ACKNOWLEDGEMENT

First and foremost, we would like to convey our gratitude to the department of Computer Engineering for giving us such an opportunity to present our skills and ability. We have tried our level best to complete our proposed project successfully. We are greatly indebted to our project supervisor **Er. Amar Bahadur Gurung** and project coordinator **Er. Reshma Maharjan** for the suggestions and guidance despite their busy schedule. We would also like to thank **Er. Sudeep Shakya** (Head of Department, Computer Engineering), and friends for giving us invaluable suggestion for decision making of this project.

ABSTRACT

Show Shopper is data analysis in an inventory system. It analyzes the data to help an organization in better and informed decision making. With analyzed data, it provides recommendation to the customer for items frequently bought together and visualization to the employee which will help them in identifying the customer's need. All the analysis is done on R: an open source data analysis tool. Apriori algorithm is used for implementation of association analysis from the dataset of transaction. This system can be implemented in any inventory market, as it also provides user with simple Management Information System (MIS) functionality which makes this system dynamic; as new transaction, by the customer, also updates the database with new patterns. Database used in this system is MySQL. This system segments customer into 4 hierarchy according to the importance towards the organization. Reference to this cluster result will give user the ability to know the valued customers and make plans according to their needs. Customer behaviors are record on the basis of RFM analysis, which will support employees in understanding the customer to the system. RFM analysis of real world data with the response of customer to a marketing scheme is used for training models like Random Forest, SVM and fast AdaBoost for finding the customer who will response to our recommendation and messages. This prediction based system will help minimize marketing budget allowing user to know the outcome beforehand.

Keywords:

RFM analysis, Apriori, Random Forest, SVM, fast AdaBoost

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIAITONS

CDA: Confirmative Data Analysis

DFD: Data Flow Diagram

EDA: Exploratory Data Analysis

MIS: Management Information System

MBA: Market Basket Analysis

RDB: Relational Database

RFM: Recency Frequency Monetary

SVM: Support Vector Machine

SQL: Structured Query Language

UPC: Universal Project Code

CHAPTER 1: INTRODUCTION

1.1 Background Theory

Data Analysis is the process of systematically applying statistical and/or logical techniques to describe and illustrate, condense and recap, and evaluate data. It is a process for obtaining raw data and converting it into information useful for decision making by users. It can be divided into exploratory data analysis (EDA) and confirmatory data analysis (CDA). EDA focuses on discovering new features in the data and CDA confirms or falsify existing hypothesis.

There are several phases in data analysis process, which are: data collection, data processing, data cleaning, exploratory data analysis, modeling algorithms and data product. Data are collection done from real world, this data that we collected are not in a formatted order therefore they must be cleaned and processed to bring it to a point where it can be analyzed.

Data analysis can be used in many field, one such field is market. Application of data analysis in market are market basket analysis, customer segmentation, customer profitability, churn analysis etc. In a market, we collect typical manner in which customer purchase good, such data are usually known as Market Basket Transaction. Each transaction is listed in a row with a unique id. Retailer are concerned with analyzing such data to produce valuable information that can be used to support business related application such as inventory management, marketing promotions and customer relationship management. Analysis in these basket formats of transaction is known as Market Basket Analysis(MBA). It is a modelling technique based upon the theory that if you buy a certain group of items, you are more (or less) likely to buy another group of items. For example, if you are in an English pub and you buy a pint of beer and don't buy a bar meal, you are more likely to buy chips at the same time than somebody who didn't buy beer [1].

Technique for implementing Market Basket Analysis to find relationship between items can be done with Association analysis; which is used to produce important information from large unknown set of data. Inventory management would get a boost if used along with this. All the data produced through regular transaction with the

costumer, if can be analyzed would be very beneficial; resulting in the boost of retailer's business.

Another type of analysis that can be performed in a market is customer profiling or also known to be segmentation. It uses RFM analysis where R stands for recency, F for frequency and M for monetary. It has many benefit such as increased customer retention, increased response rate, increased conversion rate and revenue [2].

To get to the customers and let them know about the new products in our Inventory is a challenging task, and needs a lot of creative ideas. Customers gets the stuff of the brand only they trust. Also, all the customers will not take the item recommendation in a positive way and if are persuaded more might cut the deal with the retailer. Also on recommending the customer about a product resources are spent. In order to find and select the customer who are likely to respond to the recommendation by our system, customer behavior is analyzed from the previous customer's data. On the basis of RFM analysis and using predictive algorithms, such as Random Forest, SVM and AdaBoost, customers are classified according to their decision and response to the system's recommendation. Only the customer who have high chance of responding to our system are then sent product recommendation messages and offers.

1.2 Problem Statement

Presently, small mart stores data but seldom use it in the business except for the verification. These data are just bits of information which rusts away in some storage device. Small marts lack the knowledge and understanding of data analysis. Therefore, have not opted for one yet. Which if used could be source of an invaluable information for business development. Retail market does not have the knowledge to base their marketing schemas and recommendation, and using this real-time data from the customer user can produce higher revenue with low expenditure.

1.3 Objectives

To learn behavior of customers and find their needs by analyzing data such as sales and buying pattern collected from our Inventory Management system.

1.4 Scope and Application

This system will be applicable to small and medium inventories which want to use stored data to produce result for their better business decision. It can be used to produce recommendation to the customer with frequently bought itemset suggestions. It also can produce itemset pattern visualization to the employees to help them make decision in case of schemes and advertisement. Customer's most valued to the organization are segmented which helps in marketing new scheme to with higher response rate from the customer. This system helps the owner peak into customer's mind and understand them.

CHAPTER 2: LITERATURE REVIEW

Statistician John Tukey defined data analysis in 1961 as: "Procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning, gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results of (mathematical) statistics which apply to analyzing data [3]." There have been various data analysis tool for analysis, and one of them is R.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display [4]. R can be regarded as an implementation of the S language which was developed at Bell Laboratories by Rick Becker, John Chambers and Allan Wilks, and also forms the basis of the S-Plus systems. The evolution of the S language is characterized by four books by John Chambers and coauthors. For R, the basic reference is The New S Language: A Programming Environment for Data Analysis and Graphics by Richard A. Becker, John M. Chambers and Allan R [5]. Wilks. The new features of the 1991 release of S are covered in Statistical Models in S edited by John M. Chambers and Trevor J. Hastie [6]. The formal methods and classes of the methods package are based on those described in Programming with Data by John M. Chambers[7].

R is a statistical tool with many classical and modern statistical techniques as packages. It has about 25 packages that are standard and recommended. One of those package is arules. It provides the generic function and the methods to abbreviate long item labels in transactions, associations (rules and itemsets) and transaction ID lists [8]. A market transaction data is cleaned, preprocessed and mined for data through arules in thesis published by Pazaras Christos, where a case study on a dataset containing 247535 records with 62037 items contained in 33701 transactions of a supermarket was performed [9]. The association rules, after the analysis can be viewed using another package in R which is arulesViz. This apriori analysis when combined with RFM analysis in R, become a great tool for market revenue generation. RFM uses sales data to segment a pool of customers based on their purchasing behavior. The resulting customer segments are neatly ordered from most valuable to least valuable[10].

In present scenario, advertisement and customer response prediction is the best thing that can decreasing the marketing expenditure. In August 2015, a group of scientists

came together to produce a Journal on Marketing Effectiveness [11]. Our project produces a model that predicts the response from the customer using predicting algorithms. Our system uses predictive algorithms like SVM, random Forest and fast adBoost [12] [13].

CHAPTER 3: RELATED THEORY

Association Rule: An association rule is an implication expression of form X→Y, where X and Y are disjoint itemset, i.e. X∩Y = Ø. The strength of an association rule can be measured in terms of its support and confidence. Support determines how often arule is applicable to a given dataset, while confidence determines how frequently items in Y appear in transaction that contain X. The formal definitions of these metrics are

Support, $s(X \rightarrow Y) = \frac{\sigma(x \cup y)}{N}$ ……3.1

Confidence, $c(X \rightarrow Y) = \frac{\sigma(x \cup y)}{\sigma(X)}$ ……3.2

R: R is a language and environment for statistical computing and graphics. R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering…), graphical techniques, and is highly extensible. One of R's strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

It is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSD and Linux), Windows and MacOS [14].

PHP: PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Instead of lots of commands to output HTML (as seen in C or Perl), PHP pages contain HTML with embedded code that does "something" (in this case, output "Hi, I'm a PHP script!"). The PHP code is enclosed in special start and end processing instructions <? php and ?> that allow you to jump into and out of "PHP mode."

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. [15].

D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation. D3 allows you to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven transformations to the document. For example, you can use D3 to generate an HTML table from an array of numbers. Or, use the same data to create an interactive SVG bar chart with smooth transitions and interaction. D3 is not a monolithic framework that seeks to provide every conceivable feature. Instead, D3 solves the crux of the problem: efficient manipulation of documents based on data. This avoids proprietary representation and affords extraordinary flexibility, exposing the full capabilities of web standards such as HTML, SVG, and CSS. With minimal overhead, D3 is extremely fast, supporting large datasets and dynamic behaviours for interaction and animation. D3's functional style allows code reuse through a diverse collection of official and community-developed modules.

It is a library that's designed to manipulate graphical objects (and more) on a web page. The two of them will work really well together, but the barrier to getting data onto a web page can be slightly daunting because the combination of two non-trivial technologies can be difficult to achieve[16].

MySQL: MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. Its name is a combination of "My", the name of co-founder Michael Widenius' daughter, and "SQL", the abbreviation for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation [17].

RFM analysis: To perform RFM analysis, each customer is assigned a score for recency, frequency, and monetary value, and then a final RFM score is calculated.

Recency score is calculated based on the date of their most recent purchase. The scores are generally categorized based on the values. For example, a company may

follow a category system of 1 to 5, score of 5 being the highest. In this case, customers who purchased within the last one month have a recency score of five, customers who purchased within the last 1-3 months have a score of four and so on. Similarly, frequency score is calculated based on the number of times the customers purchased. Customers with higher frequency receive a higher score

Finally, customers are assigned a score based on the amount they spent on their purchases. For calculating this score, you may consider the actual amount spent or the average spent per visit. By combining these three scores, a final RFM score is calculated. The customers with the highest RFM score are considered to be the ones that are most likely to respond to their offers.

RFM analysis is a powerful technique to help you identify your best customers and create better targeted campaigns. However, RFM itself is not enough and retailers should focus on creating more detailed customer profiles including their demographics, behavioral and purchase patterns and use this information in conjunction with RFM to provide better value to customers [18].

Random Forest: A Random Forest consists of a collection or ensemble of simple tree predictors, each capable of producing a response when presented with a set of predictor values. For classification problems, this response takes the form of a class membership, which associates, or classifies, a set of independent predictor values with one of the categories present in the dependent variable. Alternatively, for regression problems, the tree response is an estimate of the dependent variable given the predictors. The Random Forest algorithm was developed by Breiman.

A Random Forest consists of an arbitrary number of simple trees, which are used to determine the final outcome. For classification problems, the ensemble of simple trees vote for the most popular class. In the regression problem, their responses are averaged to obtain an estimate of the dependent variable. Using tree ensembles can lead to significant improvement in prediction accuracy (i.e., better ability to predict new data cases).

The response of each tree depends on a set of predictor values chosen independently (with replacement) and with the same distribution for all trees in the forest, which is a

subset of the predictor values of the original data set. The optimal size of the subset of predictor variables is given by $\log_2 M+1$, where M is the number of inputs.

For classification problems, given a set of simple trees and a set of random predictor variables, the Random Forest method defines a margin function that measures the extent to which the average number of votes for the correct class exceeds the average vote for any other class present in the dependent variable. This measure provides us not only with a convenient way of making predictions, but also with a way of associating a confidence measure with those predictions.

For regression problems, Random Forests are formed by growing simple trees, each capable of producing a numerical response value. Here, too, the predictor set is randomly selected from the same distribution and for all trees. Given the above, the mean-square error for a Random Forest is given by:

mean error = (observed - tree response)^2

The predictions of the Random Forest are taken to be the average of the predictions of the trees:

Random forest Prediction s= $\frac{1}{K}\sum_{K-1}^{K} Kth$ tree response……3.3

where the index *k* runs over the individual trees in the forest.

Typically, Random Forests can flexibly incorporate missing data in the predictor variables. When missing data are encountered for a particular observation (case) during model building, the prediction made for that case is based on the last preceding (non-terminal) node in the respective tree. So, for example, if at a particular point in the sequence of trees a predictor variable is selected at the root (or other non-terminal) node for which some cases have no valid data, then the prediction for those cases is simply based on the overall mean at the root (or other non-terminal) node. Hence, there is no need to eliminate cases from the analysis if they have missing data for some of the predictors, nor is it necessary to compute surrogate split statistics [19].

Support Vector Machine (SVM): A Support Vector Machine is a supervised machine learning algorithm that can be employed for both classification and

regression purposes. SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes.

Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, they can be considered the critical elements of a data set.

The distance between the hyperplane and the nearest data point from either set is known as the margin. The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set, giving a greater chance of new data being classified correctly [20].

$$\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i, x) + \rho \qquad \text{.....3.4}$$

These parameters can be accessed through the members dual_coef_ which holds the difference $\alpha_i - \alpha_i^*$, support_vectors_ which holds the support vectors, and intercept_ which holds the independent term ρ

AdaBoost: Classification is a machine-learning technique that uses training data to generate a model (usually a single complex rule or mathematical equation) that assigns data items to one of several distinct categories. The model can then be used to make predictions about new data items whose category is unknown. Adaptive boosting classification is a technique in which, instead of attempting to determine a single complex prediction rule, training data is used to generate a large collection of very simple crude rules of thumb. A weight for each rule of thumb is then computed. A prediction about new input is made by combining the rules of thumb, taking into account each simple rule's weight and arriving at a consensus outcome. The term "boosting" comes from the fact that the predictive quality of the simple rules is boosted (improved) by combining them.

Adaptive boosting is a meta-heuristic, which means adaptive boosting is a set of guidelines that can be used to create a specific classification algorithm. There are many variations of adaptive boosting algorithms and there are many existing standalone tools that implement some form of adaptive boosting, so why bother to code adaptive boosting classification from scratch? Existing adaptive boosting

classification tools can be difficult or impossible to customize, they might be difficult to integrate into a software system, and they may have copyright or intellectual property issues [21].

CHAPTER 4: METHODOLOGY

4.1 System Planning

4.1.1 Software Model (Incremental Model)

In Incremental model, the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a multi waterfall cycle. Cycles are divided up into smaller, more easily managed modules. Each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is achieved.

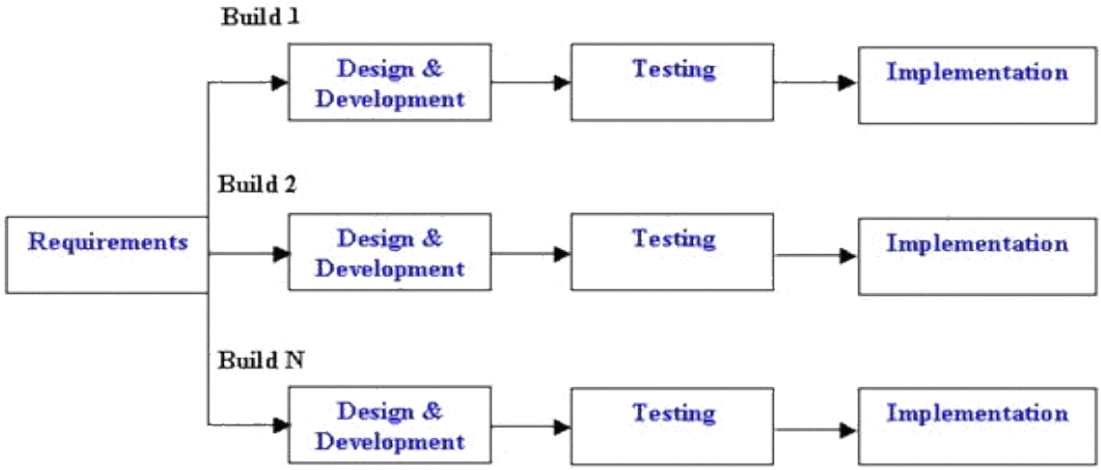


Figure 4.1: Incremental model

Advantages of Incremental model:

- Generates working software quickly and early during the software life cycle.
- This model is more flexible-less costly to change scope and requirements.
- It is easier to test and debug during a smaller iteration.
- In this model customer, can respond to each built.
- Lowers initial delivery cost.

- Easier to manage risk because risky pieces are identified and handled during it'd iteration.

Disadvantages of Incremental model:

- Needs good planning and design
- Needs a clear and complete definition of the whole system before it can be broken down and built incrementally.

4.1.2 Data Analysis Process

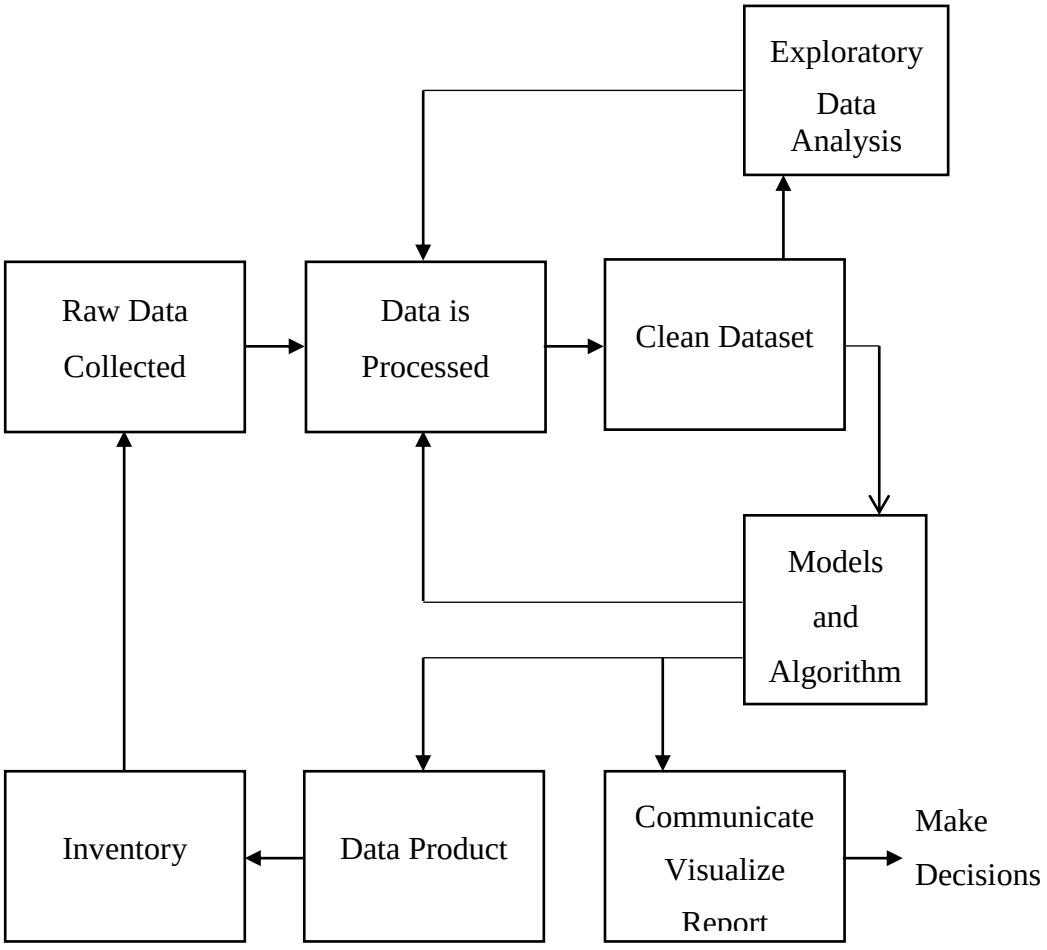


Figure.4.2: Data analysis process

The above figure is the block diagram of Data Analysis process. Raw data of from the real-world transaction was collected from an open source platform with customers behavior data. These raw data were then processed and changed into the format on which we can use market basket analysis and other predictive analysis. After we got

the data in basket form, an apriori algorithm was applied on this “clean dataset” to generate frequent itemset which is later used for recommendation of items in our Show Shopper system. Apriori algorithm is to be performed on R in our project. The obtained result was further processed and modeled to properly upload the generated association rules on our local server database (MySQL). The results were extracted from database in the form of related itemset and visualized properly. Then raw dataset of customers is analyzed on the basis of Recency, Frequency and Monetary (RFM), which then helped in training predictive model that predicted the response of the customer to our recommendation and schema.

4.2 System Analysis

4.2.1 Feasibility Study

Technical Feasibility

The above-mentioned project is technically feasible since the necessary tools and techniques are already available. All the software is easily available, we will use R and other analyzing system for statistical analysis. R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

Schedule Feasibility

For the development of this application the suggested process model is incremental modeling. Following the process model, we will prioritize our schedule on bare minimum functional application development. With this objective in mind with the given time frame of 6 months we will be able to deliver the product at the end of major project.

Economic Feasibility

Most of the project tools and technique that we will be using in this project will be open-source and these tools are easily available and are in disposal for our use. Most of the tools are free with great support, in context to our project with available resources this project is economically feasible.

