focuses on providing an opportunity to our customers where their every purchase can benefit them.

Through the payback card system we give our customers an opportunity to earn points with every purchase which they make at one of our partners. These points can then be redeemed in form of exciting discount or gifts.

# SCOPE

1. The software can be used for generating revenue through sale of products offered by it's affiliated partners.
2. Software is made for centralized information maintenance of Payback card system.
3. Central administrator has the control of all the Partners and the Offers, w.r.t. adding ,deleting, modifying and analysing database information about them.
4. Each Partner and User has their own login. Products on the website are controlled by the Partners.

# FEATURES

1. Software is used to keep a track of purchases made by our customers in order to provide them with valuable feedback and suggestions using relational automata.
2. Upon completion of each transaction a mail is sent to the User's email address confirming their address.

# LIMITATIONS

1. High security risk as the data is insecure.
2. The design of the project is not reusable.

# FLOW DIAGRAM

|  |  |
| --- | --- |
| Edit Customer Information  (Customer can edit, update and delete his account)   |  | | --- | | LOGIN |   New Customer    Post Home  (Partner Pages  Displaying their products)  Home Page  Partner as Admin  Displays Cart  (Displays Information of Products in the cart)  Admin  (Add or Remove Partner)  Confirmation Page |

# SCHEMA DIAGRAM

Payback

Partner

Name

Partner ID

Product Group

User

Card Number

Name

Phone Number

Address

Password

Redeem

Points

Gifts

User

Card Number

Shop History

Group Name

Validity

Offers

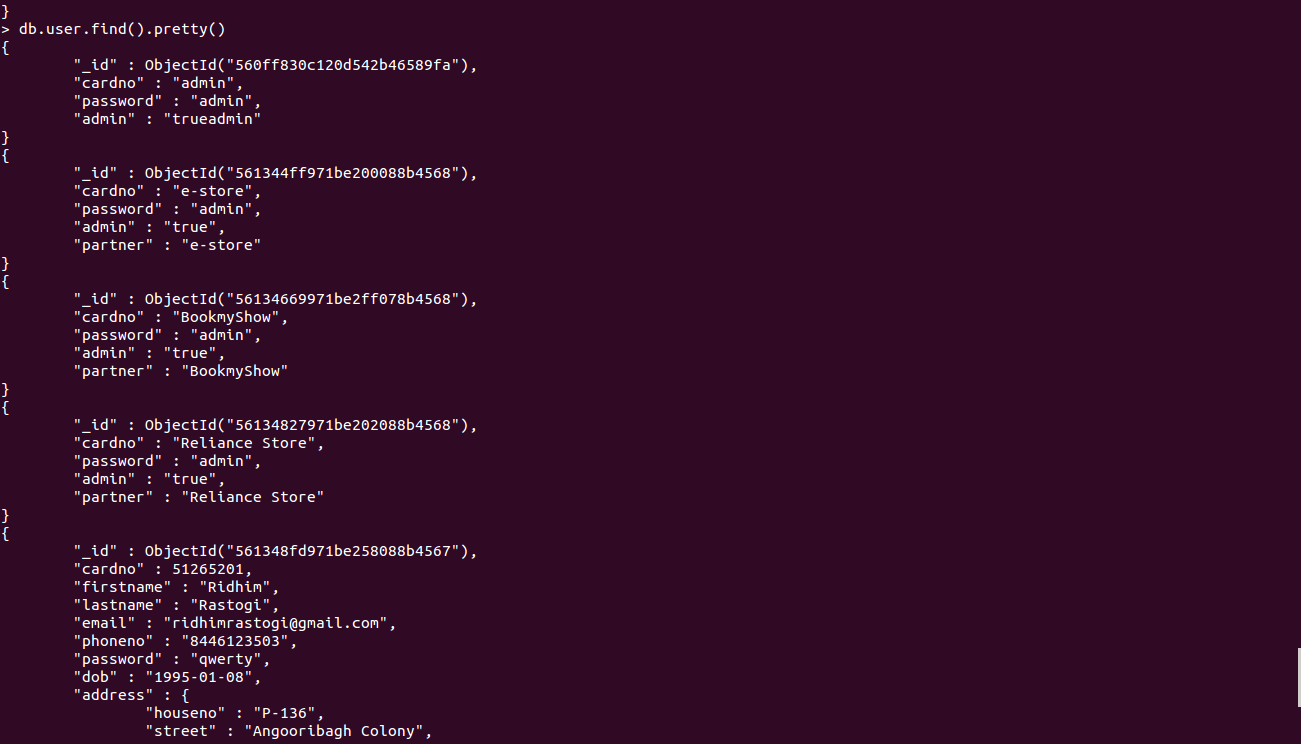
# Collection List

There are three collections in the database. They are as follows:

1.User- This collection contains information about the customers as well as login details for the partners.

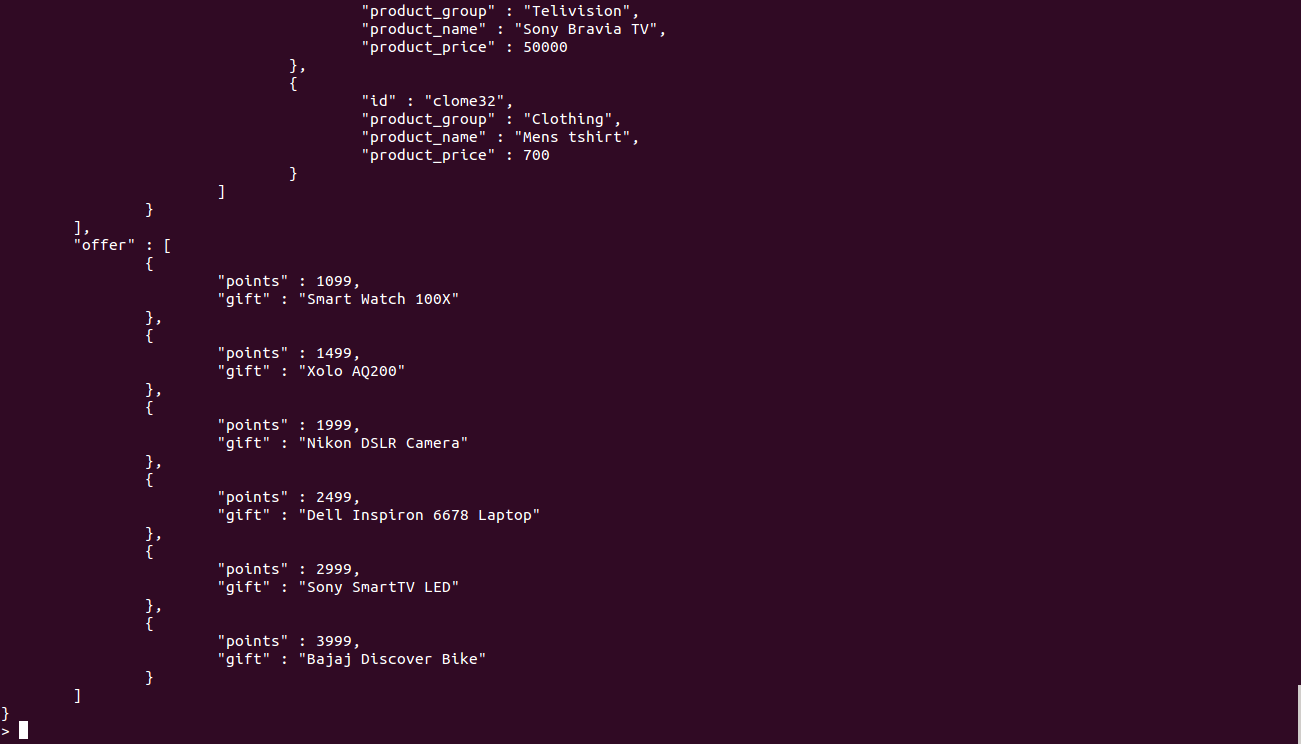
2.Partner-This collection contains information about the products that each partner has made available for selling.

