INTELLIGENT PRODUCT PREDICTION USING SUPERVISED MACHINE LEARNING

A PROJECT REPORT

Submitted by

DIVYALAKSHMI.U (171421601006)

RAMYA.J (171421601026)

SNEHA.V (171421601029)

Under the guidance of

Ms. FEBY A VINISHA

In partial fulfillment for the award of the degree of

Bachelor of Computer Applications

B.S. Abdur Rahman
Crescent
Institute of Science & Technology
Deemed to be University u/s 3 of the UGC Act, 1956
GST Road, Vandalur, Chennai 600 048

www.crescent.education

JULY 2020

BONAFIDE CERTIFICATE

Certified that this project “ **INTELLIGENT PRODUCT PREDICTION USING SUPERVISED MACHINE LEARNING”** is the bonafide work of **DIVYALAKSHMI U (1714216006), RAMYA J (171421601026), SNEHA V (171421601029)** who certified further, that to the best of my knowledge the work reported herein does not form part of any other project report on an earlier occasion on this or any other candidate.

SIGNATURE

Mrs. FEBY A VINISHA

SUPERVISOR

Assistant Professor Head
Department of Computer
Applications
B.S Abdur Rahman Crescent
Institute of Science and Technology
Vandalur, Chennai-600048

SIGNATURE

Dr. GUFRAN AHMAD ANSARI

HEAD OF THE DEPARTMENT

Professor and Head
Department of Computer
Applications
B.S Abdur Rahman Crescent
Institute of science and
Technology
Vandalur, Chennai-600048

B.S.ABDUR RAHMAN
CRESCENT UNIVERSITY
B.S.Abdur Rahman Crescent Institute of Science & Technology
(Estd. u/s 3 of the UGC Act, 1956)

www.crescent.education

VIVA VOICE EXAMINATION

The viva-voice examination of the project work titled “**INTELLIGENT PRODUCT PREDICTION USING SUPERVISED MACHINE LEARNING”** submitted by **DIVYA LAKSHMI U (171421601006**), **RAMYA J (171421601026), SNEHA V (171421601029)**

held on ________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I thank the **Almighty** for showering his blessing upon me in completing the project. I submit this project with a deep sense of gratitude and reverence for my beloved parents for their moral support and encouragement.

I am extremely thankful to **Prof. Dr. A. Peer Mohammed**, Pro Vice Chancellor, B.S. Abdur Rahman Crescent Institute of Science & Technology, for inspiring and motivating us to bring out a perfect and successful project work.

I sincerely thank **Dr. A.AZAD,** Registrar, B.S. Abdur Rahman Crescent Institute of Science & Technology for furnishing every essential facility for doing my project.

I express my sincere gratitude to **Prof. Dr. Venkatesan Selvam**, Dean of School of Computer , Information and Mathematical Science (SCIMS), **Dr. Gufran Ahmad Ansari**, Professor and Head, Department of Computer applications for providing strong oversight of vision , strategic directions, encouragement and valuable suggestions in completing my project work.

I convey my earnest thanks to my project guide, **Ms.FEBY A VINISHA**, Assistant Professor, Department Of Computer Applications, for her valuable guidance and support throughout the project.

I express my gratitude to the Project Coordinators and other Project Committee Members of the Department of Computer Applications for their support and continued assistance in the process

I extend my sincere thanks to all my faculty members for their suitable suggestions, timely advice and support to complete the project.

DIVYALAKSHMI U

RAMYA J

SNEHA V

Abstract

Data is expanding day by day, and it becomes impossible to understand data with higher accuracy. This is where machine learning comes into action. Machine learning provides system the ability to automatically learn and improve from experience. This helps to find natural patterns in the data that provides insight. We use supervised machine learning algorithm to predict and analyze products efficiently and SENTI WORDNET for opinion mining of the products sold.

Opinion mining helps to summarize the appreciation and criticism in the feedback of the customers and rates the feedback accordingly. The system will rank the products based on the feedback from the customers. This classified and summarized information about the product purchased by the existing user help new users to recognize desirable products they wanted to buy.

This project targets on product recommendations for new users. This is attained by using Logistic regression to predict products precisely. Logistic regression is used for predicting binary class and computes the probability of an event occurrence. When the new user visit the application, he views products that has frequent purchase and high rating. Thus, the application is beneficial for customers and helps in decision making.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction of project

In earlier years, data processing techniques like machine learning were developed in the field of information science, which are being applied in various practical areas today. Machine learning is effective for data with strong insufficiency. A Predictive analysis on retail stores is an efficient way to achieve the aforementioned goals and lead to improve the customers satisfaction, reduce destruction of products, increase sales revenue and make production plan efficiently. The research constructs a sales prediction model from the point-of-sale (POS) data of a supermarket and proposes a method for using the advanced data analysis of machine learning for sales analysis .An effective method can help the decision makers in this case, retailers calculate the production costs.