4.3 System Design

4.3.1 Overview of the System

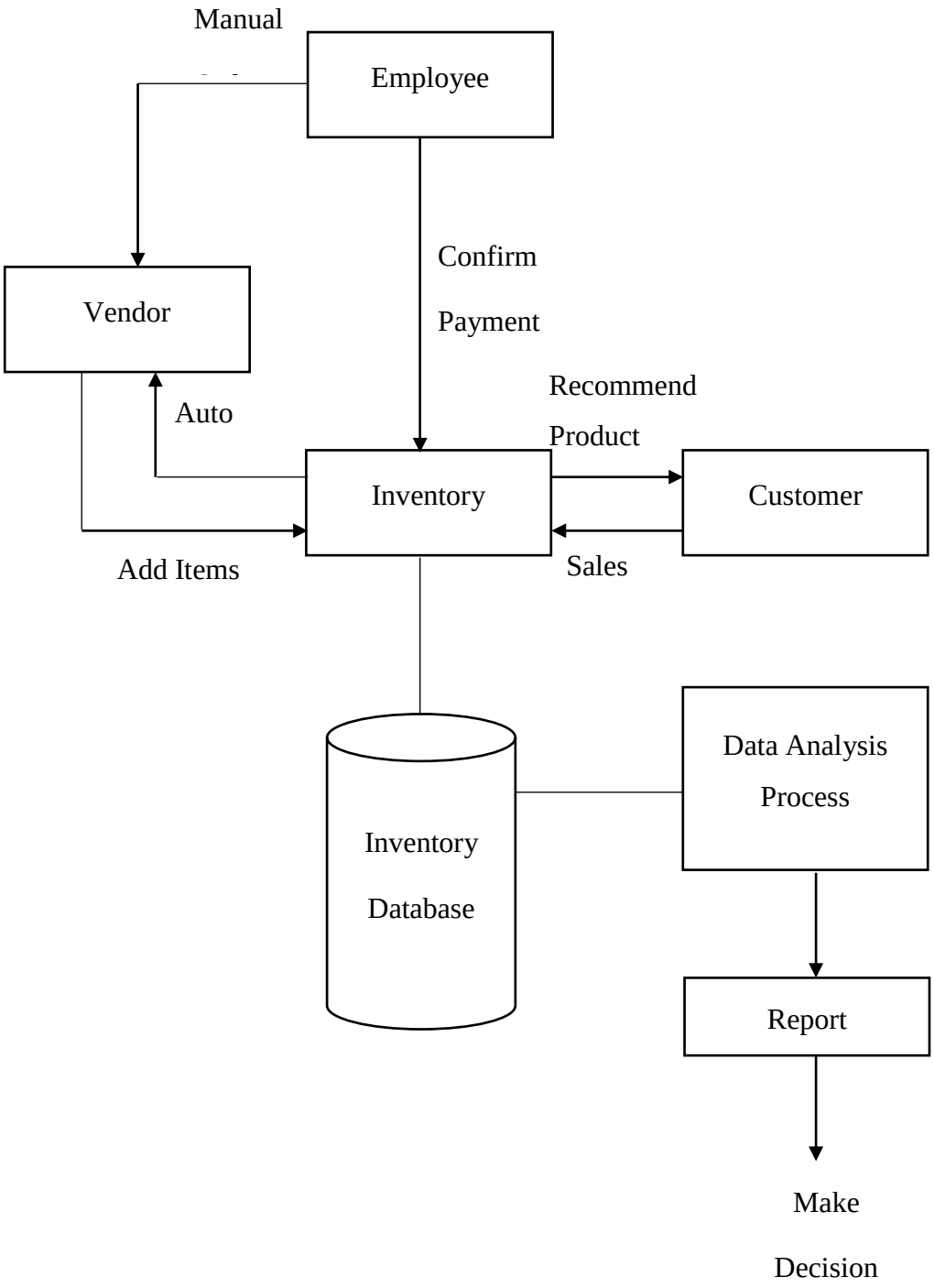


Figure 4.3: Complete system overview

The above figure shows a block diagram of our entire system: Show Shopper. Show Shopper has an inventory acting as a central unit. There are basically two main

entities in the system: Customer and Employee. Each user must login into the system before accessing any features. Customers can view the item list and order the required item by adding it into the cart. During adding items to the cart, customers get a recommended list of items continually. Recommended items are shown for easy and convenient booking of items. Employee can access the customer's cart and after payment confirmation, cart items are handled accordingly. Recommendation to the customers are given after analyzing the data set for frequent itemset, plus a RFM analysis is also done which helps understand the behavior of the customer. According to the customer behavioral data, predictive models are also trained and produced. These models will help in predicting response of customer according to the behavioral pattern.

4.3.2 Data Flow Diagram

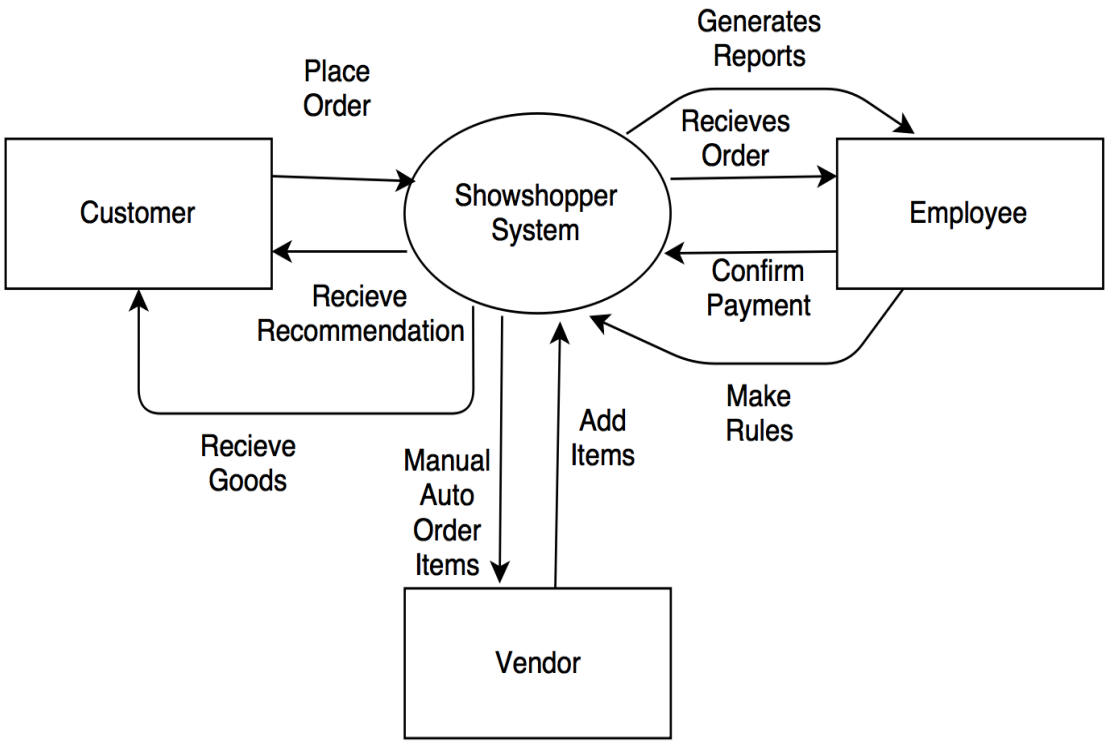


Figure 4.4: Context diagram

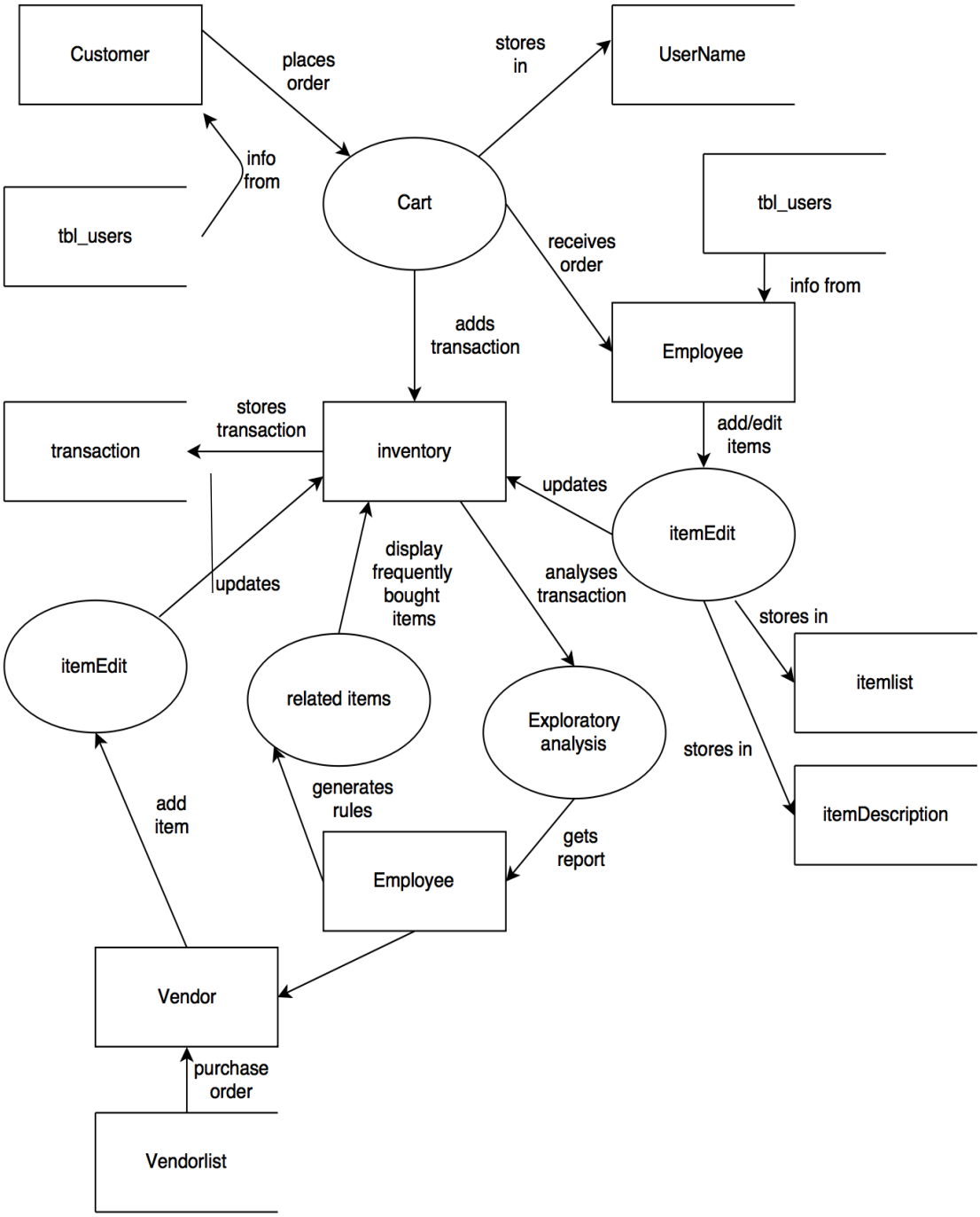


Figure 4.4: DFD level 0 for show shopper

4.3.3 Schema Diagram

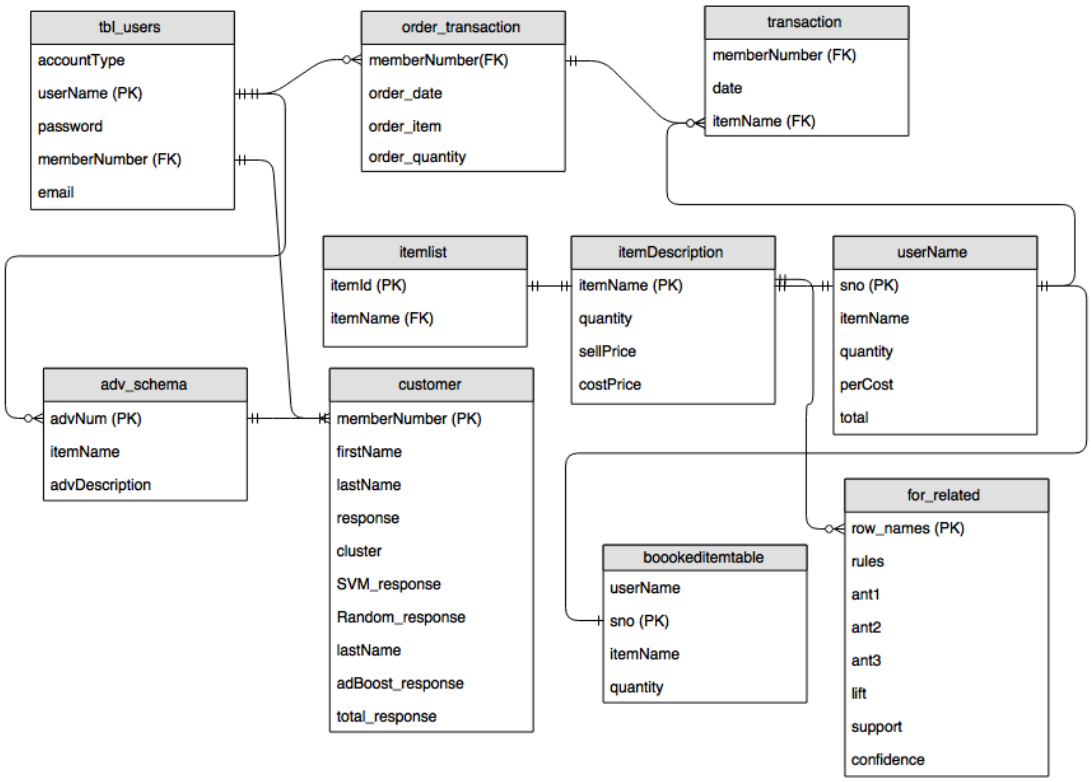


Figure 4.5: Schema diagram for show shopper

4.4 Implementation

Implementation plan

First of all, the end-user is going to use the new system for a week. Any kind of error are going to be recorded in the error log systematically. For us to install the software into the system the computer must need a certain hardware and software. This is managed through the present computer or any kind of addition.

Hardware: The requirement of the system is very low, it will require a perfect running computer. The only piece of equipment that is going to be expensive is 1280 x 1024 resolution screen monitor. This is required in the system to run the program in a sufficient resolution. This would enable a greater amount of data to be viewed on the screen and minimize the use of scroll bars.

Software: To run the new system in the computer, program named R has to be installed into the computer system. It will be installed through a DVD-ROM into the system. It must also have a browser with access to the internet.

Data Structure Used

Table 4.1: Database for adv_schema

Field Name	Data type	Fields Size	Sample	Description
advNum *(primary key)*	int	11	2	Automatically generated, primary key
itemName	varchar	255	beef	Item that is being advertised
advDescription	Text	<<default>>	We have a discount in beef. This will last for 2 days.	Advertisement description

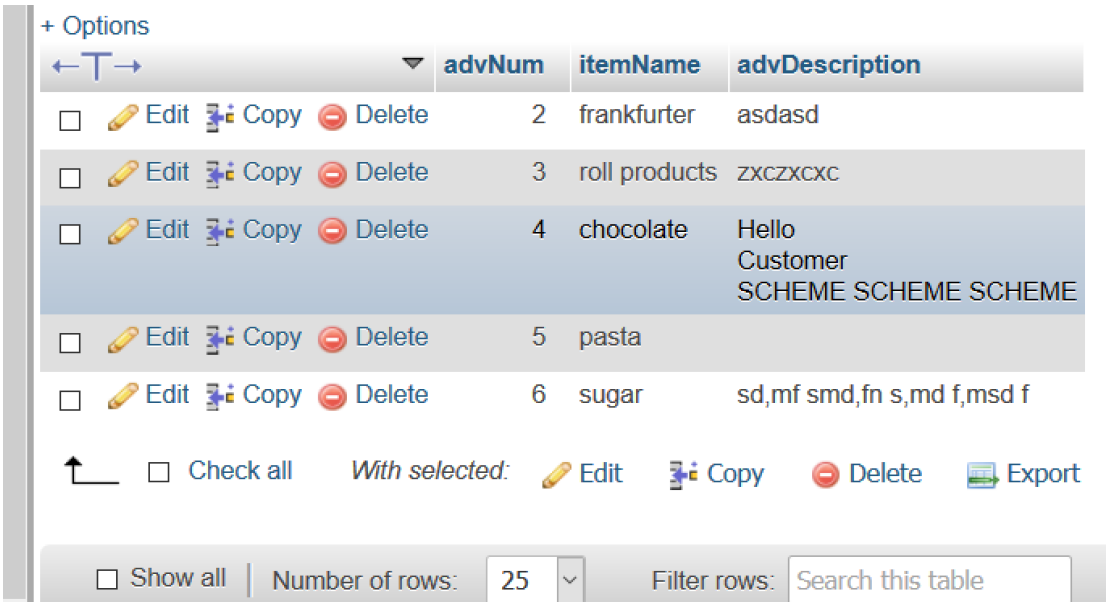
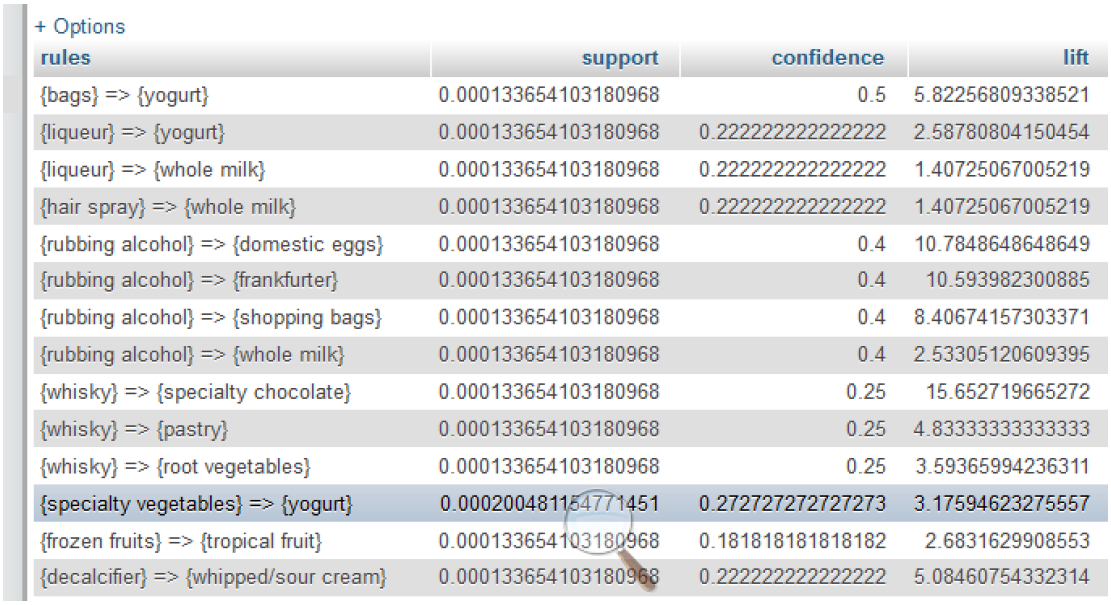


Figure 4.7: Example of adv_schema

Table 4.2: Database for association rules

Field Name	Data type	Fields Size	Sample	Description
rules	text	<<default>>	{bags} => {yogurt}	Rules generated by the R script
support	double	<<default>>	0.000133654103180968	Support of the rules generated
confidence	double	<<default>>	0.5	Confidence of the rules generated
lift	double	<<default>>	5.82256809338521	Lift of the rules generated



+ Options

rules	support	confidence	lift
{bags} => {yogurt}	0.000133654103180968	0.5	5.82256809338521
{liqueur} => {yogurt}	0.000133654103180968	0.222222222222222	2.58780804150454
{liqueur} => {whole milk}	0.000133654103180968	0.222222222222222	1.40725067005219
{hair spray} => {whole milk}	0.000133654103180968	0.222222222222222	1.40725067005219
{rubbing alcohol} => {domestic eggs}	0.000133654103180968	0.4	10.7848648648649
{rubbing alcohol} => {frankfurter}	0.000133654103180968	0.4	10.593982300885
{rubbing alcohol} => {shopping bags}	0.000133654103180968	0.4	8.40674157303371
{rubbing alcohol} => {whole milk}	0.000133654103180968	0.4	2.53305120609395
{whisky} => {specialty chocolate}	0.000133654103180968	0.25	15.652719665272
{whisky} => {pastry}	0.000133654103180968	0.25	4.83333333333333
{whisky} => {root vegetables}	0.000133654103180968	0.25	3.59365994236311
{specialty vegetables} => {yogurt}	0.000200481154771451	0.272727272727273	3.17594623275557
{frozen fruits} => {tropical fruit}	0.000133654103180968	0.181818181818182	2.6831629908553
{decalcifier} => {whipped/sour cream}	0.000133654103180968	0.222222222222222	5.08460754332314

Figure4. 8: Example of association rules

Table 4.3: Database for bookeditemtable

Field Name	Data type	Fields Size	Sample	Description
sno *(Primary Key)*	int	11	1	Automatically generated code for booked item
userName	varchar	100	test1	username of customer or employee who book the item
itemName	varchar	100	tropical fruit	Item of booking
quantity	int	11	10	Quantity of booked item