3.Redeem-This collection contains shopping history of each user and details about the offers on our website.









# CODING

# Procedures Used:

# Sample format:

1.Stored Procedure used for calculating the no. of items shopped by each user.

\_id: “recent”

value: function(x){return x+1;}

EXECUTION:

$toexec='function(x) { return recent(x) }';

$args=array($response[retval]);

$response=$db->execute ($toexec, $args);

# Mapreduce

1.Map reduce is used to calculate the total amount spent by each customer on our

website

$map=new MongoCode("function(){for(var idx=0;idx<this.user.length;idx++){var key=this.user[idx].cardno;

for(var id=0;id<this.user[idx].shop\_history.length;id++){

var value=this.user[idx].shop\_history[id].product\_price;emit(key,value);}}}");

$reduce=new MongoCode("function(keys,values){var sum=0;

for(var id=0;id<values.length;id++){sum+=values[id];}return sum;}");

$total=$db-command(array("mapreduce"=>"redeem","map"=>$map,"reduce"=>$reduce,"out"=>"mr"));

2.Map reduce is used to count the instances of a product being brought by a user. The count is displayed using a dynamic graph.

$map=new MongoCode("function(){for(var idx=0;idx<this.user.length;idx++){

for(var id=0;id<this.user[idx].shop\_history.length;id++){

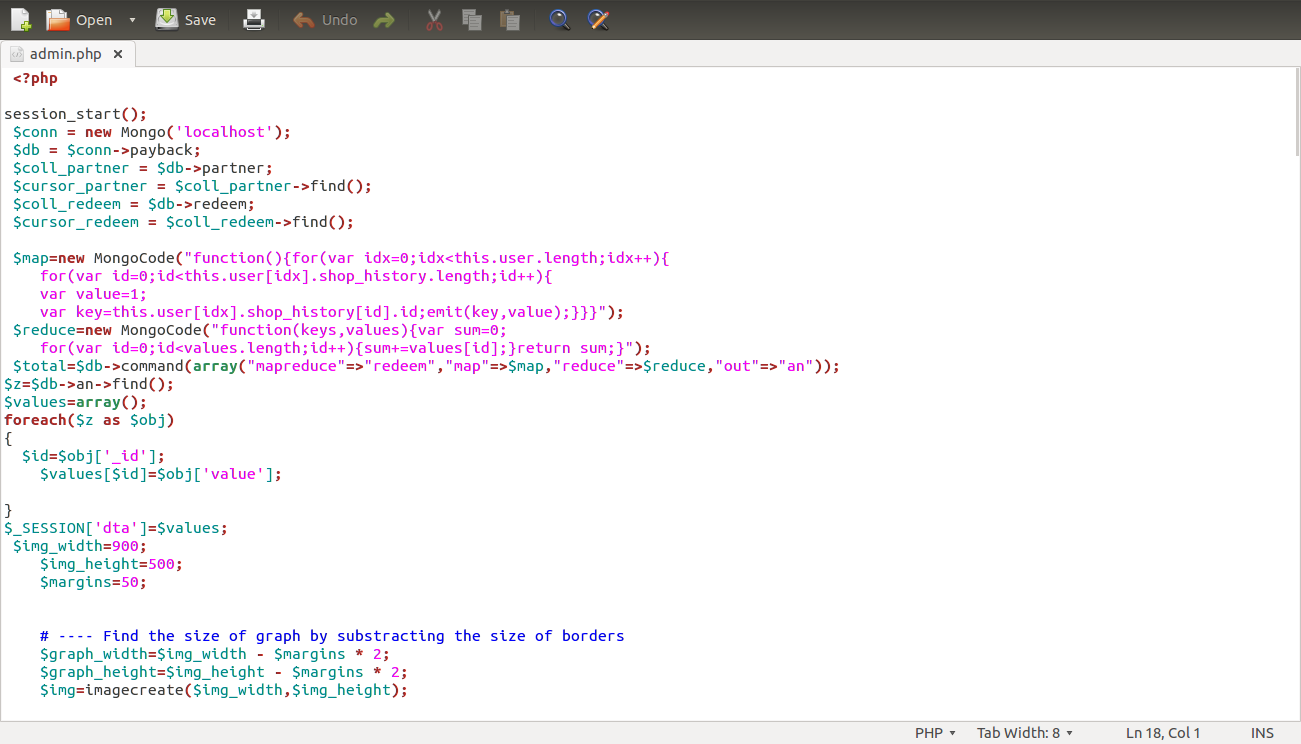
var value=1;

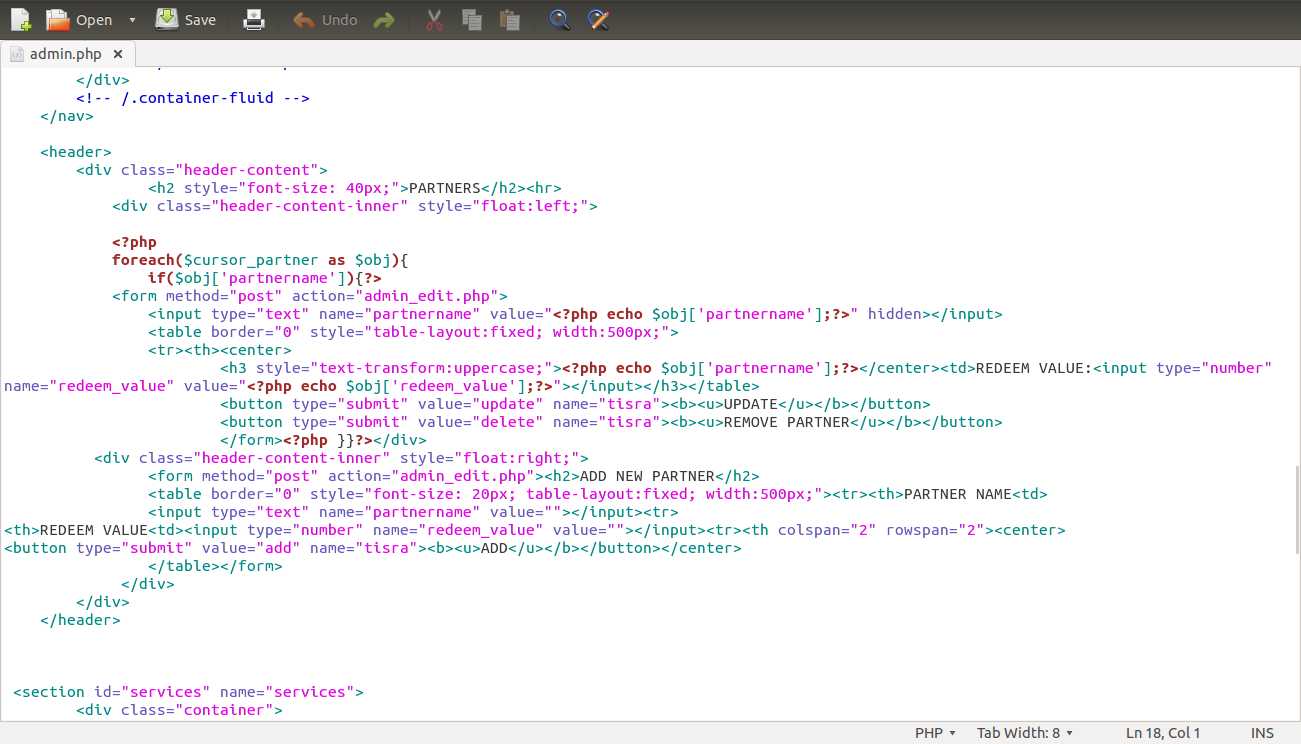
var key=this.user[idx].shop\_history[id].id;emit(key,value);}}}");