The use of existing resources such as sales data has become a major focus in order to efficiently improve the Purchasing. One use case is prediction of sales, which is an important and complex factor in running a grocery management. There are a wide range of applications where sales prediction can have an impact in the food industry and although prediction tools are used in aiding the prediction of prices and there are still many areas to explore.

There are several techniques to predict sales and historically and the grocery management system has relied on traditional statistical models. However, machine learning in the last decade has grown to become a field with broad application potential and has gradually gained ground in the grocery management system. To predict outcomes of a future event a machine learning model is exposed to data from which it learns patterns that are used to predict the outcome. Time series forecasting predicts future observations (i.e., fare prices) in time series datasets. These datasets consist of sequences of observations collected with equally spaced periods of time. So, a time series forecasting model analyzes historical data to make predictions about the future.

Introduction of domain

Machine learning

In the statistical context, Machine Learning is defined as an application of artificial intelligence where available information is used through algorithms to process or assist the processing of statistical data. While Machine Learning involves concepts of automation, it requires human guidance.

Classes of machine learning

There are two main classes of machine learning techniques:

1. **Supervised machine learning and unsupervised machine learning.**
   A. Examples of supervised learning Logistic regression vs Support vector machines Logistic regression, when used for prediction purposes, is an example of supervised machine learning. In logistic regression, the values of a binary response variable as well as a number of predictor variables are observed for a number of observation units. SVM are an example of a non-statistical supervised machine learning technique; it has the same goal as the logistic regression classifier just described: Given training data, find the best-fitting SVM model, and then use the fitted SVM model to classify new units. The optimization problem that needs to be solved is finding the hyper plane that best separates, in the predictor space, the units with response value 0 from those with response value 1. The logistic regression optimization problem comes from probability theory whereas that of SVM comes from geometry.

B. Examples of unsupervised learning Principal component analysis (statistics) vs. Cluster analysis (machine learning). The main example of an unsupervised machine learning technique that comes from classical statistics is principal component analysis, which seeks to “summarize” a set of data points in high-dimensional space by finding orthogonal one-dimensional subspaces along which most of the variation in the data points is captured.

The term “unsupervised” simply refers to the fact that there is no longer a response variable in the current setting. Cluster analysis and association analysis are examples of non-statistical unsupervised machine learning techniques. The former seeks to determine inherent grouping structure in given data, whereas the latter seeks to determine co-occurrence patterns of items

Chapter 2

LITERATURE REVIEW

2.1 Prediction The Price of National Groceries Using Average Based Fuzzy Time Series With Song - Chissom and Markov Chain Approach Abdul Aziz ; Khoirun Nissa Isti Khomah ; Samgadi Palgunadi Yohanes

Groceries are strategic commodities that have an important role in economic, social, and even political aspects in various countries including Indonesia. The groceries affect the livelihood of the people with the scale of the fulfilment of high needs as well as factors supporting the welfare of the community. The classical problem in the fulfilment of grocery is the fluctuation of the prices of groceries. The increase in the prices of groceries commodities becomes a major factor in inflation. To overcome these problems, one of the efforts made by the government is to stabilize the price policy of grocery so that farmers as producers get profitable results and the community as consumers can afford to buy groceries at affordable prices. To accommodate the afford it is needed a forecasting step to predict the prices of groceries. This study aims to predict the prices of national groceries using the Average Based Fuzzy Time Series method with Song - Chissom and Markov Chain approach. The data used are prices of groceries weekly period from 2015 - 2017. Data is divided into two phases: training and testing dataset with the ratio of 90: 10. Based on MAPE value and feasibility test, it can be concluded that Average Based Fuzzy Time Series with Markov Chain approach shew better than Song - Chissom approach for prediction the prices of national groceries.

2.2 Demand-prediction model for forecasting AGRI-needs of the society B V Balaji Prabhu ; M Dakshayini

In developing countries like India, at present there is no efficient system model to effectively manage the current grocery needs of the society and educate the farmers in growing the required crops to fulfill those needs satisfactorily. Due to this lack of demand (needs) supply data analysis and management, most of the times farmers suffer from the loss and consumers suffers from high prices. In order to address this problem, enormous data about the demand, supply and price variation of various crops have been collected and analyzed. The results have shown a huge gap between a demand and supply of various crops due to which both customers and farmers are facing problems. Hence, this paper proposes a demand-Prediction forecasting model based on a detailed analysis of varying market conditions and societal needs for food grains for several years. The forecasted values of various parameters are in the range of actual real-time market values.