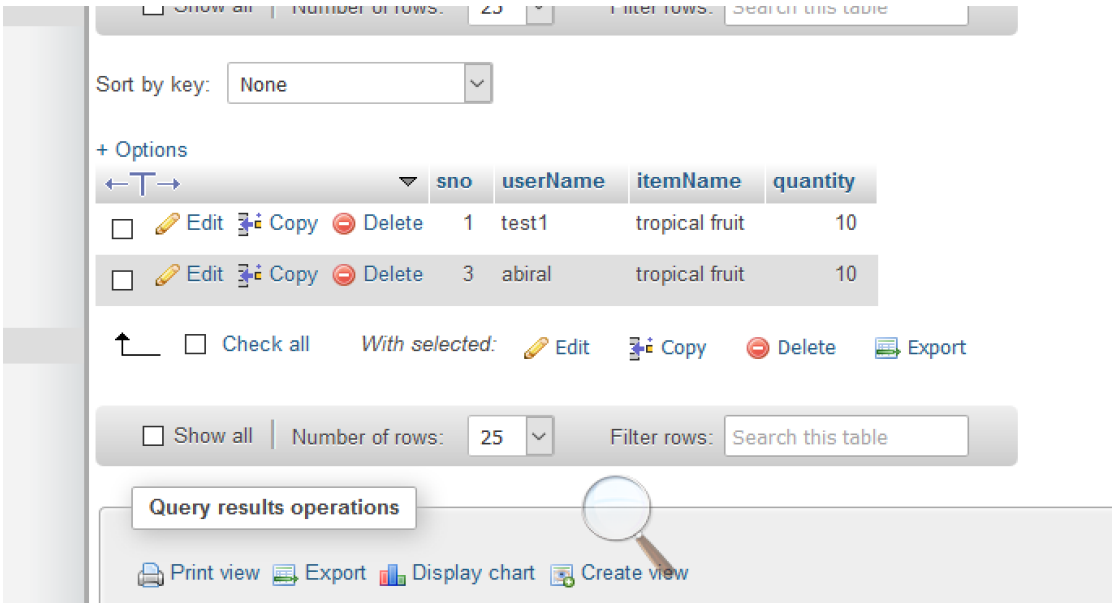


Figure 4.9: Example of bookeditemtable

Table 4.4: Database for customer

Field Name	Data type	Fields Size	Sample	Description
memberNumber	int	11	1000	Member Number of customer who take part in transaction
response	varchar	10	no	Real world response of the customer
Random_response	varchar	10	no	Predicted response from Random Forest
adaBoost_response	varchar	10	no	Predicted response from adaBoost response
cluster	int	11	4	Predicted cluster by kmeans
SVM_response	varchar	10	no	Predicted response of SVM
total_response	int	10	0	Total Oring of yes

Figure 4.10: Example of customer

Table 4.5: Database for for_related

Field Name	Data type	Fields Size	Sample	Description
row_names *(Primary Key)*	text	11	1	Auto generated number for the row count
rules	varchar	500	{frozen fish} => {whole milk}	Support of the rules generated
ant1	varchar	200	frozen fish	Separated header of rules
ant2	varchar	200		Separated head of rules part2
cons	varchar	200	whole milk	Separated tail of rules
lift	double	<<default>>	0.99335341415449	Lift of rules

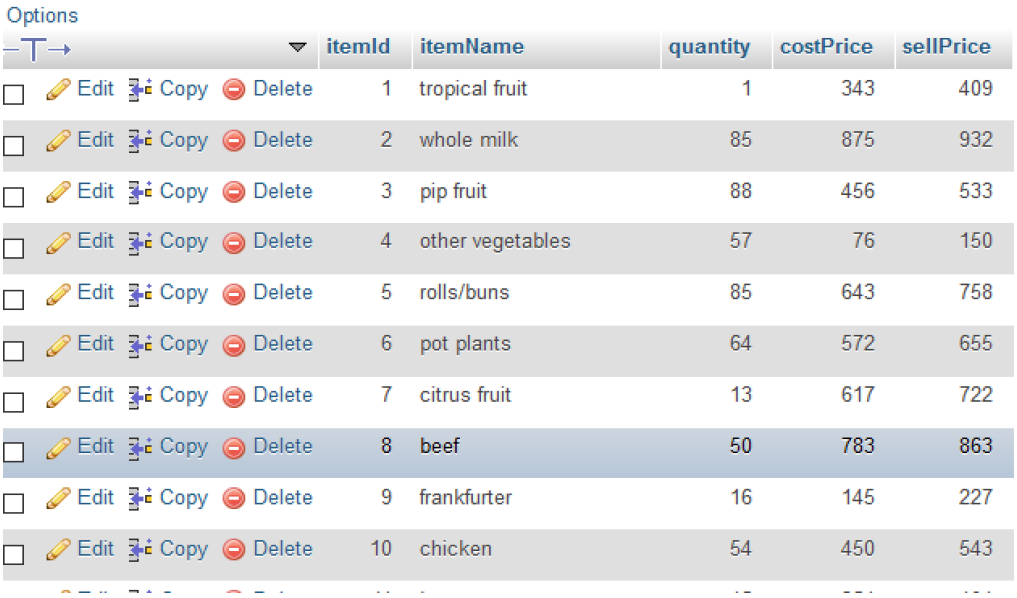
nemberNumber	response	Random_response	adaBoost_response	cluster	SVM_response	total_response
1000	no	no	no	4	no	0
1001	no	no	no	3	no	0
1002	no	no	no	3	yes	1
1003	no	no	no	3	no	0
1004	yes	no	no	4	no	0
1005	no	no	no	1	yes	1
1006	yes	yes	yes	4	no	1

Figure 4.11: Example of for_related

Table 4.6: Database for itemlist

Field Name	Data type	Fields Size	Sample	Description
itemId *(Primary Key)*	int	11	1	Auto Generated Id for item
itemName	varchar	100	tropical fruit	Name of item
quantity	int	11	1	Number of item in stock
costPrice	float	<<default>>	343	Cost Price for the item
sellPrice	float	<<default>>	409	Sell Price for the item

row_names	rules	ant1 ant2	cons	lift
1	{frozen fish} => {whole milk}	frozen fish	whole milk	0.99335341415449
2	{seasonal products} => {rolls/buns}	seasonal products	rolls/buns	1.28648066209679
3	{pot plants} => {other vegetables}	pot plants	other vegetables	1.05006105006105
4	{pot plants} => {whole milk}	pot plants	whole milk	0.811875386568573
5	{pasta} => {whole milk}	pasta	whole milk	0.83737229953519
6	{pickled vegetables} => {whole milk}	pickled vegetables	whole milk	0.708876270362112
7	{packaged fruit/vegetables} => {rolls/buns}	packaged fruit/vegetables	rolls/buns	1.28850661589537
8	{rolls/buns} => {packaged fruit/vegetables}	rolls/buns	packaged fruit/vegetables	1.28850661589537
9	{detergent} => {yogurt}	detergent	yogurt	1.4443579766537
10	{yogurt} => {detergent}	yogurt	detergent	1.4443579766537
11	{detergent} => {rolls/buns}	detergent	rolls/buns	1.05710814094775
12	{detergent} => {whole milk}	detergent	whole milk	1.03089293271265



Options

	itemId	itemName	quantity	costPrice	sellPrice
Edit Copy Delete	1	tropical fruit	1	343	409
Edit Copy Delete	2	whole milk	85	875	932
Edit Copy Delete	3	pip fruit	88	456	533
Edit Copy Delete	4	other vegetables	57	76	150
Edit Copy Delete	5	rolls/buns	85	643	758
Edit Copy Delete	6	pot plants	64	572	655
Edit Copy Delete	7	citrus fruit	13	617	722
Edit Copy Delete	8	beef	50	783	863
Edit Copy Delete	9	frankfurter	16	145	227
Edit Copy Delete	10	chicken	54	450	543

Figure 4.12: Example of itemlist

Table 4.7: Database for order_transaction

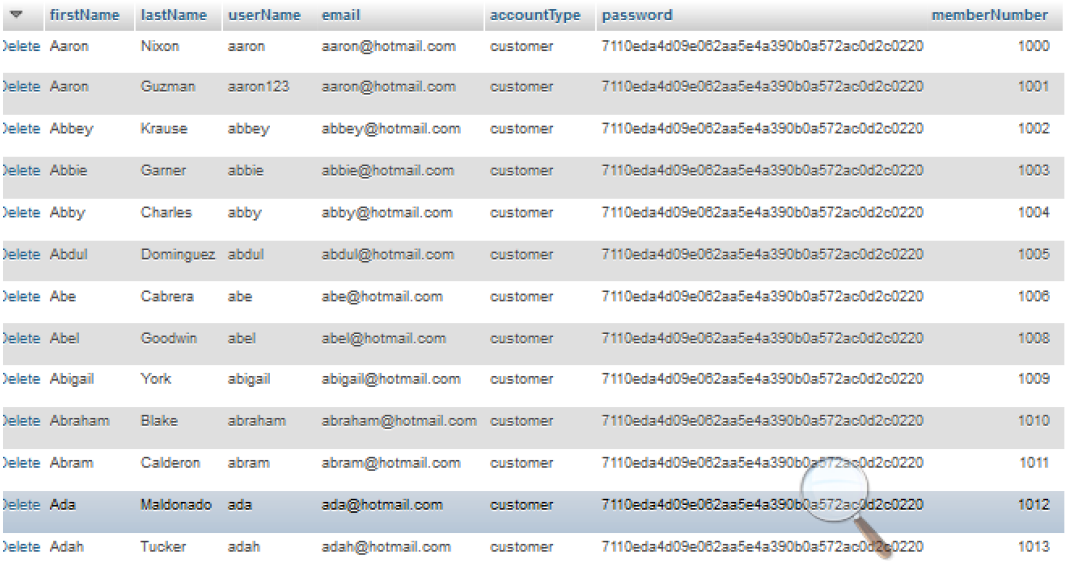
Field Name	Data type	Fields Size	Sample	Description
order_id *(Primary Key)*	int	11	1	Auto generated Id for the order
order_date	date	<<default>>	2017-06-07	Date of ordering
order_item	text	<<default>>	potato products	Ordered Item
order_quantity	int	11	1	Quantity of order item

Figure 4.13: Example of order_transaction

Table 4.8: Database for tbl_user

Field Name	Data type	Fields Size	Sample	Description
firstName	varchar	50	Aaron	First Name of the Customer and Employee
lastName	varchar	50	Nixon	Last Name of customer and employee
userName *(Primary Key)*	varchar	20	aaron	User Name of customer and employee

email	text	<<default>>	aaron@hotmail.com	Email of customer and employee
accountType	varchar	20	customer	Type of account either employee and customer
password	varchar	50	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	Password of employee and customer
memberNumber *(Unique Key)*	int	11	1000	Member Number of the customer or employee



	firstName	lastName	userName	email	accountType	password	memberNumber
Delete	Aaron	Nixon	aaron	aaron@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1000
Delete	Aaron	Guzman	aaron123	aaron@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1001
Delete	Abbey	Krause	abbey	abbey@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1002
Delete	Abbie	Garner	abbie	abbie@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1003
Delete	Abby	Charles	abby	abby@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1004
Delete	Abdul	Dominguez	abdul	abdul@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1005
Delete	Abe	Cabrera	abe	abe@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1006
Delete	Abel	Goodwin	abel	abel@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1008
Delete	Abigail	York	abigail	abigail@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1009
Delete	Abraham	Blake	abraham	abraham@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1010
Delete	Abram	Calderon	abram	abram@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1011
Delete	Ada	Maldonado	ada	ada@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1012
Delete	Adah	Tucker	adah	adah@hotmail.com	customer	7110eda4d09e062aa5e4a390b0a572ac0d2c0220	1013

Figure 4.14: Example of tbl_user

Table 4.9: Database for transaction

Field Name	Data type	Fields Size	Sample	Description

memberNumber	int	11	1808	Member Number of customer from tbl_users
purchaseDate	date	<<default>>	2015-07-21	Date of purchase of any item
itemName	varchar	200	tropical fruit	Name of item purchased

+ Options

memberNumber	purchaseDate	itemName
1808	2015-07-21	tropical fruit
2552	2015-01-05	whole milk
2300	2015-09-19	pip fruit
1187	2015-12-12	other vegetables
3037	2015-02-01	whole milk
4941	2015-02-14	rolls/buns
4501	2015-05-08	other vegetables
3803	2015-12-23	pot plants
2762	2015-03-20	whole milk
4119	2015-02-12	tropical fruit
1340	2015-02-24	citrus fruit
2193	2015-04-14	beef
1997	2015-07-21	frankfurter
4546	2015-09-03	chicken

Figure 15: Example of transaction

Table 4.10: Database for vendor

Field Name	Data type	Fields Size	Sample	Description
vendorId *(Primary Key)*	int	11	1	Automatically generated Id for vendor
vendorName	varchar	200	Buddha Vendors	Name of the vendor

vendorEmail	text	<<default>>	buddhavendors@gmail.com	Email of the vendor
vendorLocation	varchar	100	Balaju-15	Vendor Location
itemCategory	int	50	1	Category of item that they provide

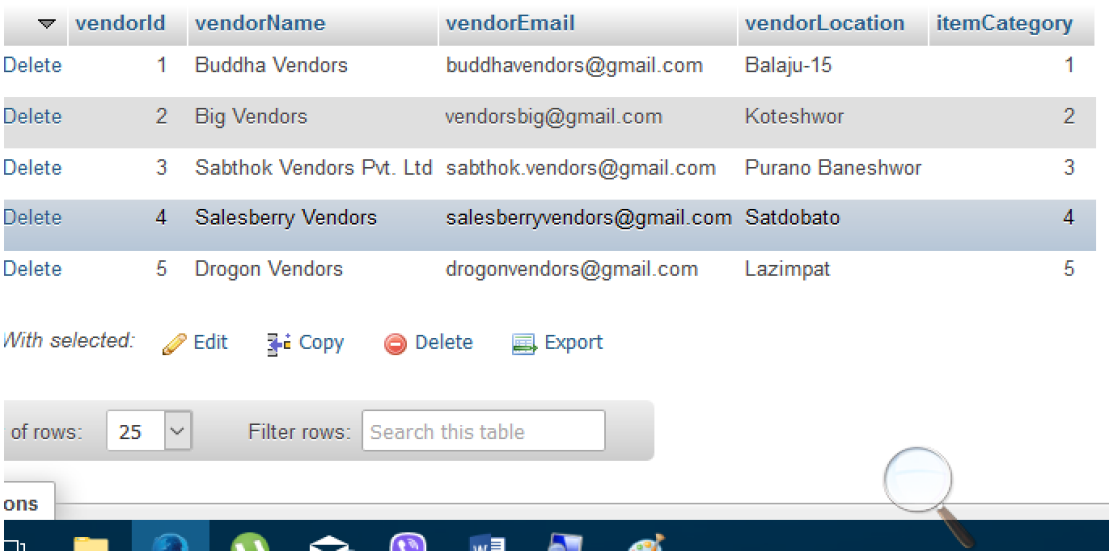


Figure 4.16: Example of vendor

Data Analysis Process

Data Collection: The data collection for the analysis process was obtained through an open source platform. The data has 5000 unique customers, with 38766 transaction data. The data has 3 attributes: date, member number and item. The data sample is shown below.

	A	B	C	D
1	member_number	date	itemDescription	
2	1808	21/07/2016	tropical fruit	
3	2552	05/01/2016	whole milk	
4	2300	19/09/2016	pip fruit	
5	1187	12/12/2016	other vegetables	
6	3037	01/02/2016	whole milk	
7	4941	14/02/2016	rolls/buns	
8	4501	08/05/2016	other vegetables	
9	3803	23/12/2016	pot plants	
10	2762	20/03/2016	whole milk	
11	4119	12/02/2016	tropical fruit	

Figure 4.17: All the data collected

Data cleaning & processing: This data was imported into MySQL database where it was stored for future use. The stored data is then retrieved into R for cleaning and preprocessing. White strip before and after the data tuples where removed when reading the data from MySQL. With the help of plyr function in R, we processed the atomic data to produce transactional data. As we require the data to be in basket format for market basket analysis. Data set after processing is shown below:

	itemList
1	whole milk,pastry,salty snack
2	sausage,whole milk,semi-finished bread,yogurt
3	soda,pickled vegetables
4	canned beer,misc. beverages
5	sausage,hygiene articles
6	sausage,whole milk,rolls/buns
7	whole milk,soda
8	frankfurter,soda,whipped/sour cream
9	beef,white bread
10	frankfurter,curd
11	frozen vegetables,other vegetables
12	butter,whole milk
13	tropical fruit,sugar
14	butter milk,specialty chocolate
15	frozen meals,dental care

Figure 4.18:6 Data in basket format after cleaning & preprocessing

Above dataset are used for market basket analysis and then to train a model. Customers behavior on the basis of RFM is calculated and produced

	memberNumber	Recency	Frequency	Monetary	recency.log	frequency.log	monetary.log	recency.z
55	1000	615.2396	13	696.2308	6.422012	2.5649494	6.545681	-1.037194299
27	1001	822.2396	12	690.3333	6.712032	2.4849066	6.537175	0.452404568
77	1002	702.2396	8	512.1250	6.554275	2.0794415	6.238569	-0.357866707
69	1003	903.2396	8	721.0000	6.805988	2.0794415	6.580639	0.934980785
96	1004	608.2396	21	525.0000	6.410569	3.0445224	6.263398	-1.095967249
69	1005	1286.2396	4	773.5000	7.159478	1.3862944	6.650926	2.750575299
08	1006	779.2396	15	557.8000	6.658319	2.7080502	6.324000	0.176522893
60	1008	668.2396	12	628.9167	6.504647	2.4849066	6.443999	-0.612764982
21	1009	666.2396	9	541.4444	6.501649	2.1972246	6.294240	-0.628160341
02	1010	732.2396	12	504.9167	6.596108	2.4849066	6.224393	-0.143003709
70	1011	601.2396	13	569.0000	6.398993	2.5649494	6.343880	-1.155420525
76	1012	621.2396	11	668.2727	6.431717	2.3978953	6.504696	-0.987347284
84	1013	669.2396	19	536.6842	6.506142	2.9444390	6.285410	-0.605084573
62	1014	668.2396	10	654.3000	6.504647	2.3025851	6.483566	-0.612764982
94	1015	820.2396	7	789.0000	6.709596	1.9459101	6.670766	0.439896172

Figure 4.19: RFM analysis after cleaning & preprocessing

Model & algorithm: After the basket of item where produced from the data set, we applied the apriori analysis to produce association rules. Association rules achieved is shown below:

	rules	support	confidence	lift
1	{frozen fish} => {whole milk}	0.001069233	15.686275	0.9933534
2	{seasonal products} => {rolls/buns}	0.001002406	14.150943	1.2864807
3	{pot plants} => {other vegetables}	0.001002406	12.820513	1.0500611
4	{pot plants} => {whole milk}	0.001002406	12.820513	0.8118754
5	{pasta} => {whole milk}	0.001069233	13.223140	0.8373723
6	{pickled vegetables} => {whole milk}	0.001002406	11.194030	0.7088763
7	{packaged fruit/vegetables} => {rolls/buns}	0.001202887	14.173228	1.2885066
8	{rolls/buns} => {packaged fruit/vegetables}	0.001202887	1.093560	1.2885066
9	{detergent} => {yogurt}	0.001069233	12.403101	1.4443580
10	{yogurt} => {detergent}	0.001069233	1.245136	1.4443580
11	{detergent} => {rolls/buns}	0.001002406	11.627907	1.0571081

Figure 4.20 Association Rules extracted from the data set

Models like Random Forest, SVM, fast adaBoost are trained for classification. Present data set are divided into training set and testing set with 30 percent in training set and 70 percent in testing set. Model is then used to predict the response of the customer.