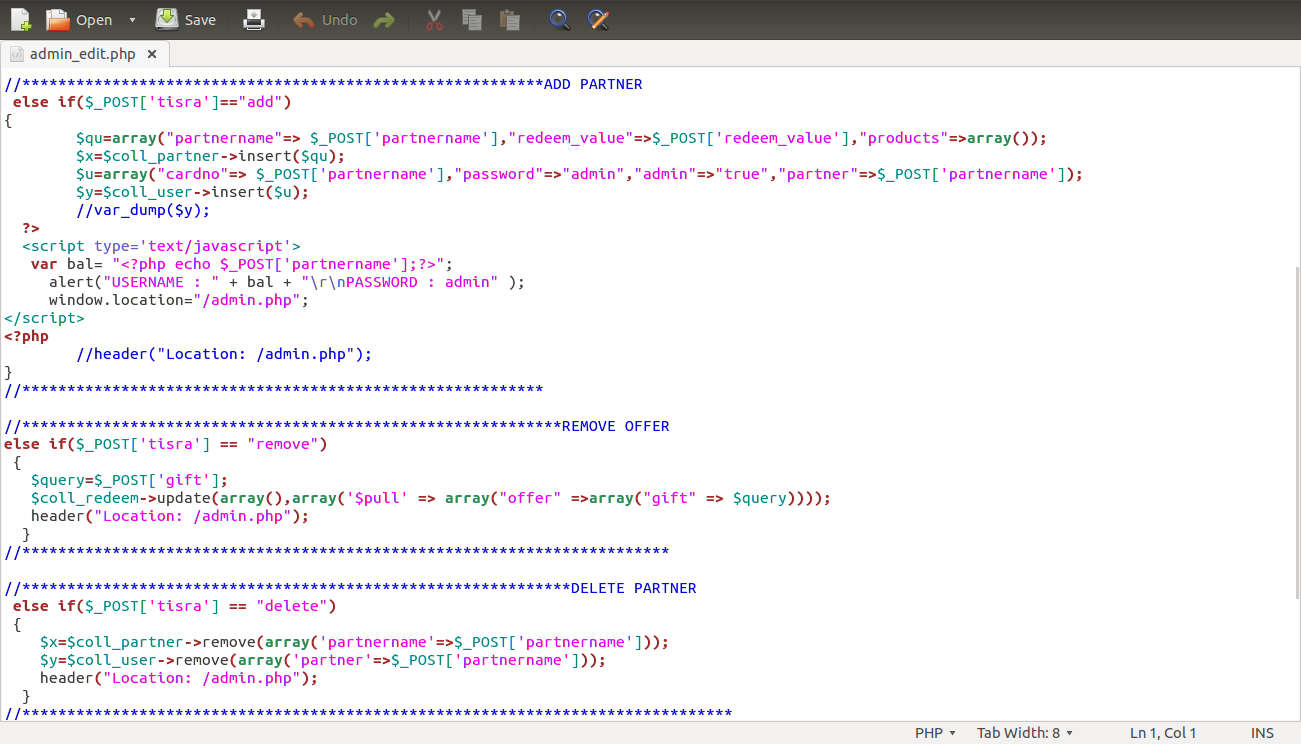
$reduce=new MongoCode("function(keys,values){var sum=0;

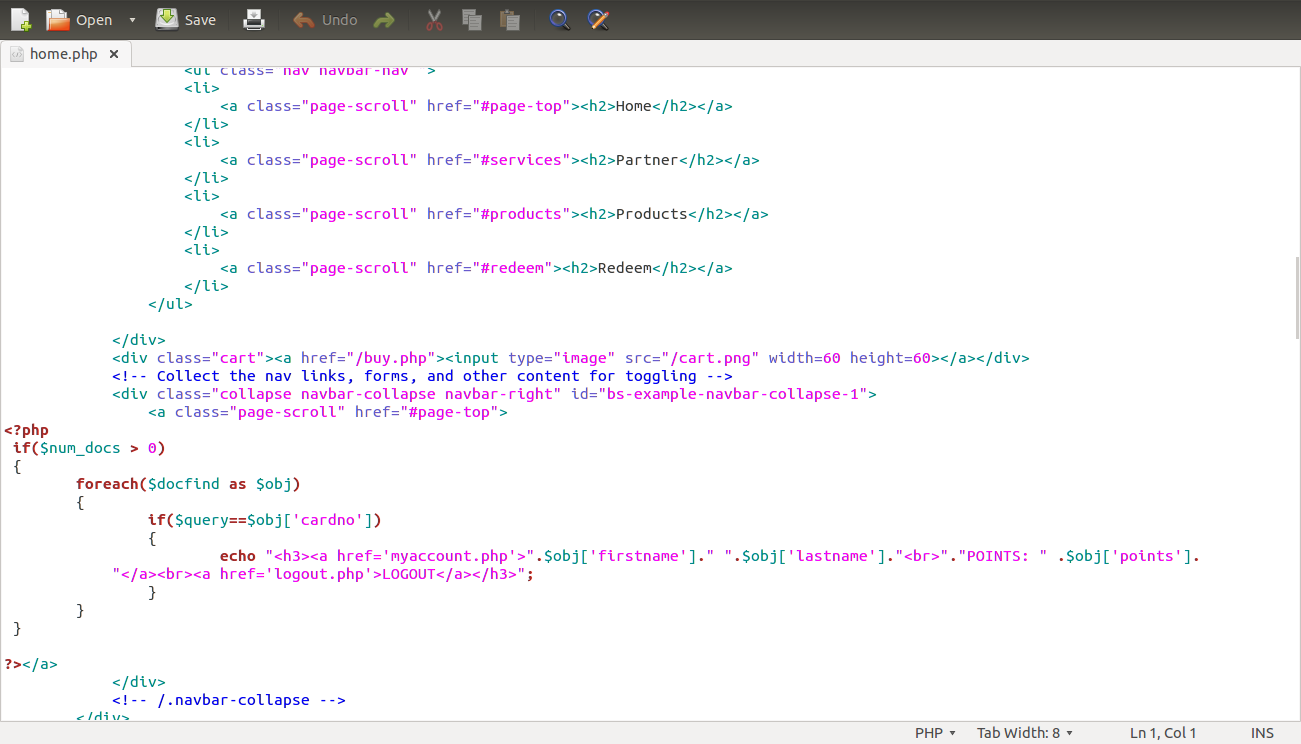
for(var id=0;id<values.length;id++){sum+=values[id];}return sum;}");

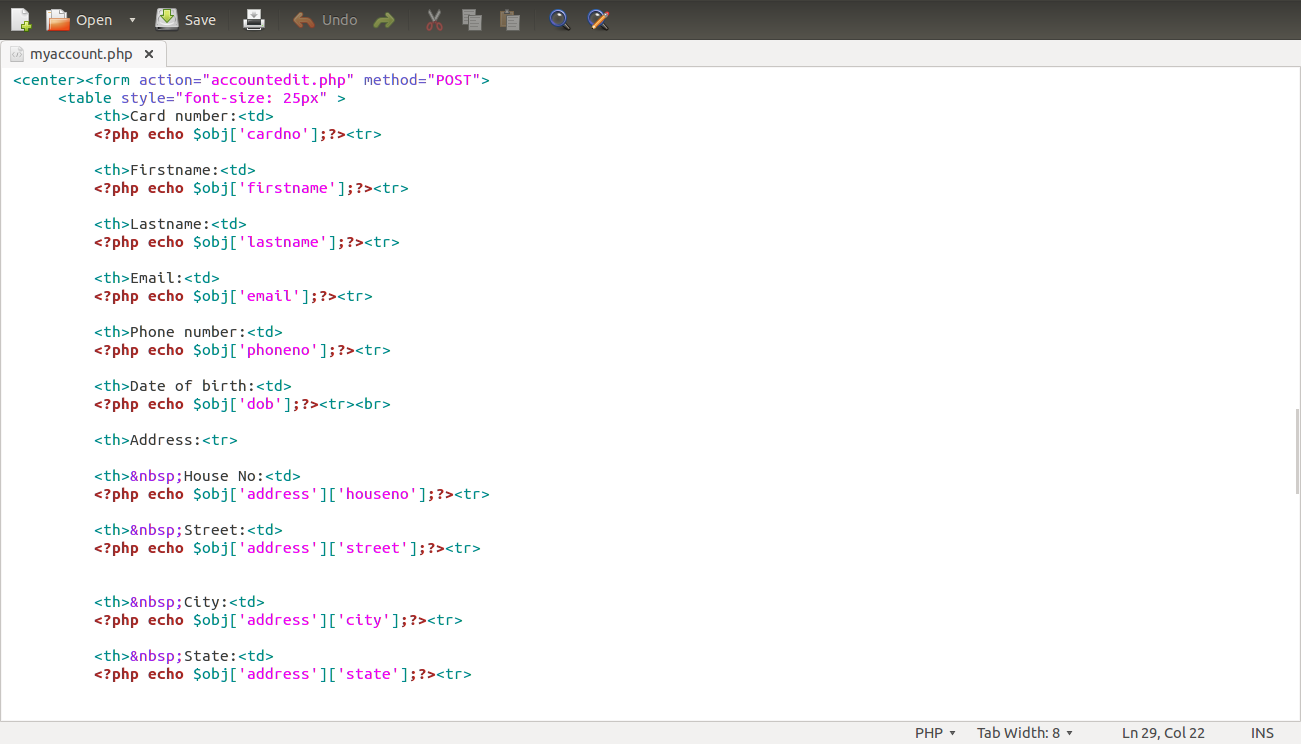
$db-command(array("mapreduce"=>"redeem","map"=>$map,"reduce"=>$reduce,"out"=>"an"));

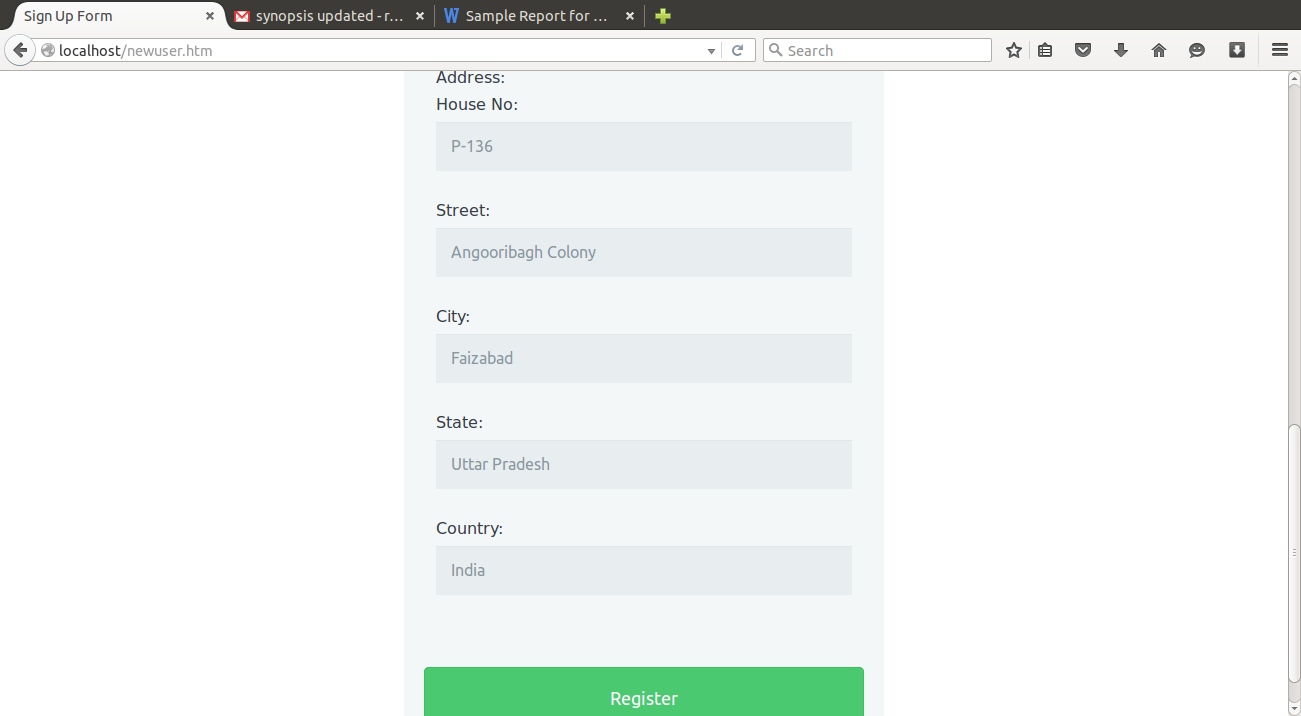
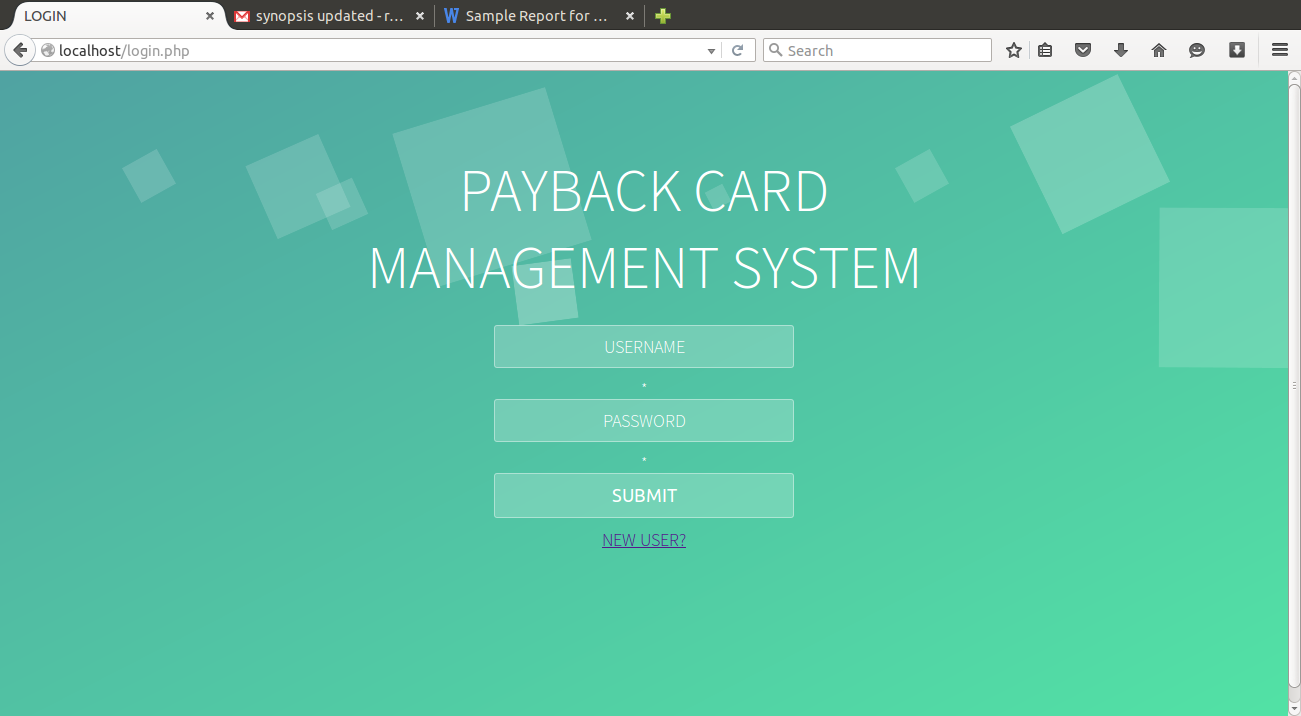