Testing Plan

Table 4.11: Testing Plan

Purpose	Input	Expected Result
Test 1: To test whether the field is empty or not during sign up process	We leave out Last Name field during sign up.	Prints “Field Empty” just at the side of the empty field
Test 2: To test whether the signed up user is uploaded into the database “tbl_users”	We fill out every field details and Sign up.	All the details is added into the database “tbl_users”
Test 3: To test with wrong username or/and password during login	We fill out wrong password for “abhaya”.	Alert of Unsuccessful login is shown
Test 4: To test with correct username and password during login	We fill out correct password for “abhaya”.	“abhaya” is logged in and enters into itemlist page.
Test 5: To test whether the cart table is created as soon as the user logs in	We login as “abhaya” into the system.	A table named “abhaya” is created in the database which is used for carting purpose.
Test 6: To test add items to cart with quantity greater than remaining quantity	We try to add 100 “whole milk” when remaining quantity is 75.	Error message prints out.
Test 7: To test whether the items with quantity are being add to cart and database	We add 50 whole milk and 20 ham.	Items are added in cart and cart database
Test 8: To test empty field in quantity field	We put 0 or either leave empty quantity field.	Error message prints out

Test 9: To test by viewing empty cart	We delete all cart records by “Clear All” button.	Alert message saying “Cart is empty” pops up.
Test 10: To test new member is divided into cluster	Input is “abhaya” customer with member Number = 5003	Customer Abhaya is put into one of the cluster
Test 11: To test whether prediction is made for new customer	Input is “abhaya” customer with member Number = 5003	Customer Abhaya response if predicted by the model

Testing

Test 1: To test whether the field is empty or not during sign up process

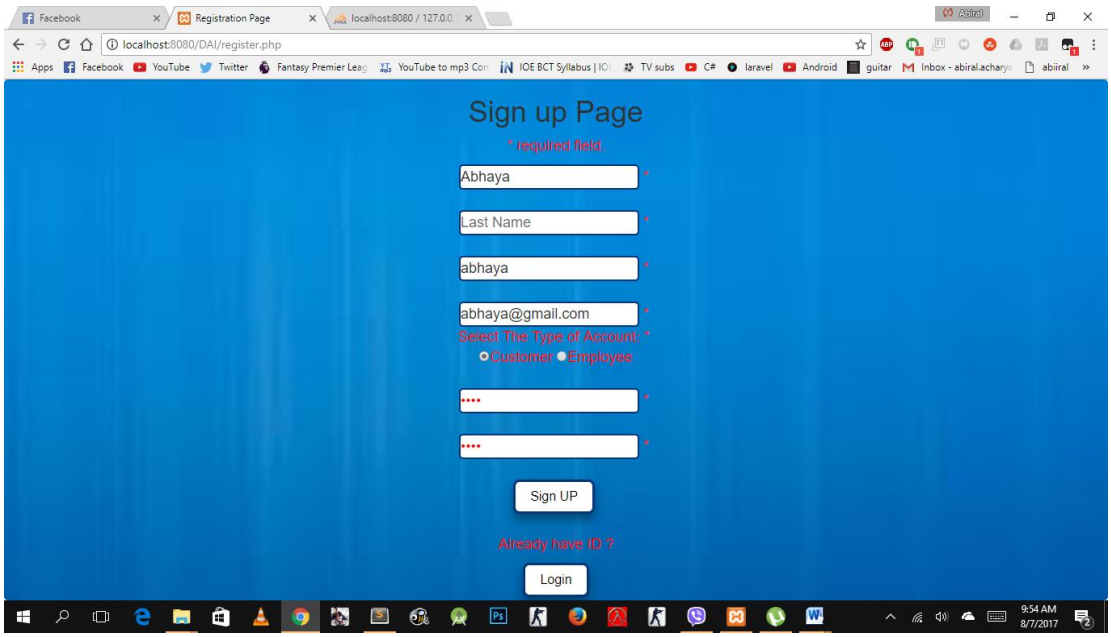


Figure 4.21: Last name is left out empty

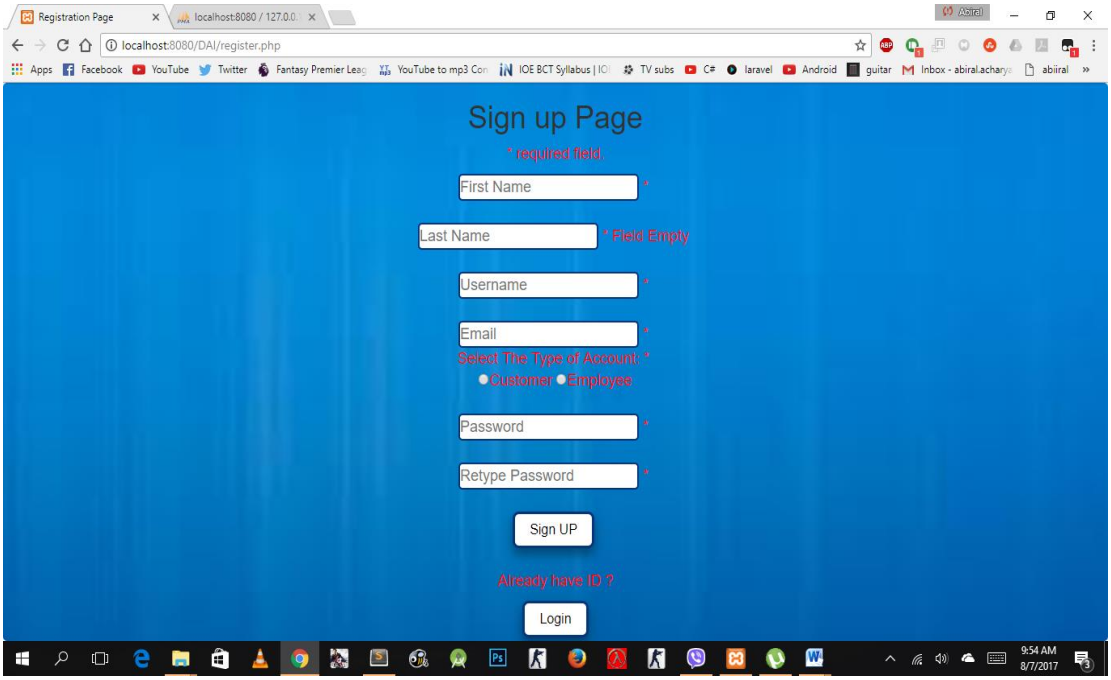


Figure 4.22 Error message (field empty)

Test 2: To test whether the signed up user is uploaded into the database "tbl_users"

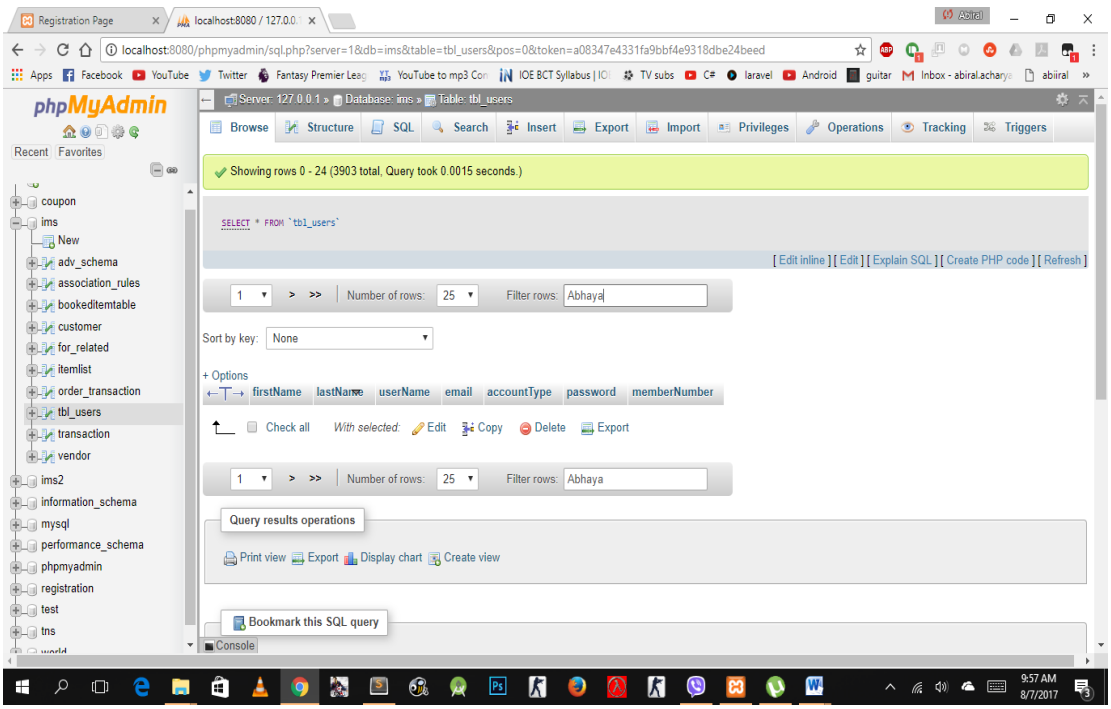


Figure 4.23 Before database (search "abhaya" in database)

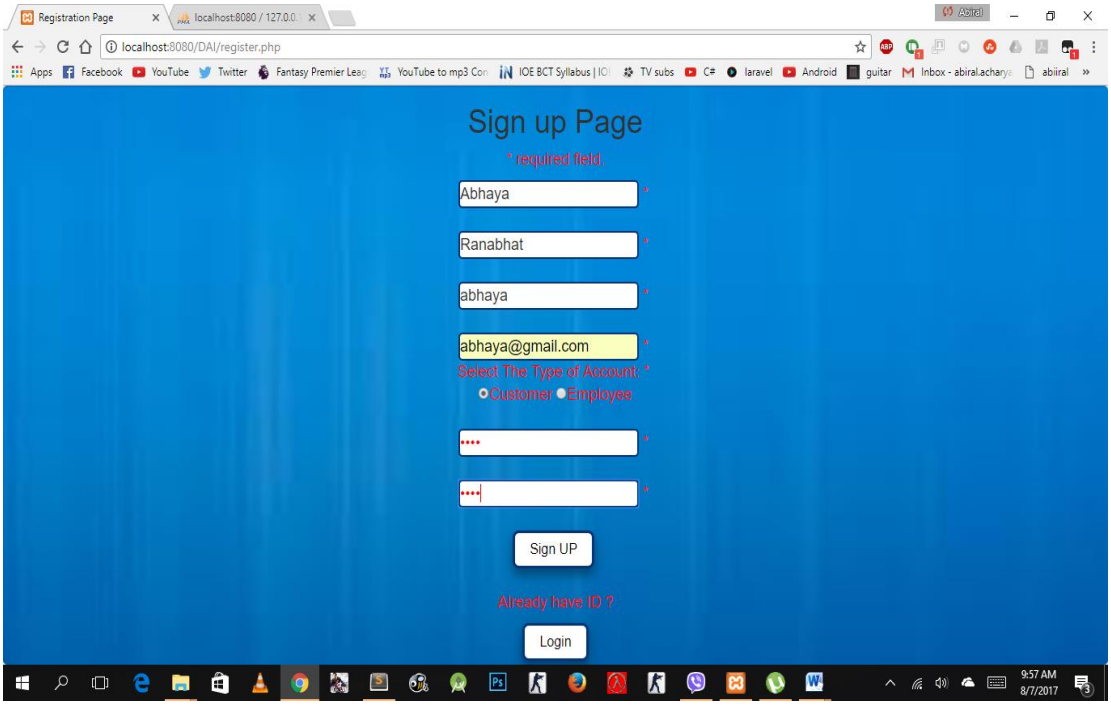


Figure 4.24: All fields are filled correctly and signed up

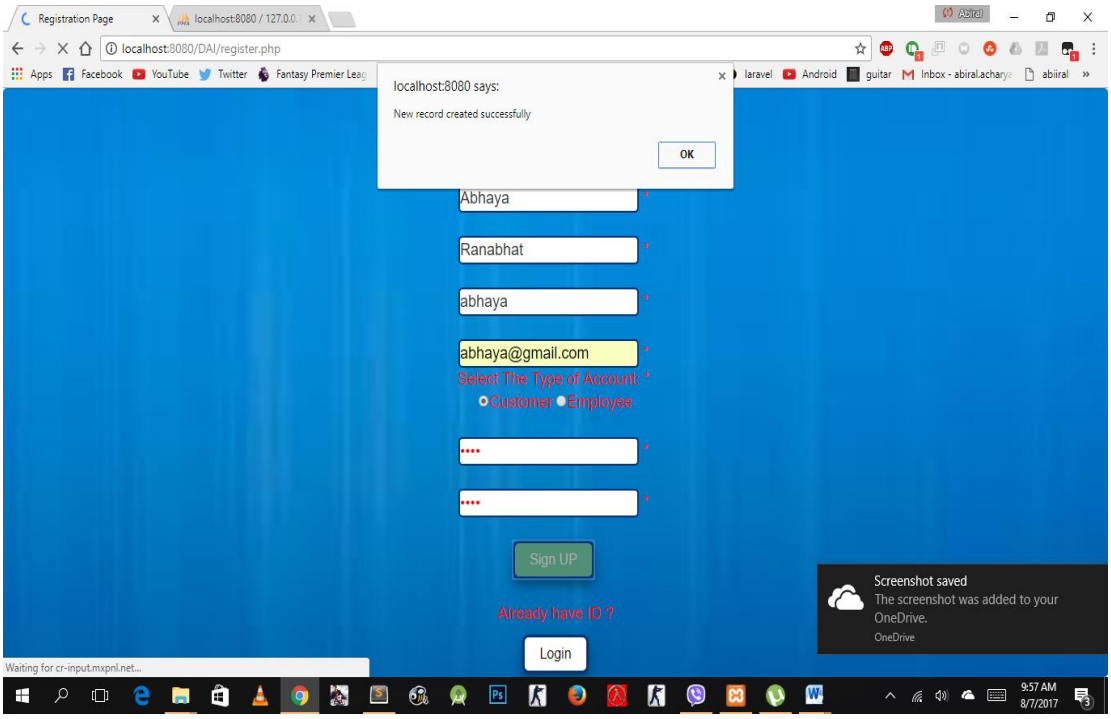


Figure 4.25: Sign up successful

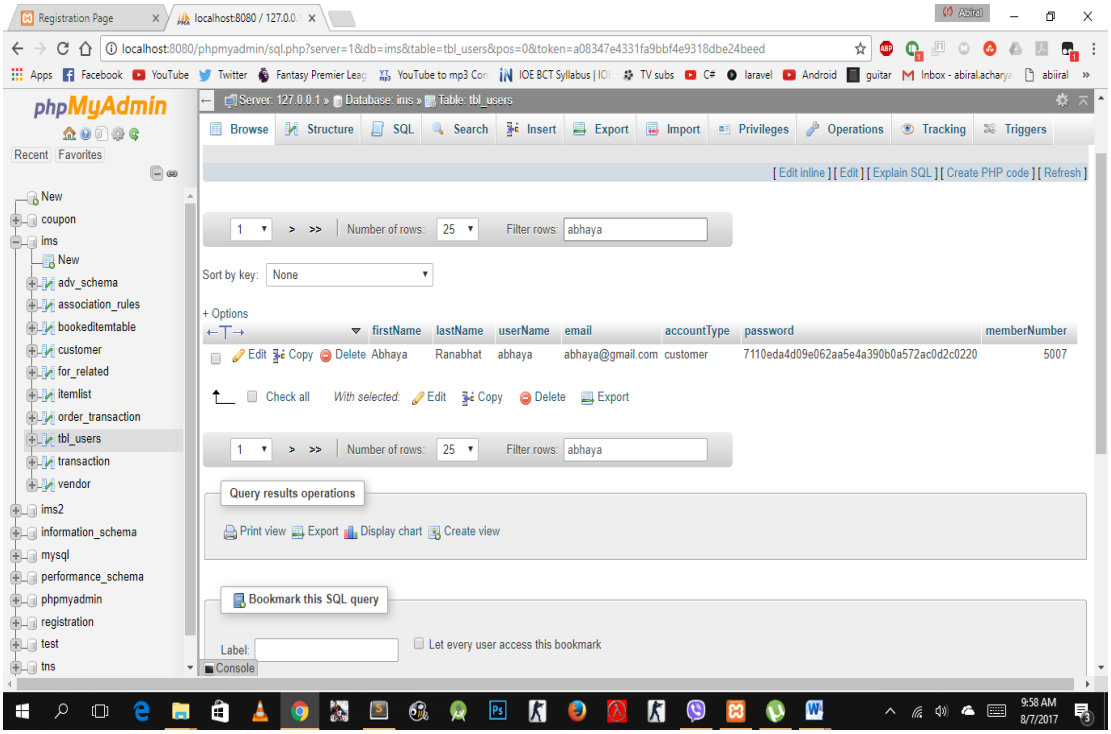


Figure 4.26: Database check

Test 3: To test with wrong username or/and password during login

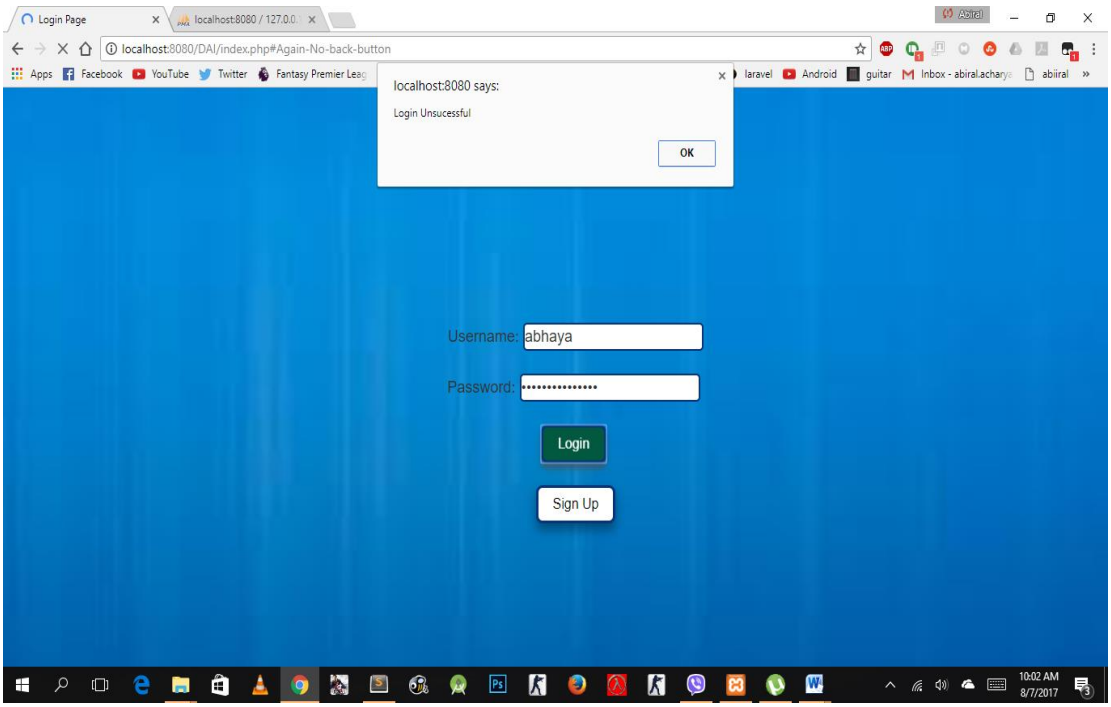


Figure 4.27: Wrong password for "abhaya"

Test 4: To test with correct username and password during login

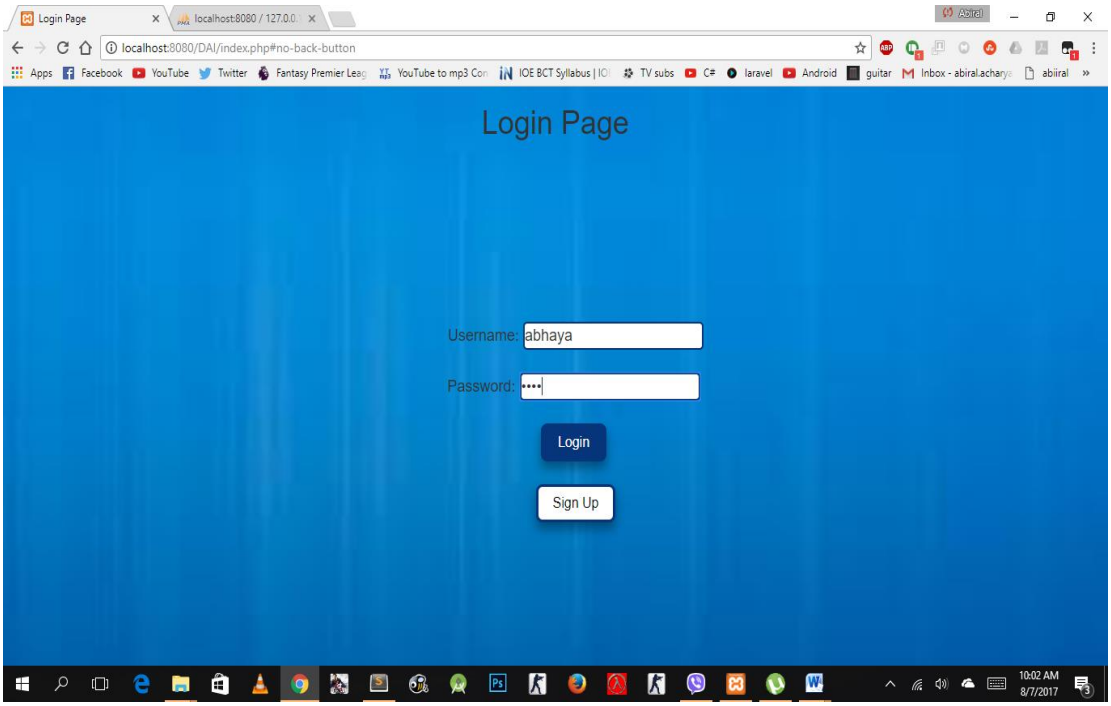


Figure 4.28 Correct password for "abhaya"