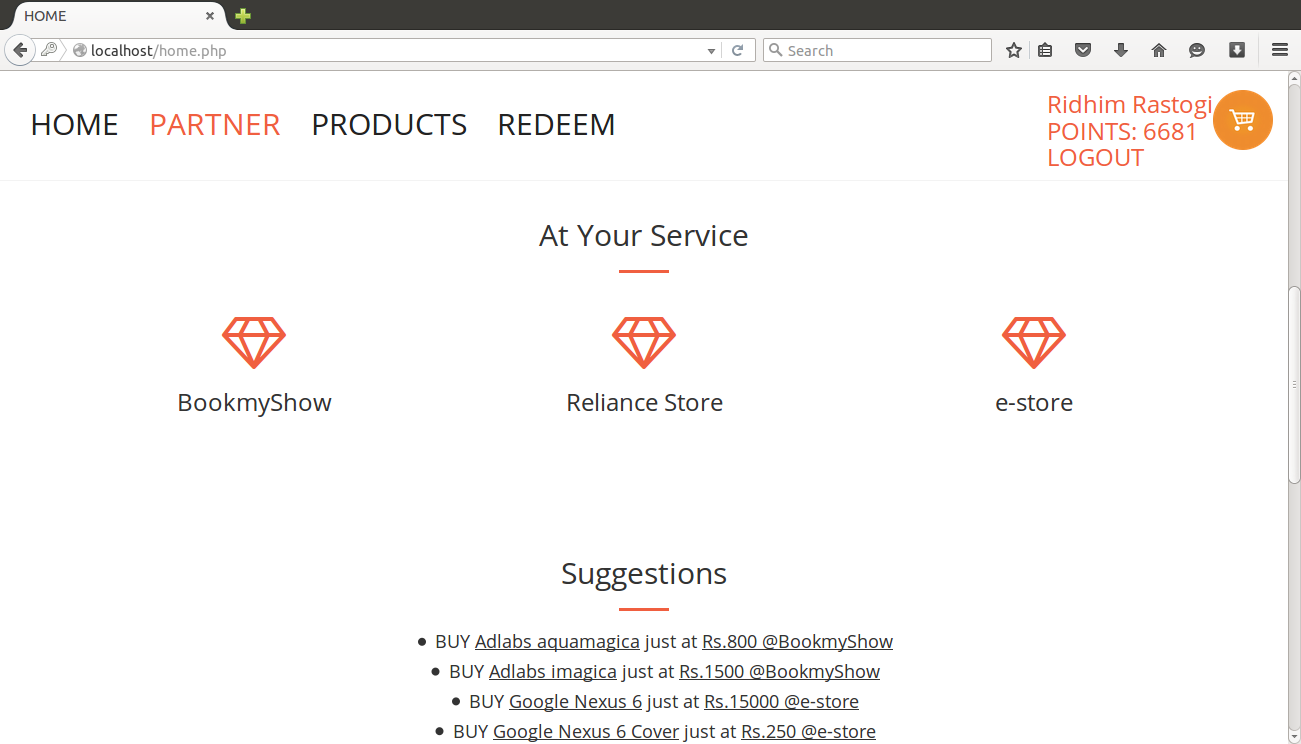


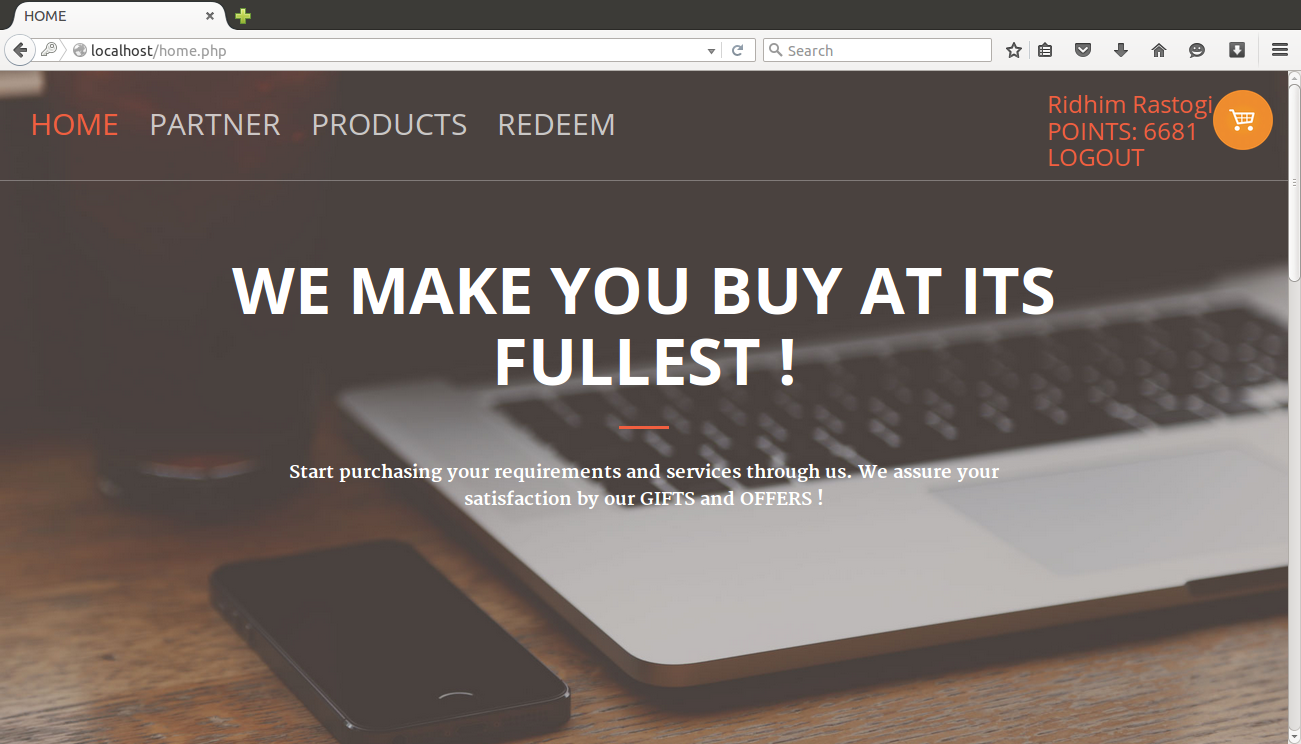


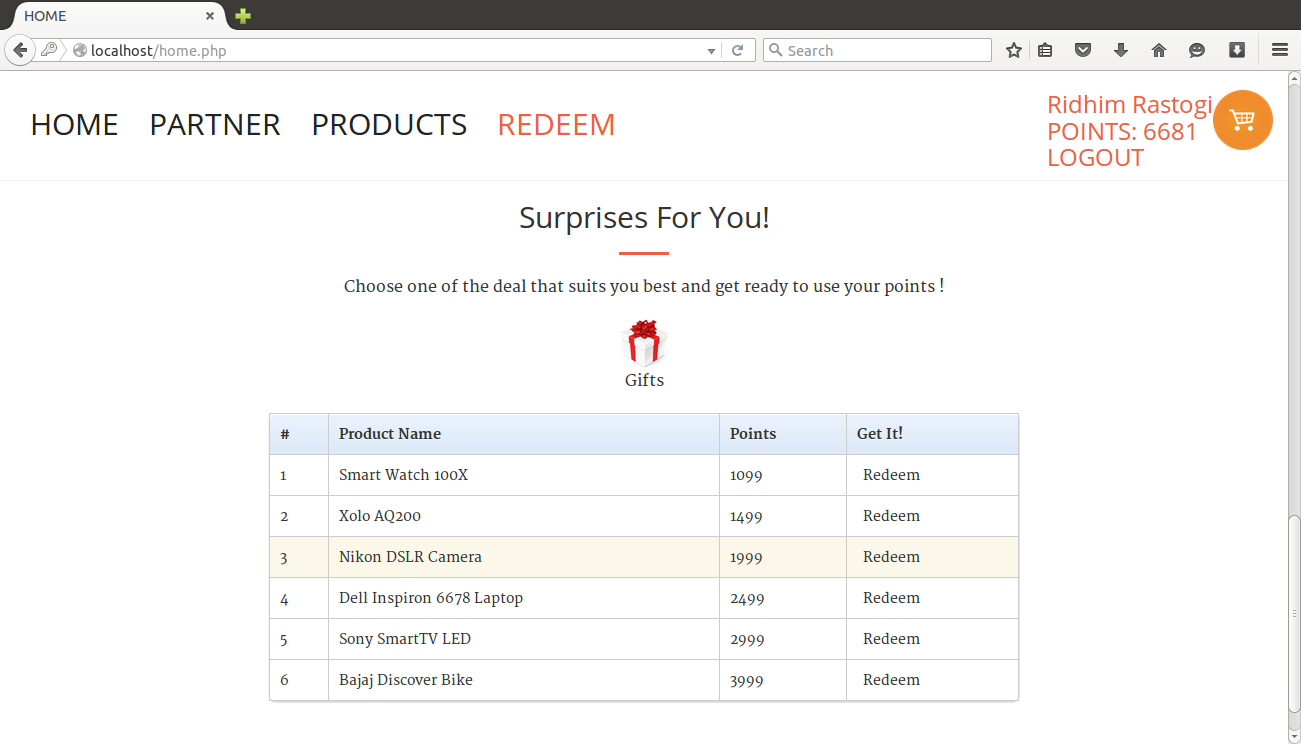
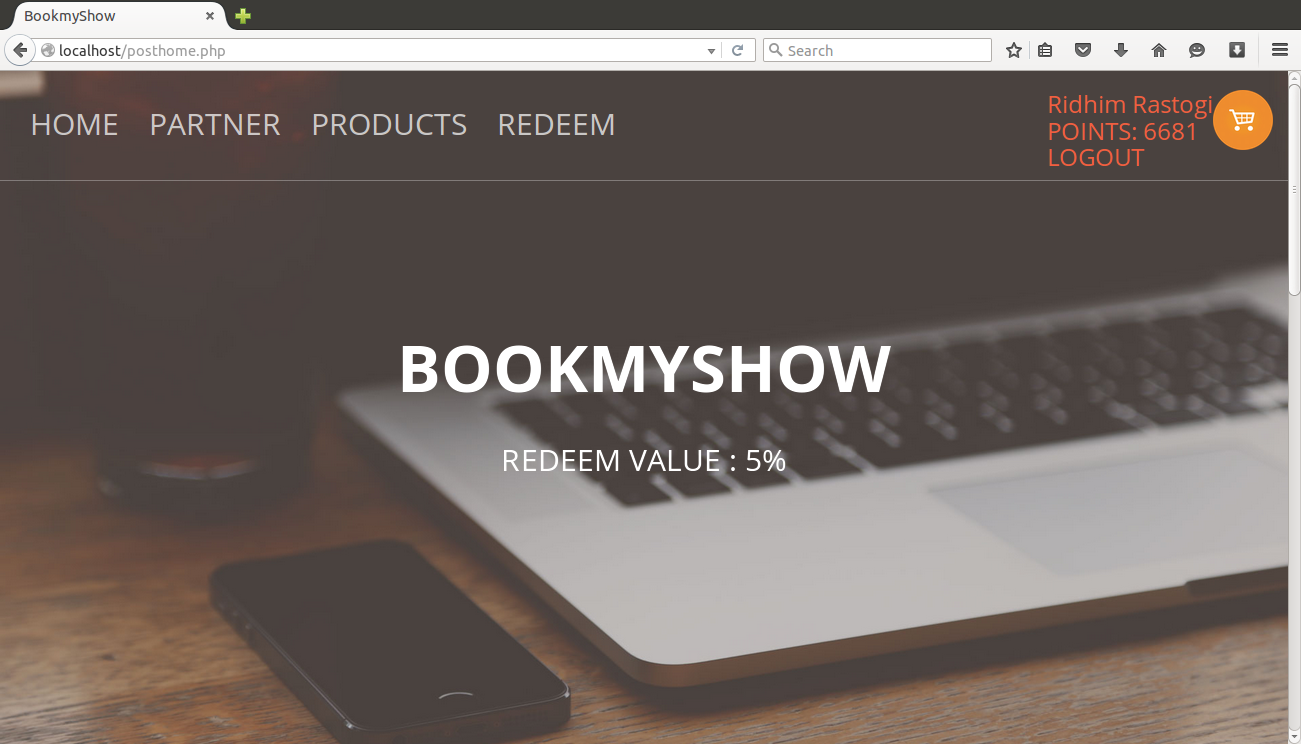


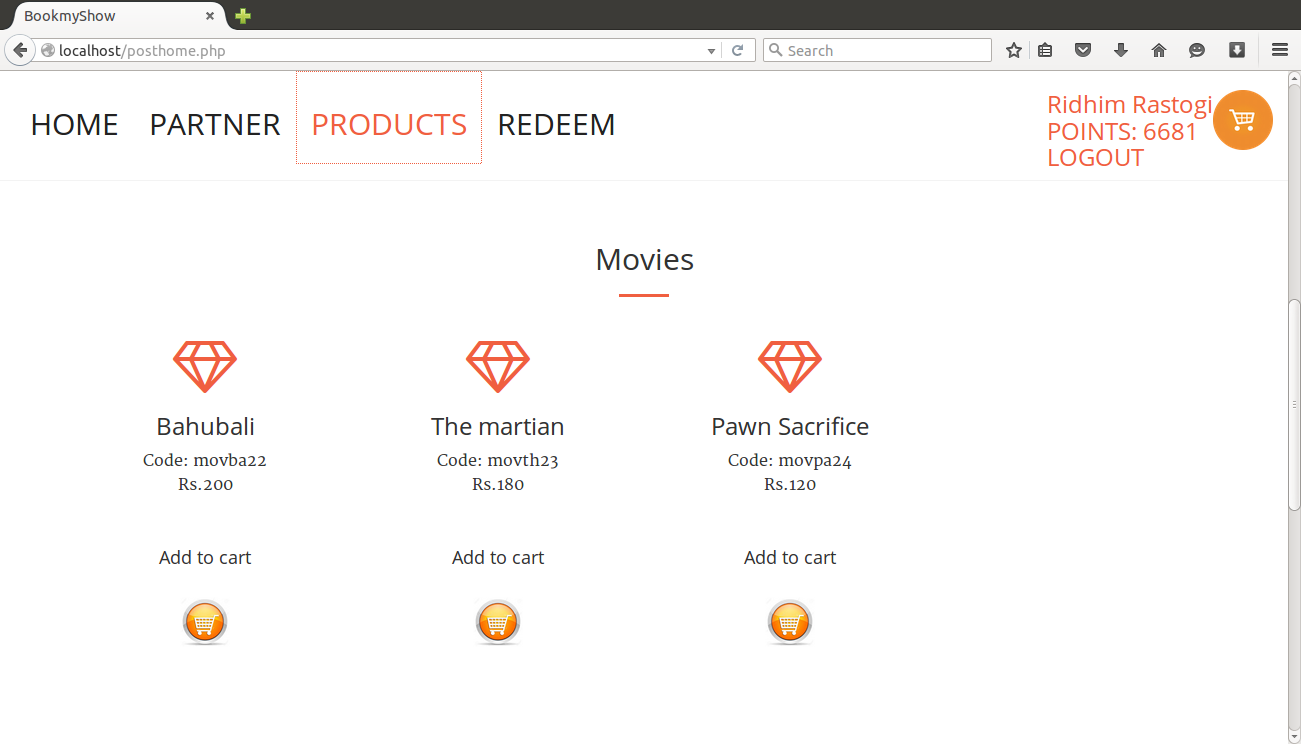
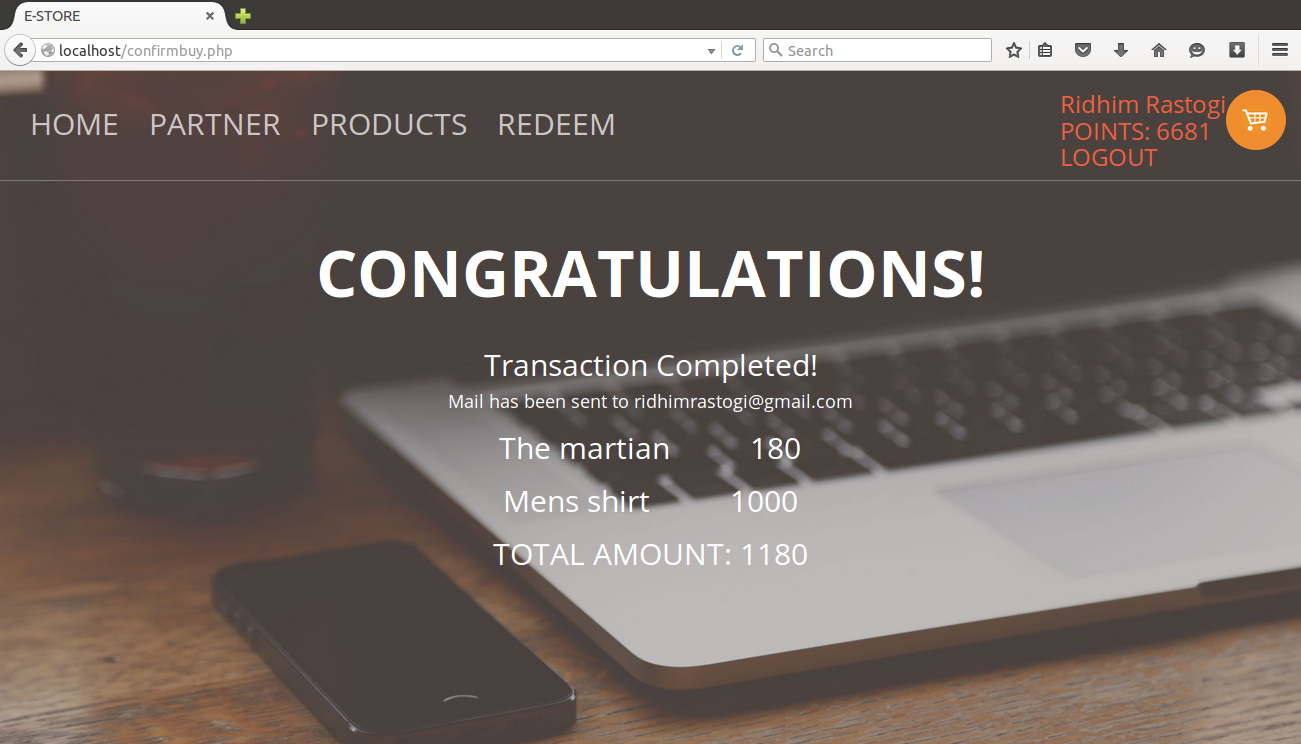