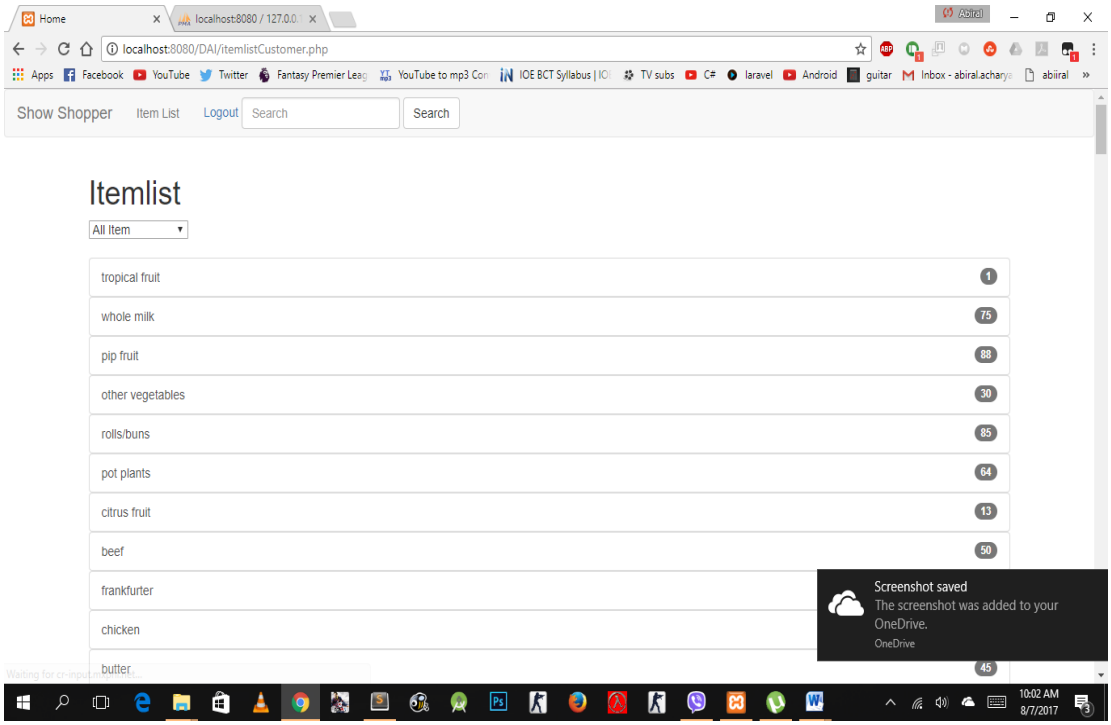


Figure 4.29: Logged in successfully

Test 5: To test whether the cart table is created as soon as the user logs in

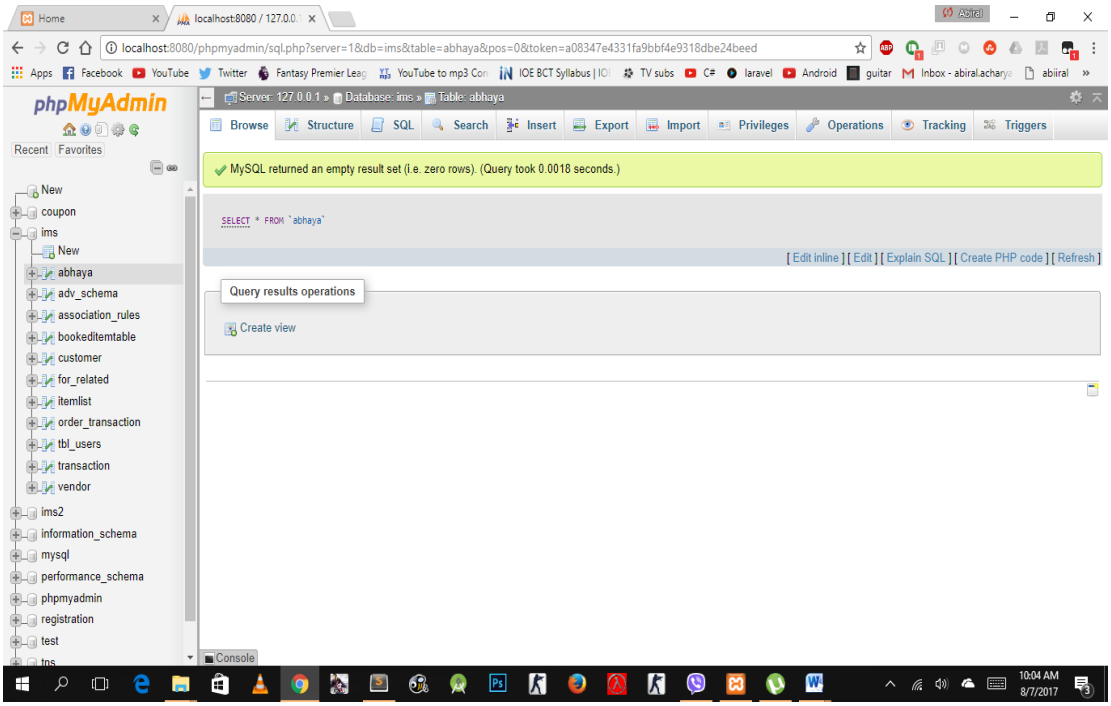


Figure 4.30: Cart created in database automatically after login

Test 6: To test add items to cart with quantity greater than remaining quantity

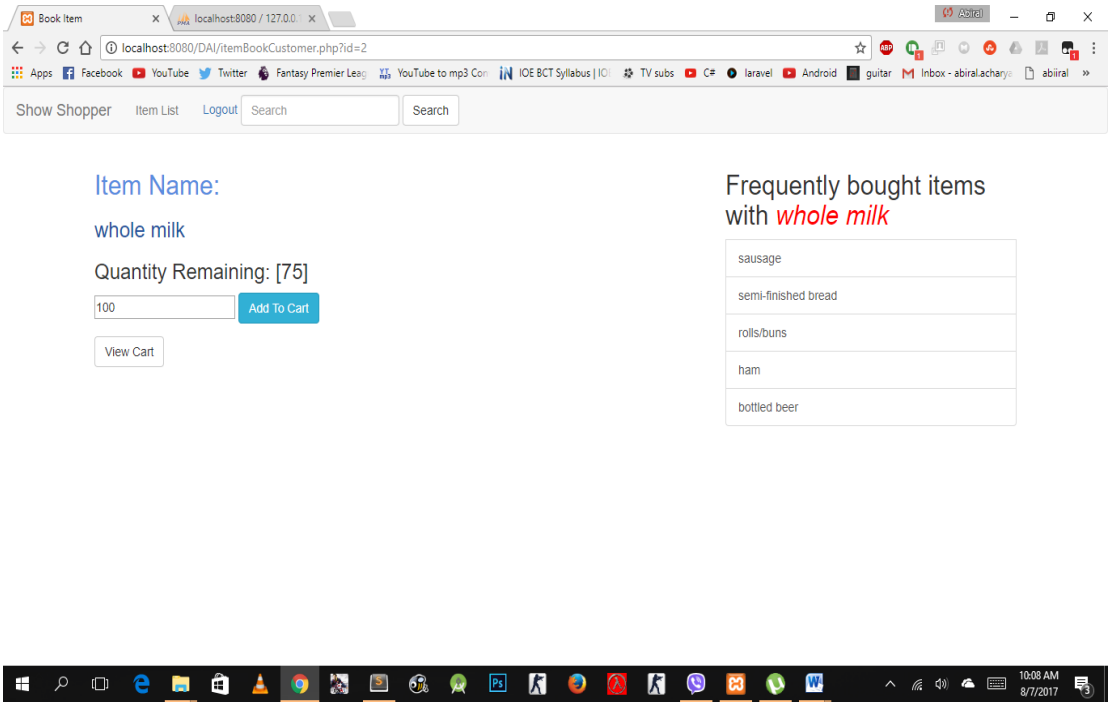


Figure 4.31: Trying to add 100 items (greater than available)

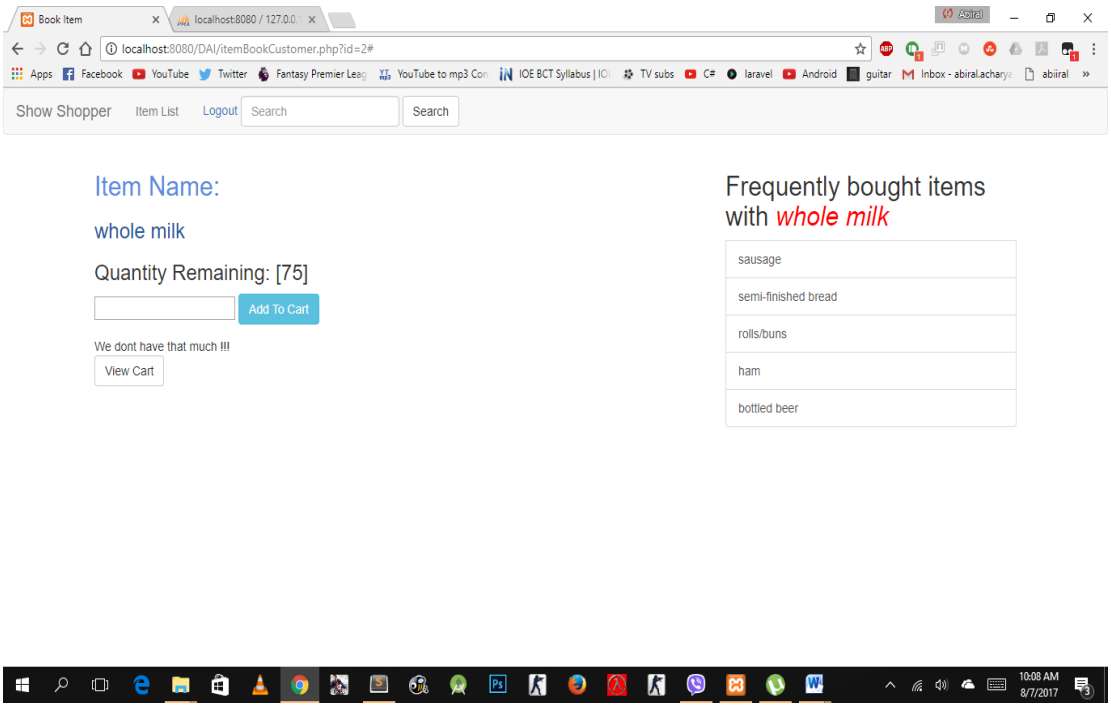


Figure 4.32: Error message

Test 7: To test whether the items with quantity are being add to cart and database

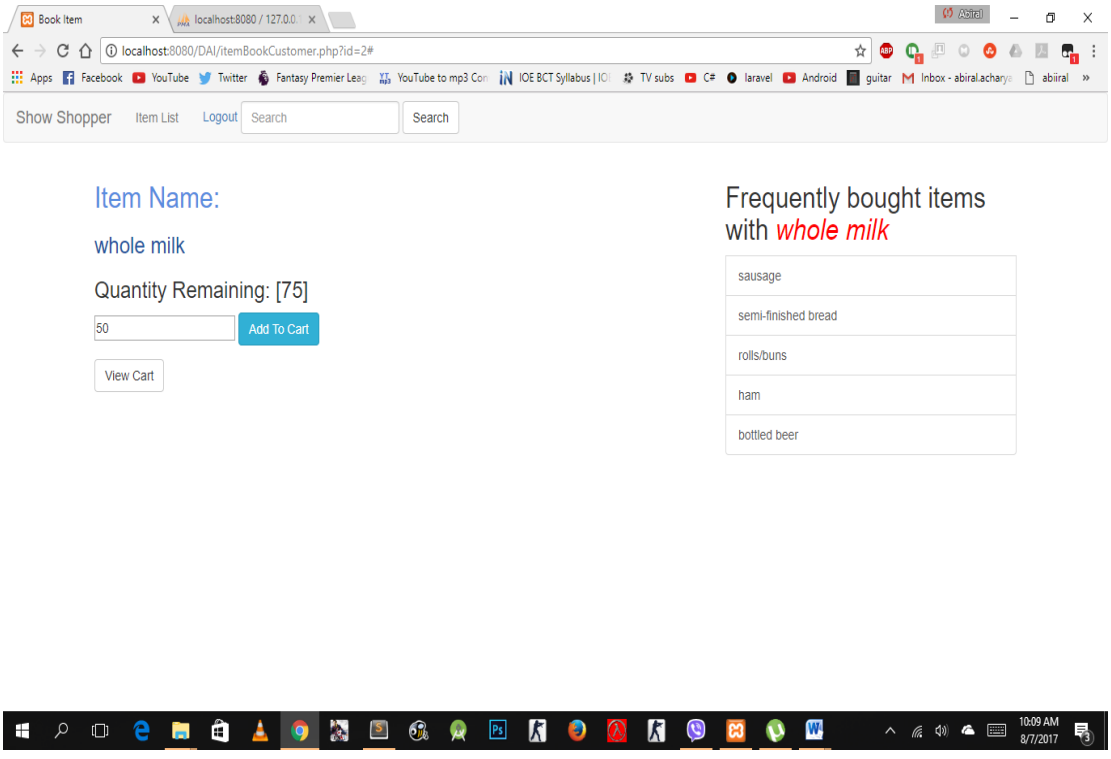


Figure 4.33: Trying to add 50 items (lower than available)

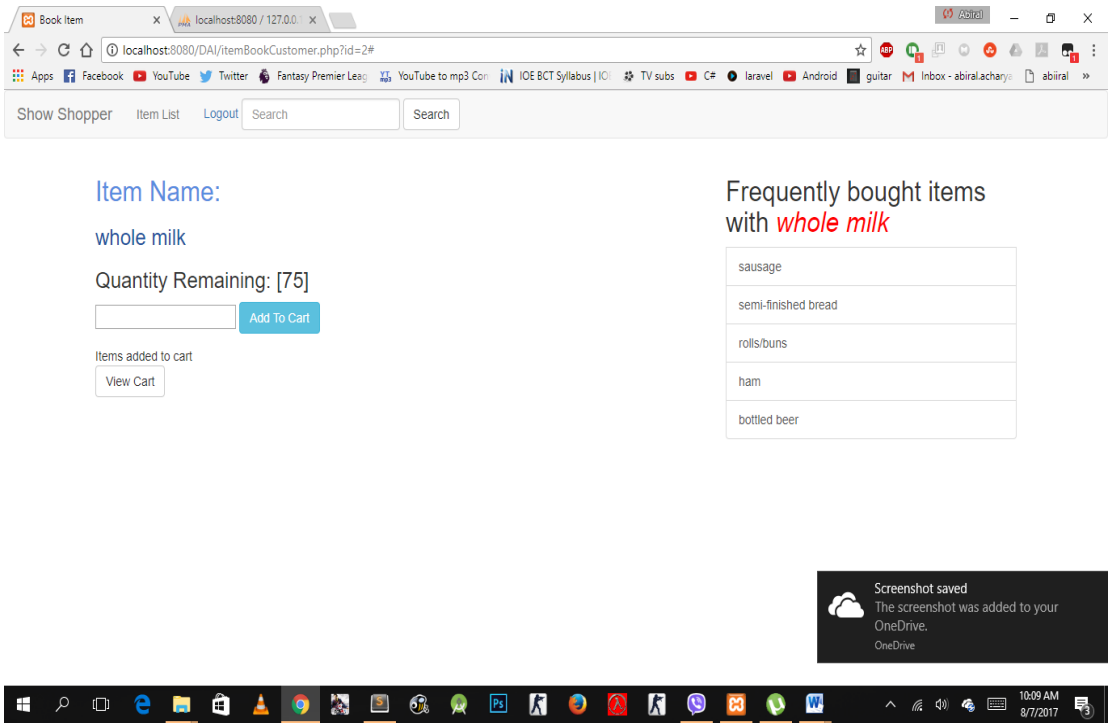


Figure 4.34: Success message

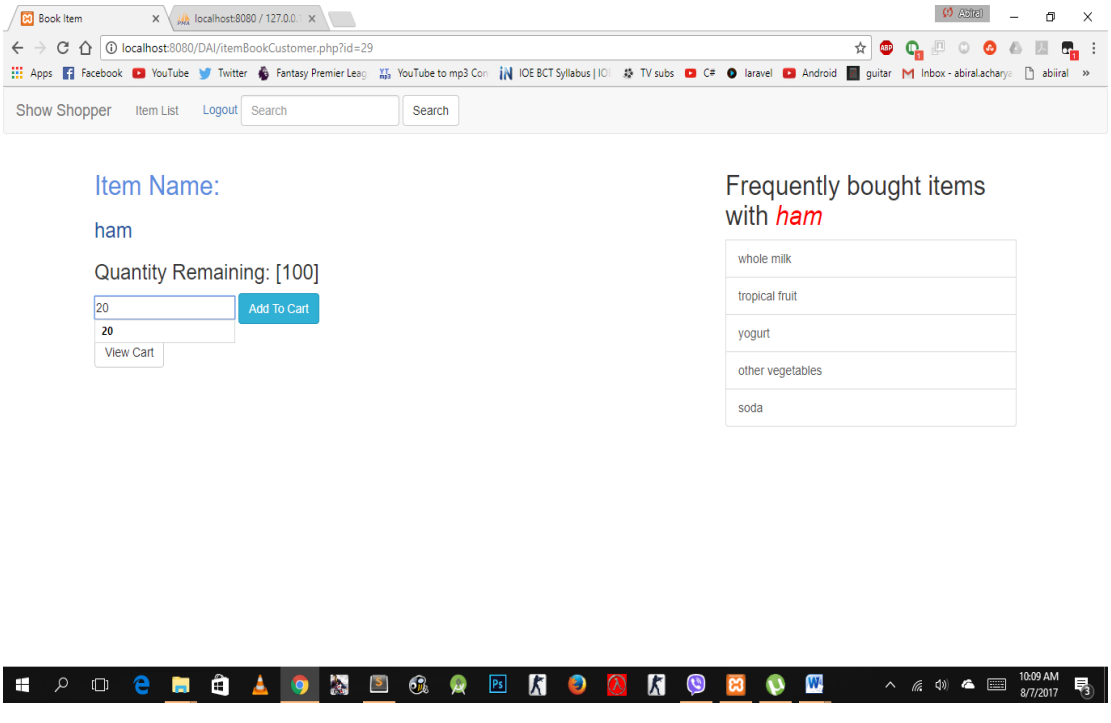


Figure 4.35: Adding other items (20 ham)

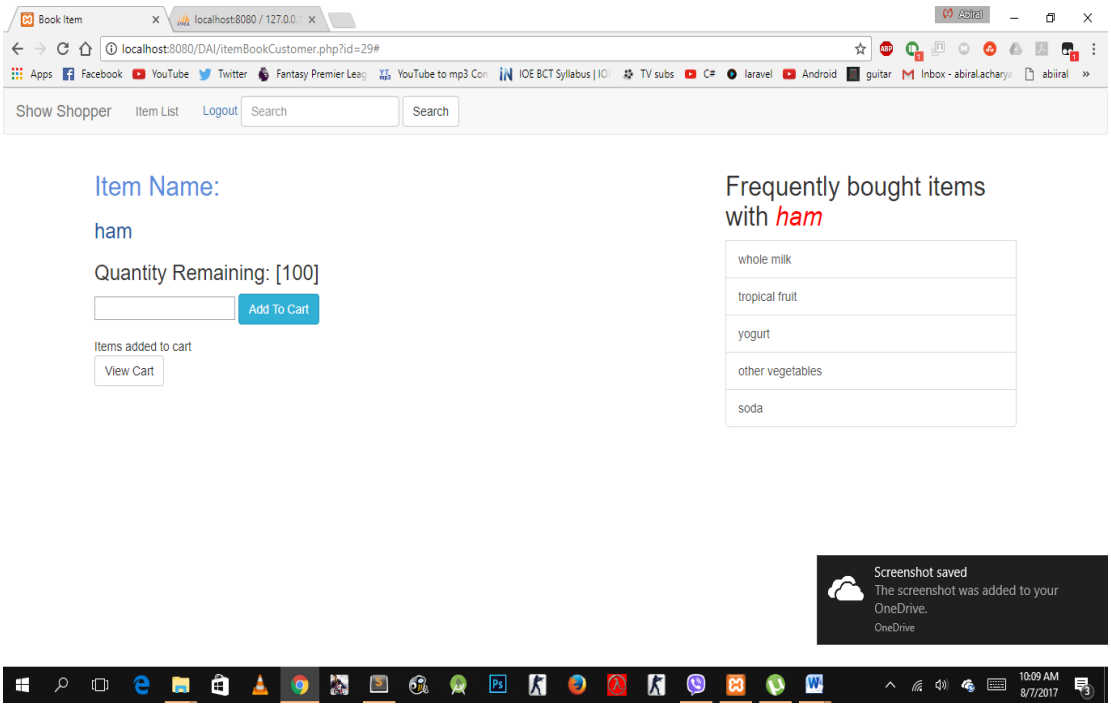


Figure 4.36: Ham added

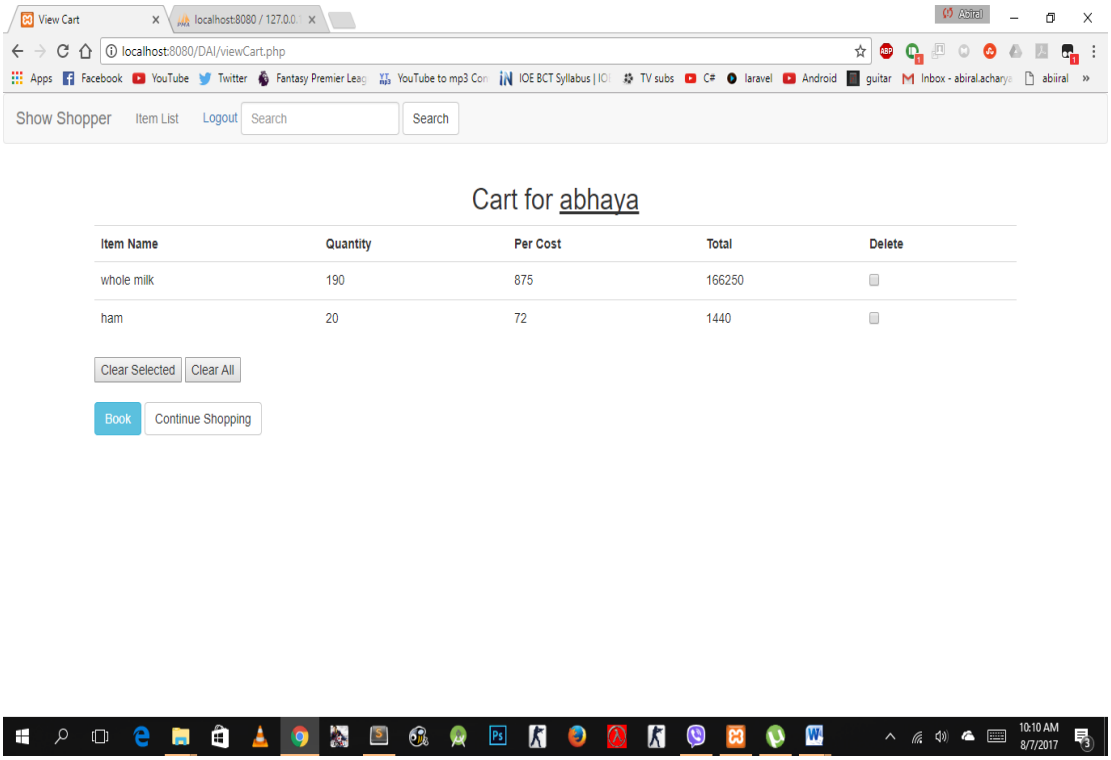


Figure 4.37: Check cart

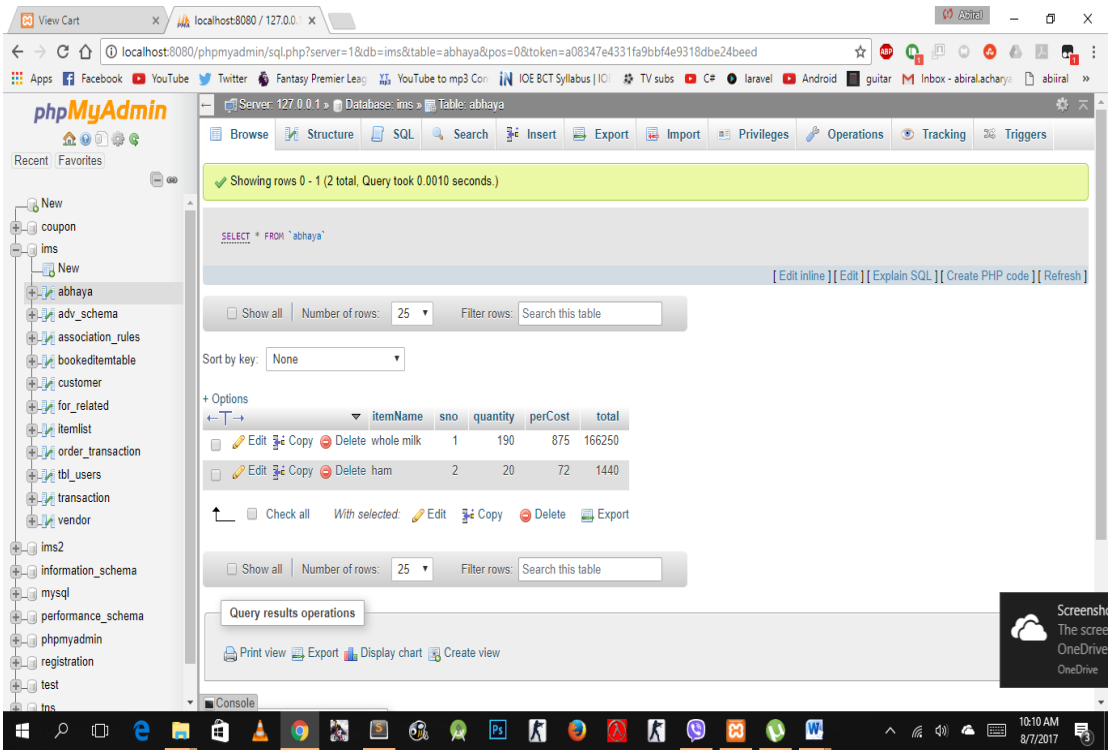


Figure 4.38: Check cart database

Test 8: To test empty field in quantity field

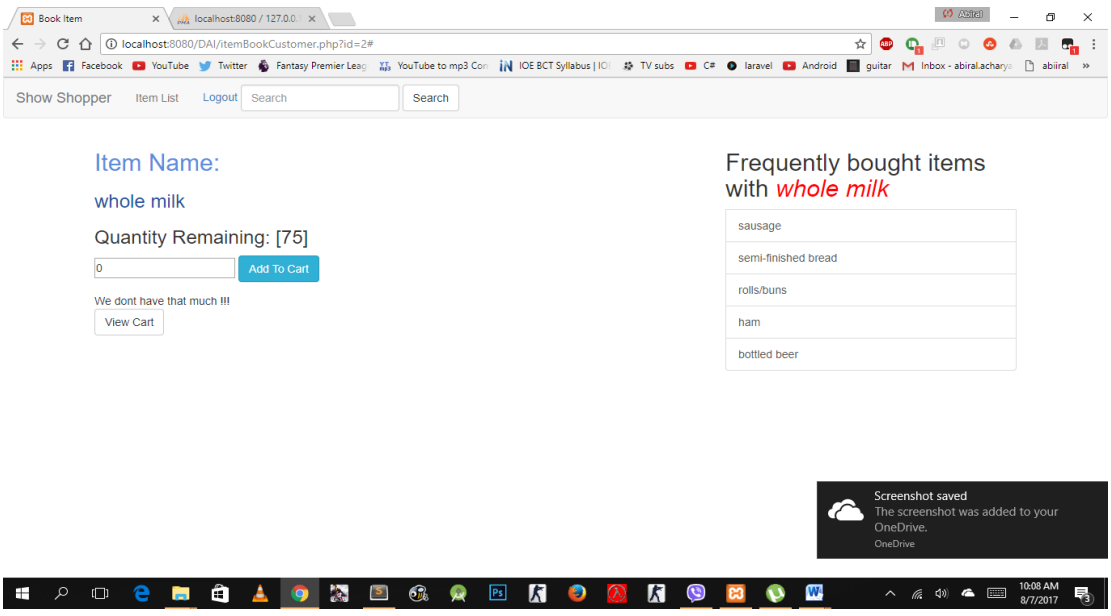


Figure 4.39: Trying to add 0 items

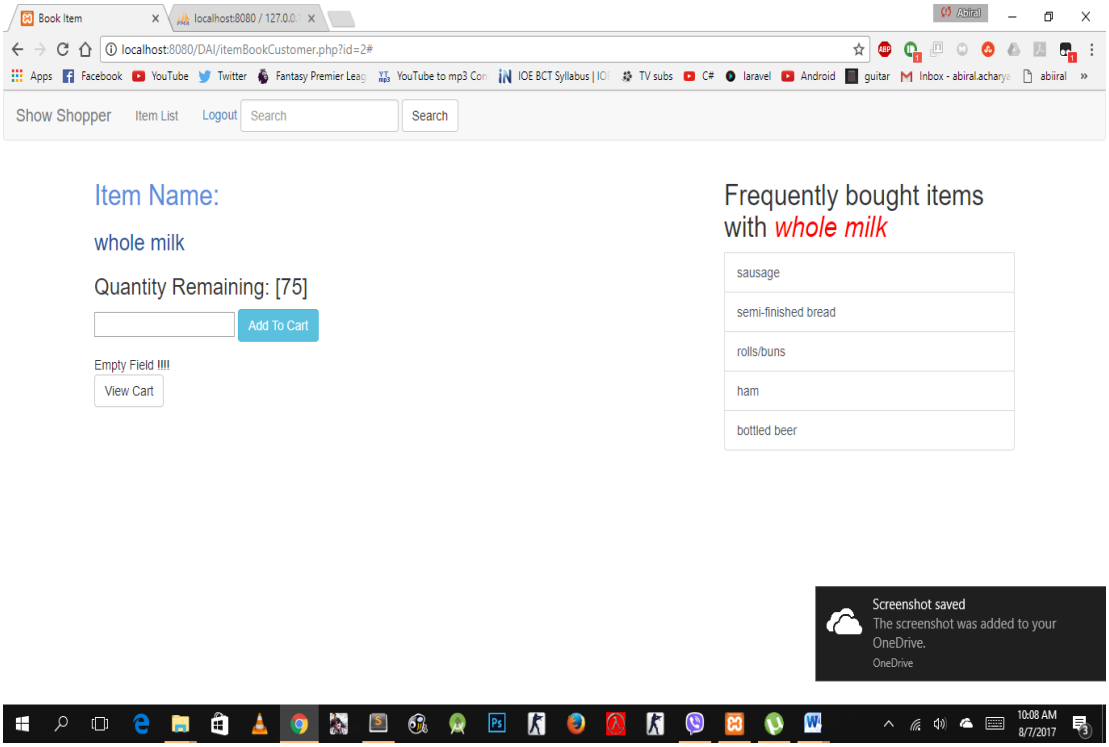


Figure 4.40: Error message

Test 9: To test by viewing empty cart

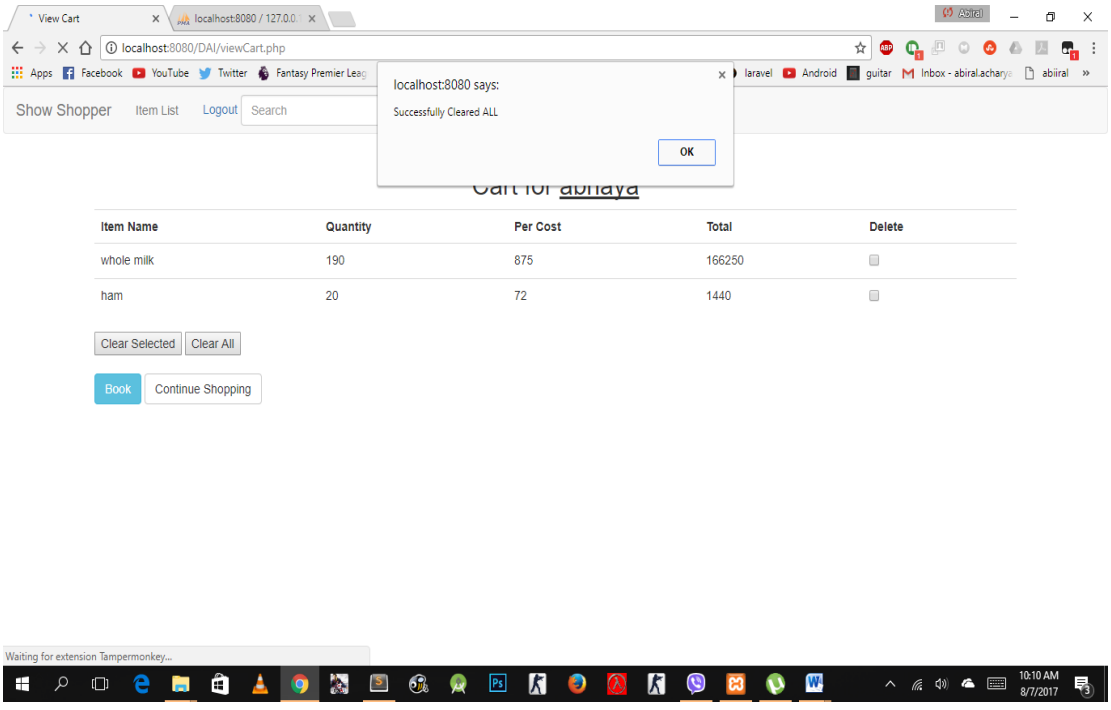


Figure 4.41: Clear all cart items

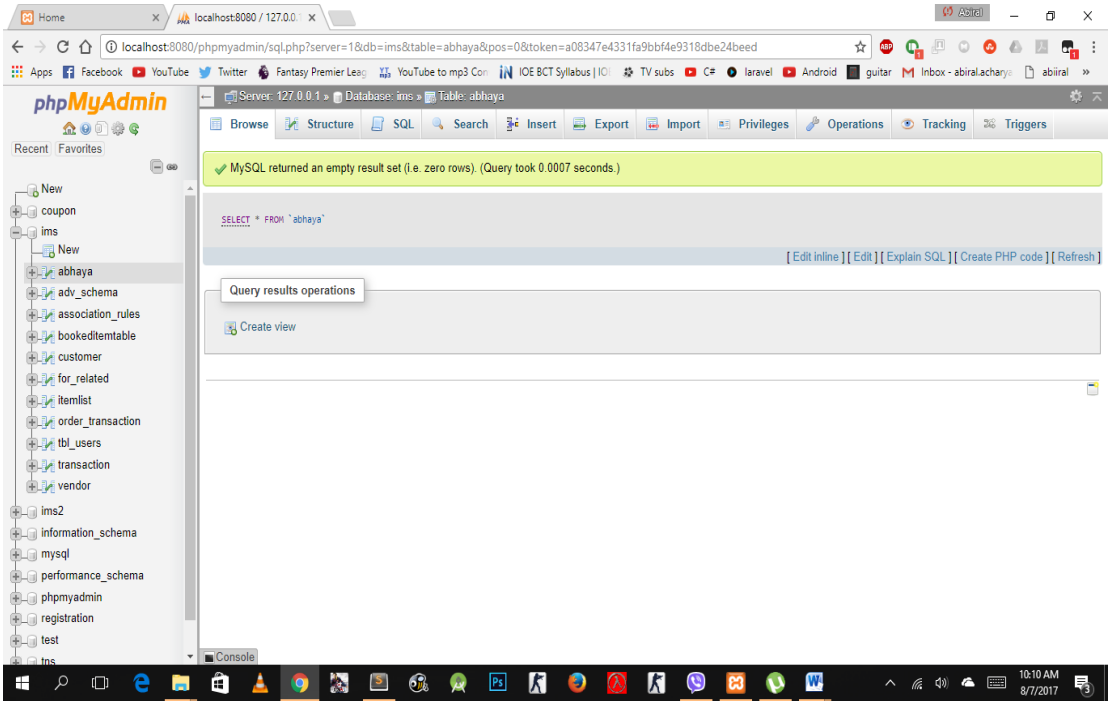


Figure 4.42: Check Database whether all items are cleared or not

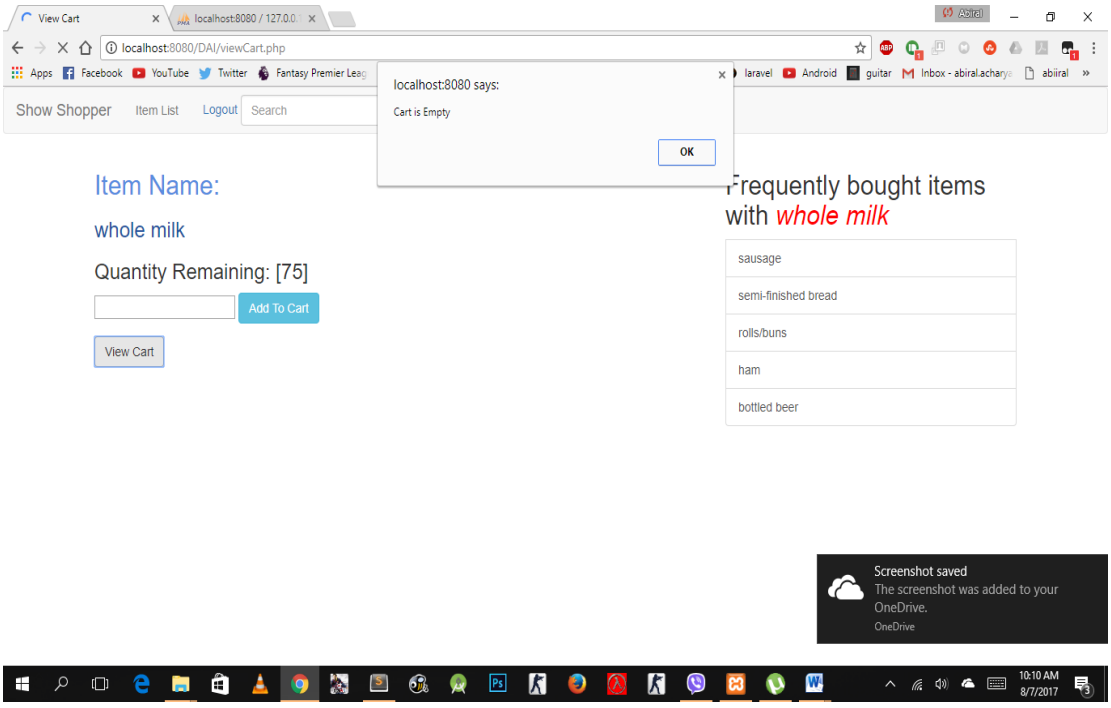


Figure 4.43: Try to view cart

Test 10: To test new member is divided into cluster

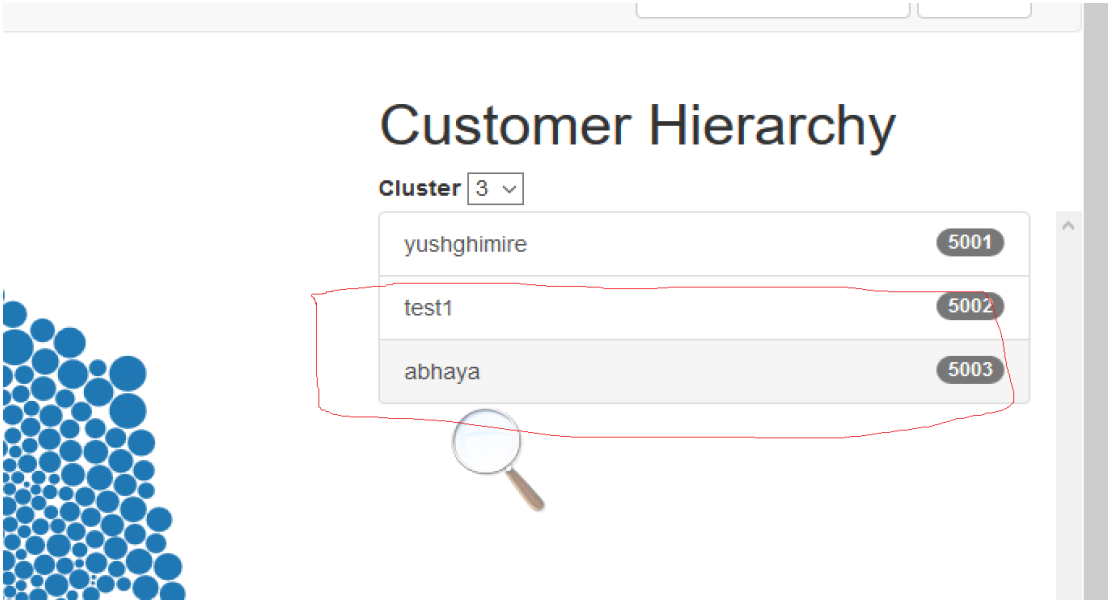


Figure 4.44: Abhaya in cluster 3

Test 11: To test whether prediction is made for new customer

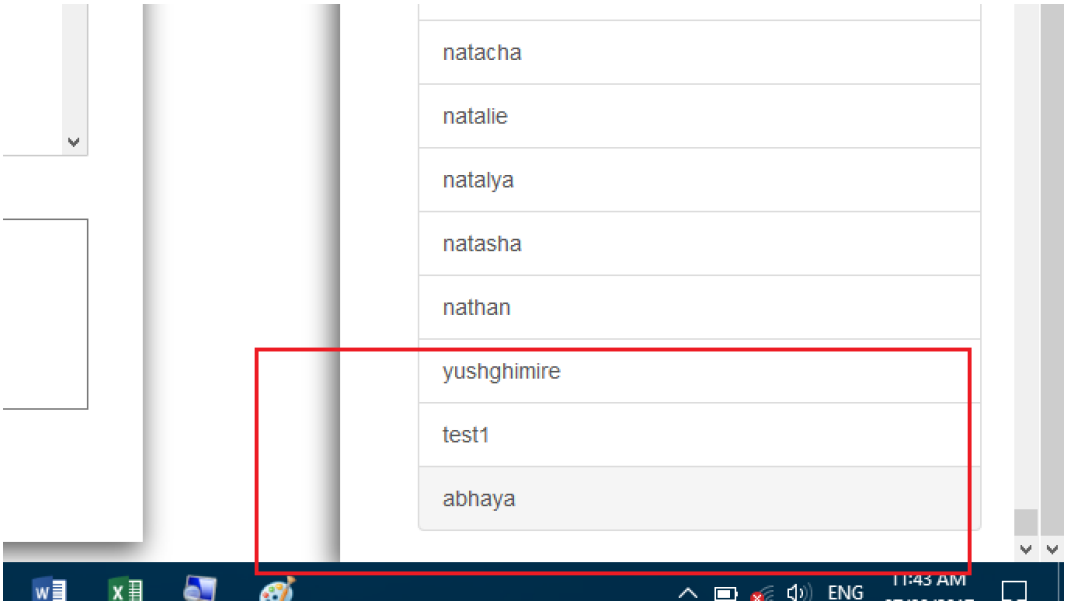


Figure 4.45: Abhaya's response is predicted as 'YES'

CHAPTER 5. EPILOGUE

5.1 Result

Inventory Management system: Basic functionalities of Inventory Management system is incorporated in our system. With basic security feature like login, item searching viewing, auto ordering and booking. Our system's MIS functionality is shown below:

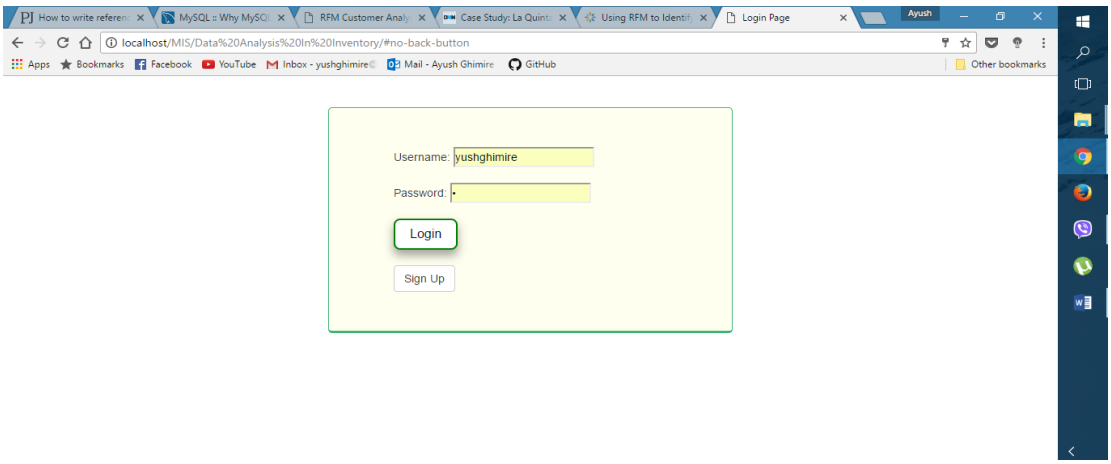


Figure 5.1: Login for employee and customer

Customer View:

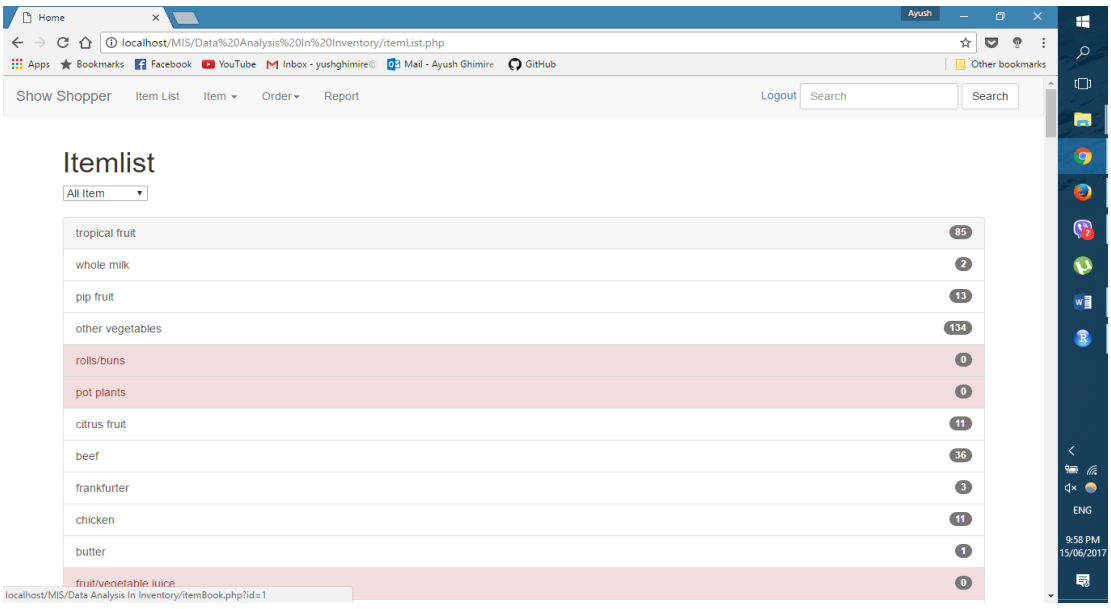


Figure 5.2: List of items

After the association rules are calculated customer were given recommendation for each item bought along with the item they have chosen. Below is the recommendation given to the customer who chose whole milk as the item in inventory.