****USER INTERFACE SNAPSHOTS:

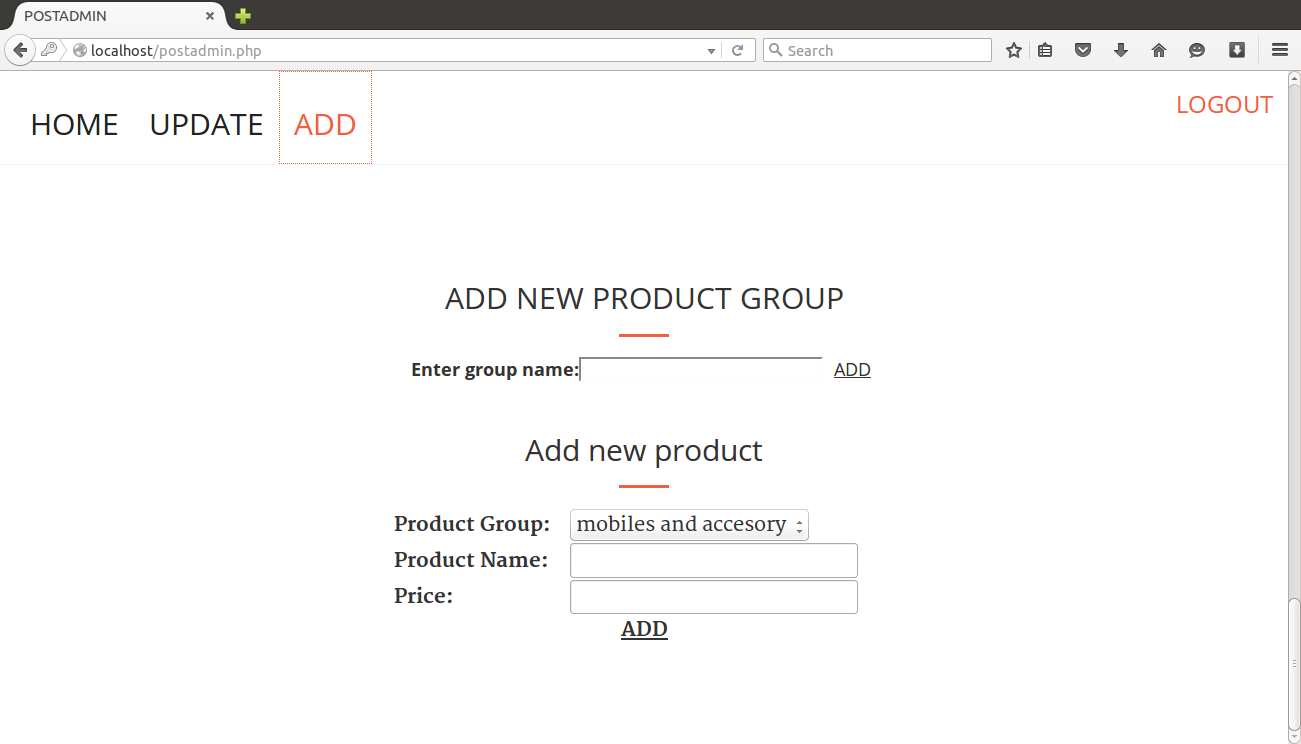
****

****

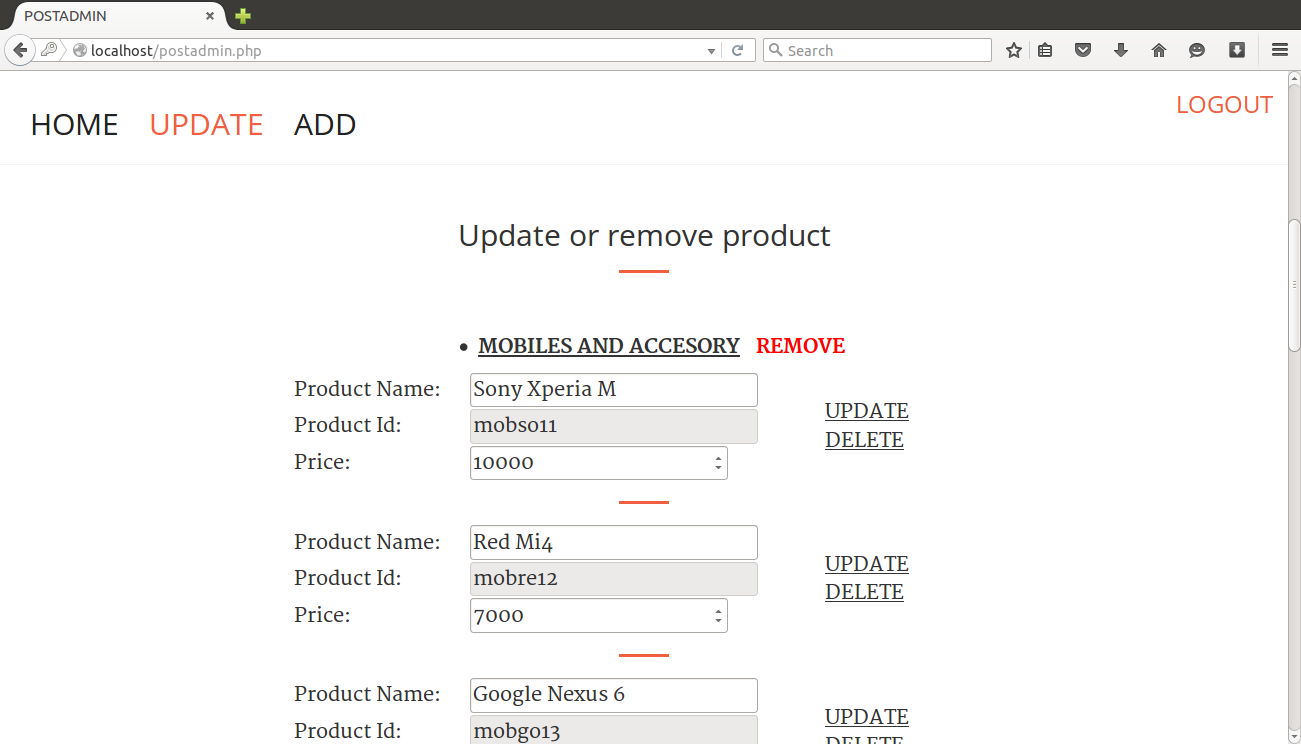


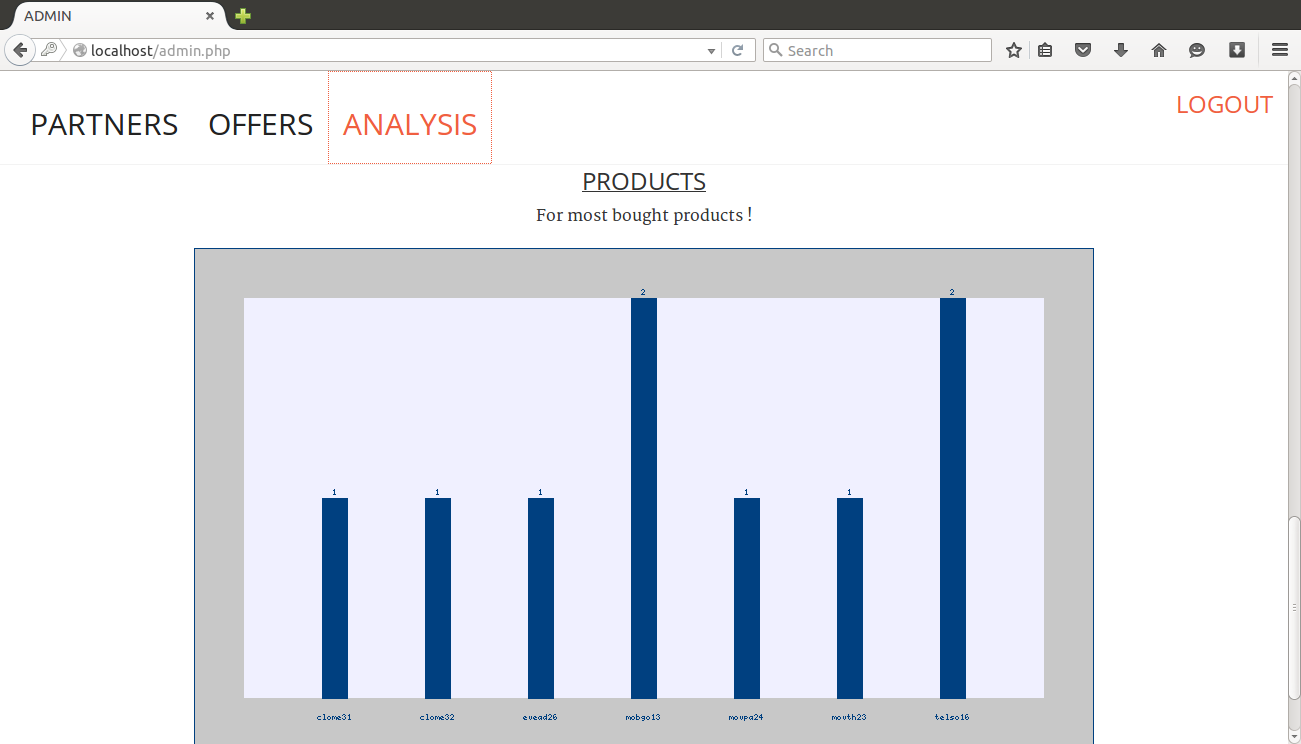


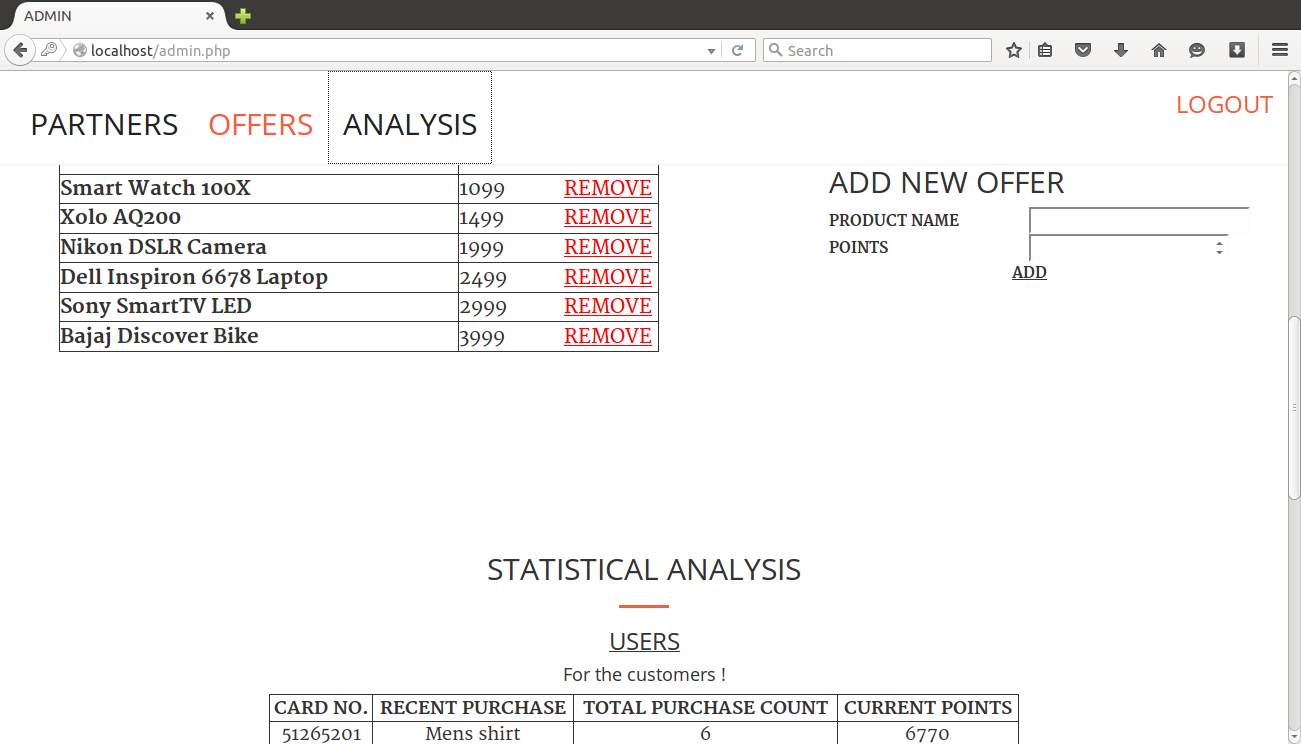
# 











# CONCLUSION-

The three parts which are essential for this project are User interface, Creation of collections and NoSQL engine.

The Project is entirely based on database management system concepts. The back-end use for project is MongoDB and front-end is based on PHP . The Coding of NoSql queries is done through PHP.

**Concepts used-**

1)MongoDB Schema

2)PHP

3)Apache 2.4.7 (ubuntu) server

4) Connectivity with Front End Language

5)Procedure in MongoDB

6)MapReduce