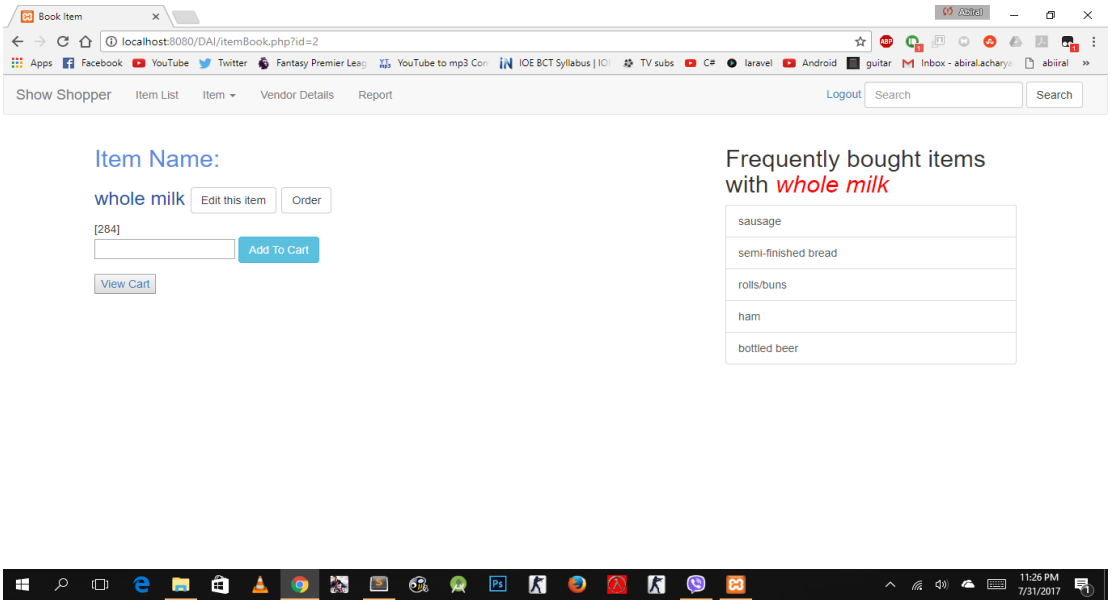


Figure 5.3: Add item to cart.

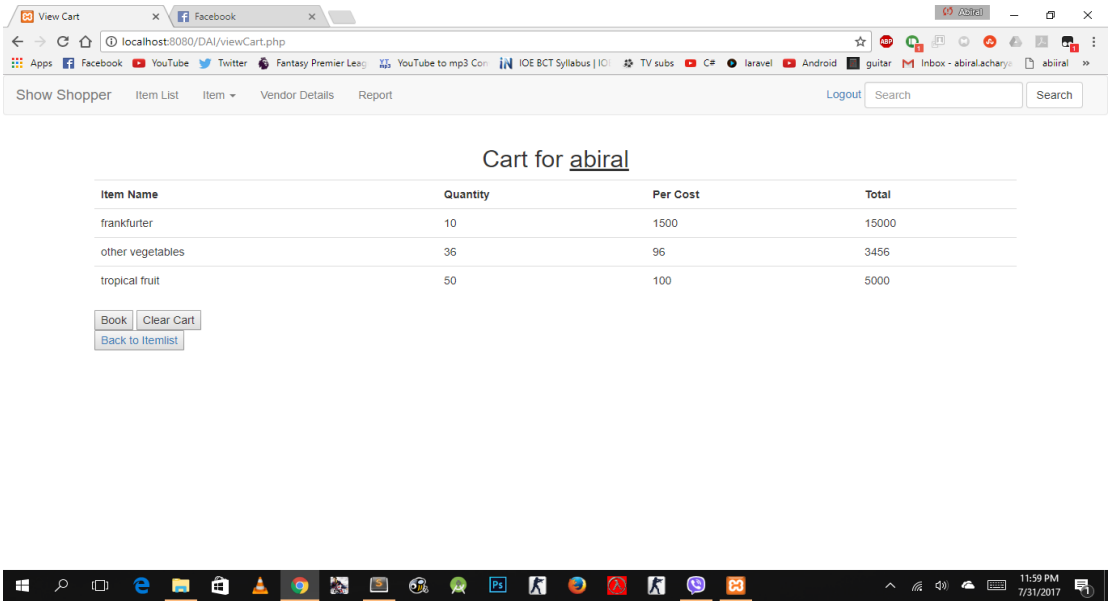


Figure 5.4: Item added in cart

Employee View:

Employee will be able to add new item in the database, which include these fields: item name, cost price, selling price and quantity.

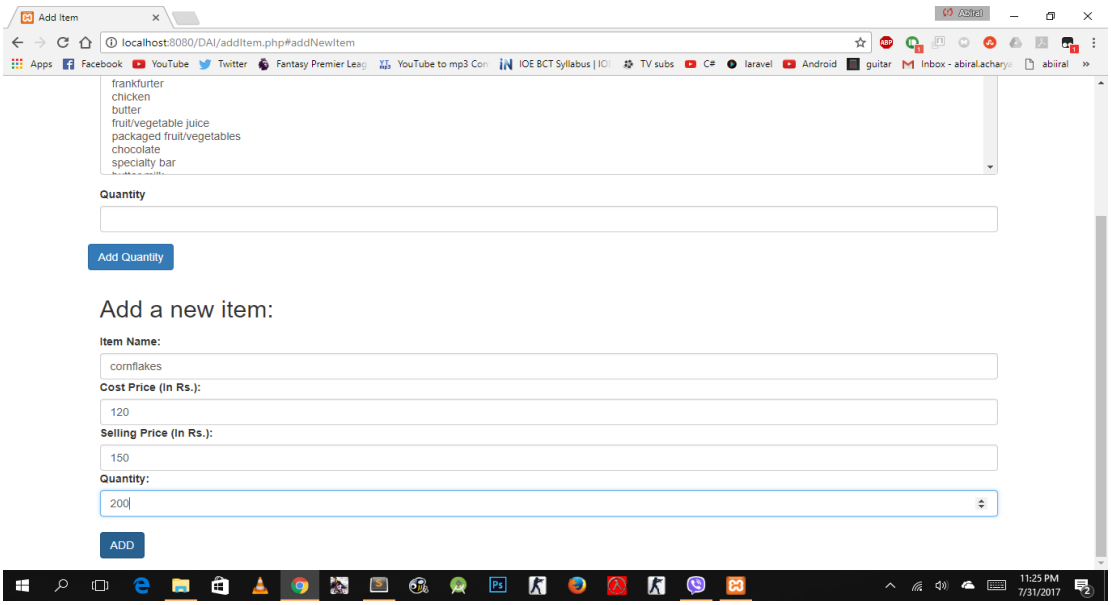


Figure 5.5: Add new item

Employee will be able to change item details, updating quantity, cost price and selling price of the item.

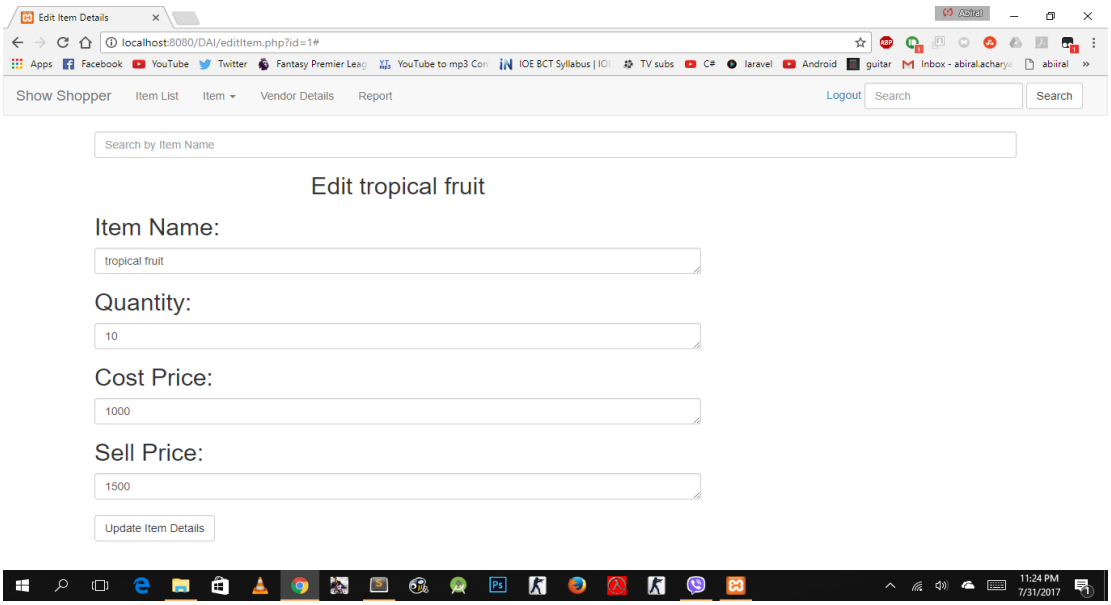


Figure 5.6: Edit items

When the item runs low in stock employee are able to order the item, from the selected vendors

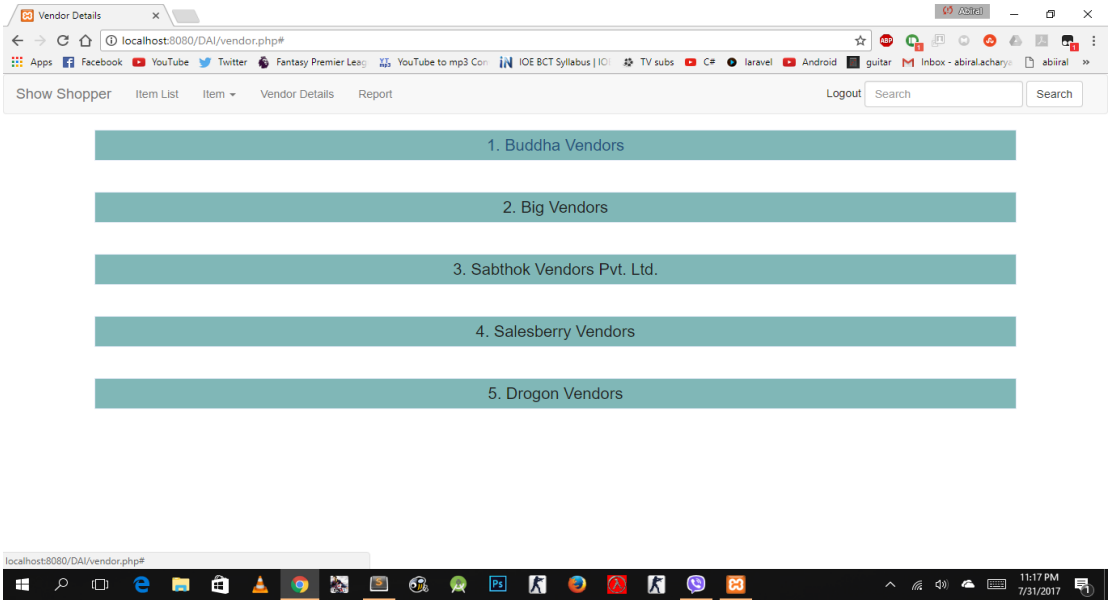


Figure 5.7: Vendors list

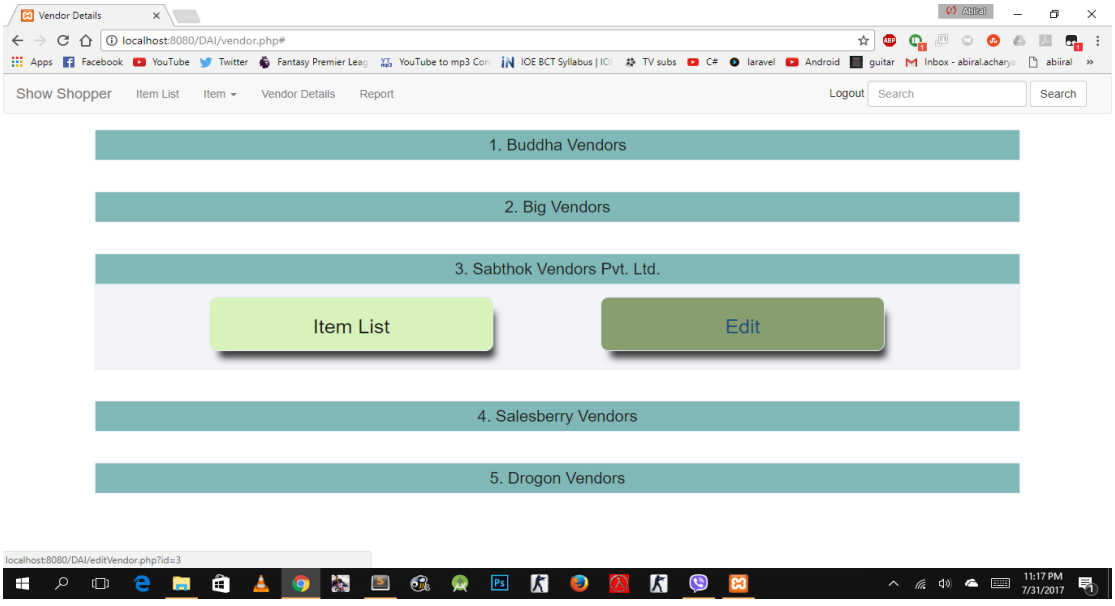


Figure 5.8: Vendor option

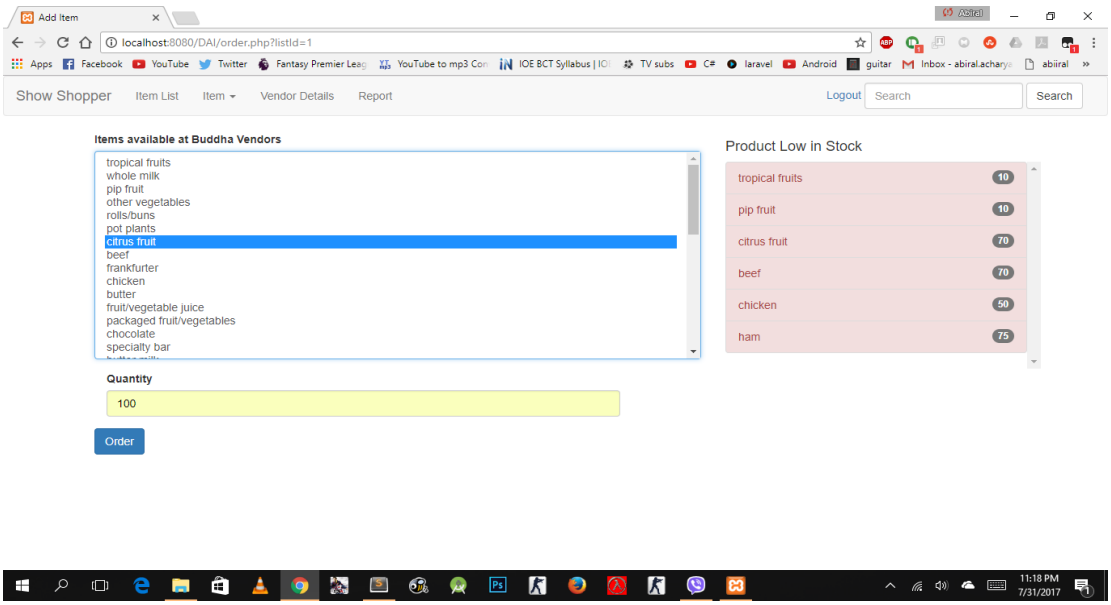


Figure 5.9: Item order

The employee of the system is also given a visual representation for the combination of itemset, so that they can have the knowledge of association for scheme development. Below we can see the visualization of top 25 itemset in terms of lift.

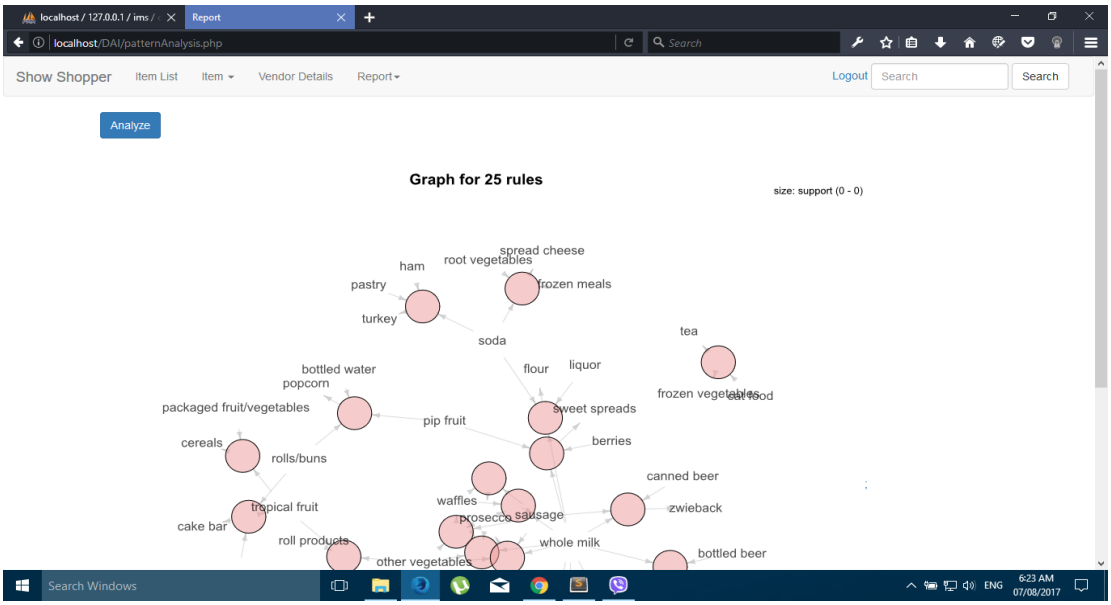


Figure 5.10: Visualization to the employee

Employee and Customer both can search for item using keywords Below is a search preform for ham.

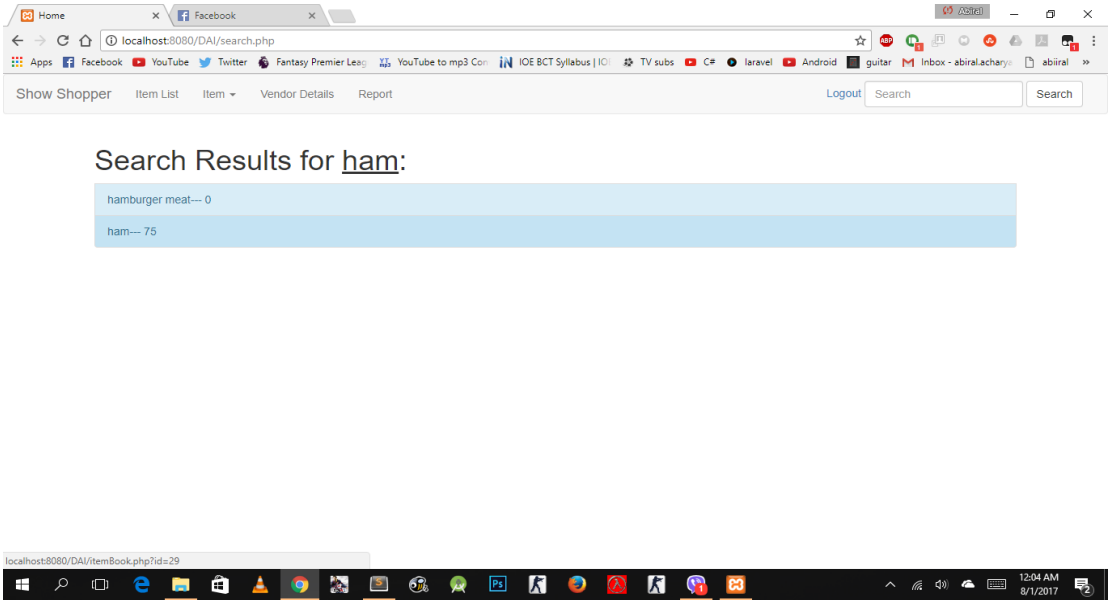
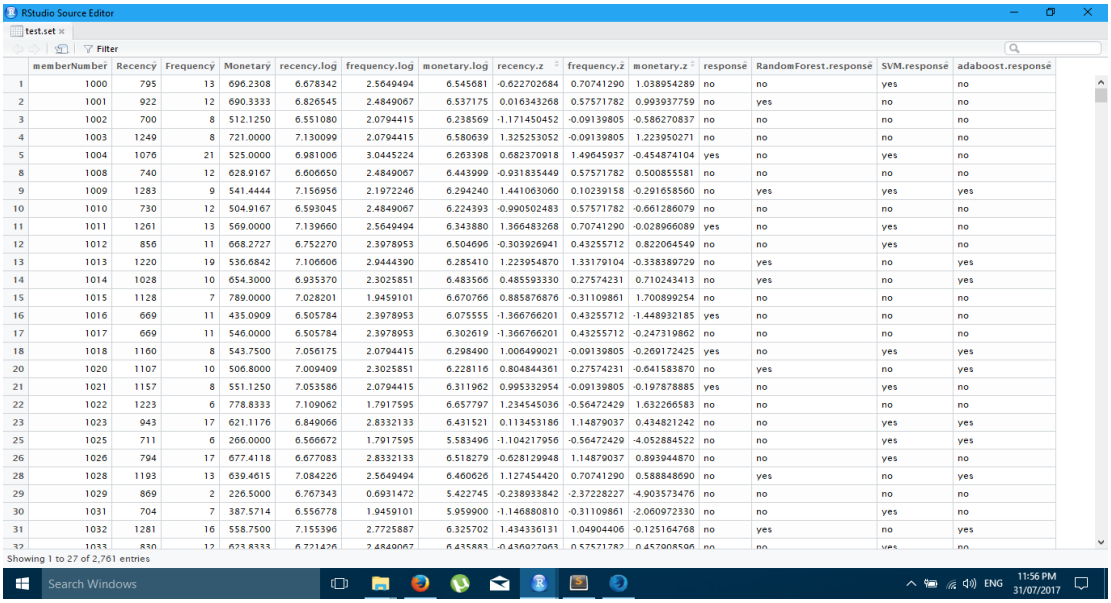


Figure 5.11: Search result

Customer Data Analysis and It's result:

Customer behavior on the basis of Recency, Frequency and monetary data are calculated as shown below. This analysis helps understand user in case of marketing.



	memberNumber	Recency	Frequency	Monetary	recency.log	frequency.log	monetary.log	recency.z	frequency.z	monetary.z	response	RandomForest.response	SVM.response	adaboost.response
1	1000	795	13	696.2308	6.678342	2.5649494	6.545681	-0.622702684	0.70741290	1.038954289	no	no	yes	no
2	1001	922	12	690.3333	6.826545	2.4849067	6.537175	0.016343268	0.57571782	0.993937759	no	yes	no	no
3	1002	700	8	512.1250	6.551080	2.0794415	6.238569	-1.171450452	-0.09139805	-0.586270837	no	no	no	no
4	1003	1249	8	721.0000	7.130099	2.0794415	6.580639	1.325253052	-0.09139805	1.223950271	no	no	no	no
5	1004	1076	21	525.0000	6.981006	3.0445224	6.263398	0.682370918	1.49645937	-0.454874104	yes	no	yes	no
8	1008	740	12	628.9167	6.606650	2.4849067	6.443999	-0.931835449	0.57571782	0.500855581	no	no	no	no
9	1009	1283	9	541.4444	7.156956	2.1972246	6.294240	1.441063060	0.10239158	-0.291658560	no	yes	yes	yes
10	1010	730	12	504.9167	6.593045	2.4849067	6.224393	-0.990502483	0.57571782	-0.661286079	no	no	no	no
11	1011	1261	13	569.0000	7.139660	2.5649494	6.343880	1.366483268	0.70741290	-0.028966089	yes	no	yes	no
12	1012	856	11	668.2727	6.752270	2.3978953	6.504696	-0.303926941	0.43255712	0.822064549	no	no	yes	no
13	1013	1220	19	536.6842	7.106606	2.9444390	6.285410	1.223954870	1.33179104	-0.338389729	no	yes	no	yes
14	1014	1028	10	654.3000	6.935370	2.3025851	6.483566	0.485593330	0.27574231	0.710243413	no	yes	no	yes
15	1015	1128	7	789.0000	7.028201	1.9459101	6.670766	0.885876876	-0.31109861	1.700899254	no	no	no	no
16	1016	669	11	435.0909	6.505784	2.3978953	6.075555	-1.366766201	0.43255712	-1.448932185	yes	no	no	no
17	1017	669	11	546.0000	6.505784	2.3978953	6.302619	-1.366766201	0.43255712	-0.247319862	no	no	no	no
18	1018	1160	8	543.7500	7.056175	2.0794415	6.298490	1.006499021	-0.09139805	-0.269172425	yes	no	yes	yes
20	1020	1107	10	506.8000	7.009409	2.3025851	6.228116	0.804844361	0.27574231	-0.641583870	no	yes	no	yes
21	1021	1157	8	551.1250	7.053586	2.0794415	6.311962	0.995332954	-0.09139805	-0.197878885	yes	no	yes	no
22	1022	1223	6	778.8333	7.109062	1.7917595	6.657797	1.234545036	-0.56472429	1.632266583	no	no	no	no
23	1023	943	17	621.1176	6.849066	2.8332133	6.431521	0.113453186	1.14879037	0.434821242	no	no	yes	yes
25	1025	711	6	266.0000	6.566672	1.7917595	5.583496	-1.104217956	-0.56472429	-4.052884522	no	no	yes	yes
26	1026	794	17	677.4118	6.677083	2.8332133	6.518279	-0.628129948	1.14879037	0.893944870	no	no	yes	no
28	1028	1193	13	639.4615	7.084226	2.5649494	6.460626	1.127454420	0.70741290	0.588848690	no	yes	no	yes
29	1029	869	2	226.5000	6.767343	0.6931472	5.422745	-0.238933842	-2.37228227	-4.903573476	no	no	no	no
30	1031	704	7	387.5714	6.556778	1.9459101	5.959900	-1.146880810	-0.31109861	-2.060972330	no	no	yes	no
31	1032	1281	16	558.7500	7.155396	2.7725887	6.325702	1.434336131	1.04904406	-0.125164768	no	yes	no	yes
32	1033	830	12	623.8333	6.721426	2.4849067	6.435883	-0.436927963	0.57571782	0.457908596	no	no	yes	no

After a model is trained using the train set, testing set is used to know the accuracy of our system. On performing predicitve analysis, accuracy percentage is calculated. The percentage specifies the effectiveness of using the algorithm to predicit the behaviour of customer. Random forest predicts with 68.6% of accuracy.

```
+                  reference=test.set$response)
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1750  634
       yes  224  131

               Accuracy : 0.6867
                 95% CI : (0.669, 0.7041)
    No Information Rate : 0.7207
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0691
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.8865
            Specificity : 0.1712
         Pos Pred Value : 0.7341
         Neg Pred Value : 0.3690
             Prevalence : 0.7207
         Detection Rate : 0.6389
   Detection Prevalence : 0.8704
      Balanced Accuracy : 0.5289

       'Positive' Class : no
```

Figure 5.13: Random Forest analysis

Support Vector Machine with 61.8% of accuracy.

```
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1505  576
       yes  469  189

               Accuracy : 0.6185
                 95% CI : (0.6, 0.6367)
    No Information Rate : 0.7207
    P-Value [Acc > NIR] : 1.000000

                  Kappa : 0.0099
 Mcnemar's Test P-Value : 0.001042

            Sensitivity : 0.7624
            Specificity : 0.2471
         Pos Pred Value : 0.7232
         Neg Pred Value : 0.2872
             Prevalence : 0.7207
         Detection Rate : 0.5495
   Detection Prevalence : 0.7598
      Balanced Accuracy : 0.5047

       'Positive' Class : no
```

Figure 5.14: Support Vector Machine

Fast AdaBoost with 67.7% of accuracy.

```
+                  reference=test.set$response)
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1630  592
       yes  344  173

               Accuracy : 0.6583
                 95% CI : (0.6402, 0.676)
    No Information Rate : 0.7207
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0576
 Mcnemar's Test P-Value : 6.834e-16

            Sensitivity : 0.8257
            Specificity : 0.2261
         Pos Pred Value : 0.7336
         Neg Pred Value : 0.3346
             Prevalence : 0.7207
         Detection Rate : 0.5951
   Detection Prevalence : 0.8112
      Balanced Accuracy : 0.5259

       'Positive' Class : no
```

Figure 5.15: Support Vector Machine

After finding the data predicted with each model where different in about 15 datasets, i.e. dataset that are wrongly predicted as No even though they are actual a Yes responder, we combined the result of all the model to increase the prediction of Yes. Result of all the system is ORed, this decrease the accuracy but minimize the wrongly prediction of customer who are likely to respond to our call.

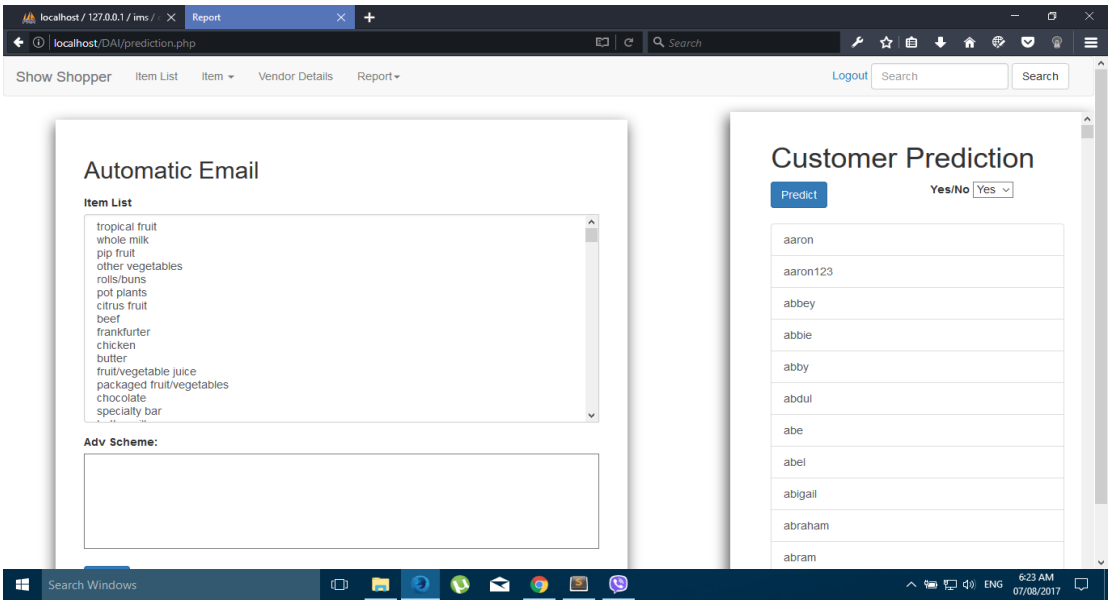


Figure 5.16: List of customers predicted as non-responder

As shown in above figure a list is presented to the user of the non-Responder customer, that helps in identifying the customers and minimizes the expenditure in market advertisement.

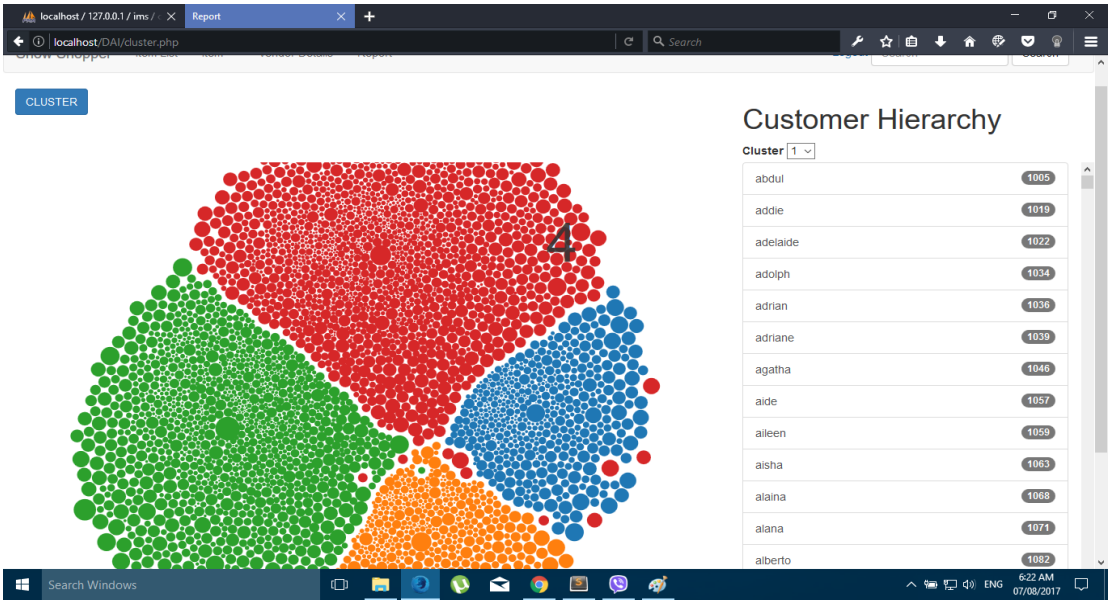


Figure 5.17: Visualization of customers classification

All the result is OR-ed to include all the customer who will be responding to the system. Though the efficiency will decrease to 64.1%, all the probable customers are included.

```
+                reference=df_item$response)
Confusion Matrix and Statistics

          Reference
Prediction   no  yes
       no  1889  426
       yes  971  613

               Accuracy : 0.6417
                 95% CI : (0.6264, 0.6568)
    No Information Rate : 0.7335
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2146
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.6605
            Specificity : 0.5900
         Pos Pred Value : 0.8160
         Neg Pred Value : 0.3870
             Prevalence : 0.7335
         Detection Rate : 0.4845
   Detection Prevalence : 0.5937
      Balanced Accuracy : 0.6252

       'Positive' Class : no
```

Figure 5.18: Response result after ORing

5.2 Problem Encountered

Low percentage of accuracy: to raise the accuracy bar, we detected the outliers and removed them which increased the accuracy by 7%.

High false negative value: Random Forest as predictive model resulted in high false negative value so we decided to experiment with other predictive models. Still all algorithms produced similar accuracy. Then we noticed 15-20 data set were differently predicted on each of the algorithms. Since, rightly predicting of the responder is more important than wrongly predicting non-responder we decided to OR each response of the models. This step increased the number of customer who might positively respond to our offer mails.

5.3 Future Enhancement

- Further demographic attributes of the customers can be added for analysis which will increase the accuracy of classification.
- In future customer will be able to rate the products so we can improve our recommendation system by collaborative filtering and contain based filtering.

5.6 Discussion

Incremental model was followed with each enhancement being added iteratively. We used a real world open source data set to perform data analysis. The data set were first cleaned and further pre-processing was done. The data set were processed into basket form which was used for association analysis (Apriori Algorithm). The item-item relationship allowed us to find items that were frequently bought together. This item result was recommended to the customer and visualize to the employee.

Customer transaction was processed to produce RFM data. From RFM analysis nature of customer was studied and classified. Customers were ranked and given priority from their RFM analysis results. Also, probable customer who would respond to the system recommendation were predicted. This prediction of customer behavior to the market recommendation will decrease the revenue lost in advertisement purpose.

5.7 Conclusion

Using programming tools such as PHP, R HTML and CSS we developed a simple inventory management system. We used the data of the system for RFM analysis and predicted the nature of the customer. Also, an item recommendation feature was added in the inventory management.

REFERENCE

[1] © Albion Research Ltd. 2017. *What is Market Basket Analysis?* [Online]. Available from: http://www.albionresearch.com/data_mining/market_basket.php [Accessed: 14th June, 2017].

[2] Joao Correia. *How RFM Analysis Boosts Sales*. [Online]. 2016 . Available from: http://www.blastam.com/blog/rfm-analysis-boosts-sales [Accessed: 14th June, 2017].

[3] © GainInsights Solutions. *Customer Segmentation Using RFM Analysis*. [Online]. 2014. Available from: http://gain-insights.com/solutions/retail-analytics/ customer-segmentation-using-rfm-analysis/ [Accessed: 14th June 2017]

[4] John Tukey. *The Future of Data Analysis*. [Online]. Princeton University. July 1961. Available from: http://projecteuclid.org/download/pdf_1/euclid.aoms/ 1177704711 [Accessed: 25th March 2016].

[5] Richard A. Becker, John M. Chambers and Allan R. Wilks. *The New S Language*. New York . Chapman & Hall. 1988. This book is often called the "Blue Book".

[6] John M. Chambers and Trevor J. Hastie eds. *Statistical Models in S*. New York. Chapman & Hall. 1992 This is also called the "White Book".

[7] John M. Chambers. *Programming with Data*. New York. . Springer . 1998 This is also called the "Green Book".

[8] Michael Hahsler, Christian Buchta, Bettina Gruen, Kurt Hornik,Ian Johnson, Christian Borgelt. *Mining Association Rules and Frequent Itemsets*.[Online]. 2017 Available from: http://mhahsler.github.io/arules/, http://lyle.smu.edu/IDA/arules [Acessed: 14th June 2017]

[9] Pazaras Christos. *DATA PREPARATION AND PREPROCESSING*

FOR DATA MINING USING R. Alexander Technological Educational Institute of Thessaloniki. 2013.

[10] Sridhar Mutyala. *Using RFM to Identify Your Best Customers.*[Online]. 2017. Available from: http://www.eightleaves.com/2011/01/using-rfm-to-identify-your-best-customers [Accessed: 14th June 2017]

[11] Vinod Venkatraman, Angelika Dimoka, Paul A. Pavlou, Khoi Vo, William Hampton, Bryan Bollinger, Hal E. Hershfield, Masakazu Ishihara, and Russell S.

Winer. 2015. *Predicting Advertising Success Beyond Traditional Measures*: New Insights from Neurophysiological Methods and Market Response Modeling. Retrieve from: http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-187.html [Accessed: 16th July,2017]

[12] Andrew Oliver Hatch. December 18, 2006. *Kernel Optimization for Support Vector Machines: Application to Speaker Verification*. [Online]. Available from: http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-187.html [Accessed: 14th July, 2017]

[13] Cuong-Nguyen, HaNam-Nguyen, Wang Yong. *Improving Classification Accuracy in Random Forest by Using Feature Impurity and Bayesian Probability*. Retrieved from: http://worldcomp-proceedings.com/proc/p2011/ICA3716.pdf

[14] © The R Foundation. *What is R?.*[Online]. Available from: https://www.r-project.org/about.html [Accessed: 13th March,2016].

[15] © 2001-2017 The PHP Group. *What is PHP?*. [Online]. Available from: http://php.net/manual/en/intro-whatis.php [Accessed:15th March,2016].

[16] Bostock, M. (2017). D3.js - Data-Driven Documents. D3js.org. Retrieved 4 August 2017, Available from: https://d3js.org/[Accessed: 14th June 2017]

[17] © Oracle Corporation and/or its affiliates. *Why MySQL?*. [Online]. Avaliable from: https://www.mysql.com/why-mysql/ [Accessed: 27th March 2016].

[18] Jack Han. *RFM Customer Analysis with R Language.* [Online]. Available from: http://www.dataapple.net/?p=84 [Accessed: 13th June 2017]

[19] Jennifer Thompson. *Random Forest.*[Online]. Available from: http://www.statsoft.com/Textbook/Random-Forest

[20]: Aylien Noel Bambrick. *Support Vector Machine[Online]*. Available from: http://www.kdnuggets.com/2016/07/support-vector-machines-simple-explanation.html

[21]: James McCaffery. *AdaBoost*[Online]. Available from: https://msdn.microsoft .com/en-us/magazine/dn166933.aspx

BIBLIOGRAPHY

- TechTarget. *Data.* [Online]. Available from: http://searchdatamanagement. techtarget.com/definition/data. [Accessed: 1st April 2016].
- © Retalon 2016. *PREDICTIVE ANALYTICS TRANSFORMS INVENTORY MANAGEMENT IN RETAIL*. [Online]. Available from: http://retalon.com /news-updates/predictive-analytics-transforms-inventory-management-in-retail [Accessed Date: 14th June 2017].
- w3schools.com. *PHP 5 Tutorial.* [Online]. Available from: https:// www.w3schools.com/php/default.asp [Accessed: 23th March 2016].
- Thinking inside the box. *#7: C++14, R and Travis — A useful hack.* [Online]. Available from: https://www.r-bloggers.com/ [Accessed: 24th May 2017].
- Anish. *RFM Analysis For Successful Customer Segmentation.* [Online]. 2017. Available from: http://www.putler.com/rfm-analysis/ [Accessed: 14th June 2017]
- © 2013-2017 MastersInDataScience.org. *Data Science in Retail*. [Online]. Available from: http://www.mastersindatascience.org/industry/retail/ [Accessed: 1 April 2016].
- © Spotless Data Ltd. *EXPLORING DATA ANALYSIS.* [Online]. Available from: https://spotlessdata.com/blog/exploring-data-analysis [Accessed: 10th April 2016].