2.3 Prediction of Consumer Purchasing in a Grocery Store Using Machine Learning Techniques Yi Zuo ; Katsutoshi Yada ; A.B.M. Shawkat Ali

Over the past decades, prediction of costumers' purchase behavior has been significantly considered, and completely recognized as one of the most significant research topics in consumer behavior researches. While we attempt to measure response of purchase intention to the contextual factors such as customers' age, gender and income, product price and sale promotion, most of business models are basing on a linear equation to estimate weight of these factors due to the linear equation is not only intuitive for other academics to compare and replicate but also luminous to explain the results for business practitioners. Nevertheless, comparing with other research fields (e.g. pattern recognition and text classification), the prediction methods for purchase behavior are overconcentration of the linear models, especially linear discriminant analysis and logistic regression analysis. On the other hand, as more and more information and communication technologies (ICT, e.g. POS and sensor) are introduced into retail, marketing and management to collect business data, the volumes of data are increasing in exponential growth. Analysis based on linear models are insufficient to satisfy the requirement of academics and practitioners any more, and machine learning techniques have been increasingly attracted us to conduct them as an alternative approach for knowledge discovery and data mining. With regard to these issues, this paper employs two representative machine learning methods: Bayes classifier and support vector machine (SVM) and investigates the performance of them with the data in the real world.

2.4 Modeling and Values of Vendor Managed Inventory in the Retail Supply Chain **Meiping Xie ; David L. Olson**

Vendor managed inventory (VMI) is one of the most widely discussed partnering initiatives for improving multi-firm supply chain efficiency. VMI became one of the key programs in the grocery industry's pursuit of "efficient consumer response" and garment industry's "quick response". And at present, we know that many companies have successfully improved their supply chain performance by implementing the approach of VMI. With VMI, vendors specify delivery quantities sent to customers through the distribution channel using data obtained from electronic data interchange (EDI). A model for a VMI system is constructed where vendors can manage stock at the retailers' locations. A supply chain system in this model has been constructed which is composed of m vendors and n identical retailers. Vendors' and retailers' profits can be calculated for different retailer order batch policies. In this model, we use several parameters because the distances between the vendors and retailers are not same; the stock costs of vendors can vary; the kinds, prices and quantities of retailers' ordering can vary; stock-out contributions of retailers to the vendors can vary. The objective of the model is to minimize total inventory costs for vendors. The model also can analyze how alternative policies can minimize vendor and retailer profits. A simulation example is given

2.5 Prediction model of agricultural product's price based on the improved BP neural network **Wei Minghua ; Zhou Qiaolin ; Yang Zhijian ; Zheng Jingui**

The price of agricultural products are affected by many factors, and the relationship between independent variables and dependent variables can not use specific mathematical formula to express. The traditional prediction methods emphasized on the linear relationship between the prices, and the limitation is apparent, which lead to the low prediction precision. This paper proposes an improved BP neural network model. Firstly, get factors of price fluctuation of agricultural products through the qualitative analysis and then use the MIV method to choose the strong influent factors as the input nodes of a neural network. Find the optimal structure of BP network through the improved learning algorithm, and then use the improved model to realize the agricultural high precision simulation of the product price. The results show that, the model provides an effective prediction tool for the agricultural product price forecasting.

Chapter 3

SYSTEM ANLYSIS

3.1 Existing system

Grocery stores often offer things like vegetables, food, daily used grocery. As the grocery prices are changing day to day with respect to the items. Now a days people are confusing to buy the grocery items as they are varying from day to day. Grocery store managers must ensure that the store runs smoothly that items are priced comparatively and that customers are satisfied.

3.2 Proposed system

Predictive analytics: Predictive analytics is about analyzing current and historical data to forecast the probability of future events, outcomes, or values in the context of price predictions. Predictive analytics requires numerous statistical techniques, such as data mining (identification of patterns in data) and machine learning. The goal of machine learning is to build systems capable of finding patterns in data, learning from it without human intervention and explicit reprogramming. To solve the price prediction problem, data scientists first must understand what data to use to train machine learning models, and that's exactly why descriptive analytics is needed.

Logistic regression was once the most popular machine learning algorithm. Its simplicity makes it the ideal algorithm to use as an introduction to the study of machine learning. Like most machine learning algorithms, logistic regression creates a boundary edge between binary labels. The purpose of a training process is to place this edge in such a way that most of the labels are divided so as to maximize the accuracy of predictions. The training process requires correct model architecture and fine-tuned hyper parameters, whereas data play the most significant role in determining the prediction accuracy.

This study aims to compare machine earning methods for sales prediction in the food industry. The methods were compared in terms of their prediction accuracy on daily sales in a food store department. Understanding of key concepts involved in effective grocery price management is imperative for people dedicated to the success of buying grocery

Then the specialists collect, select, prepare, pre-process, and transform this data. Once this stage is completed, the specialists start building predictive models. We train the data set by using machine learning algorithms to predict the future prices of the grocery A model that forecasts prices with the highest accuracy rate will be chosen to power a system or an application. So, the framework of the project may look like this:

1. Problem statement.
2. Understanding of market peculiarities. Answering the question: What factors impact prices of a commodity/product/service
3. Data collection, preparation, and pre-processing.
4. Modeling and testing.
5. Deployment of a model into a software system or application.
6. Time series forecasting is quite an interesting task which doesn't have one solution to work best all the time. Different domains and data require different approaches. Sometimes you can use some classical methods like ARIMA [a class of models widely applied for time series data analysis and forecasting]. But much more often a recurrent neural network (RNN) or XG Boost give better accuracy.

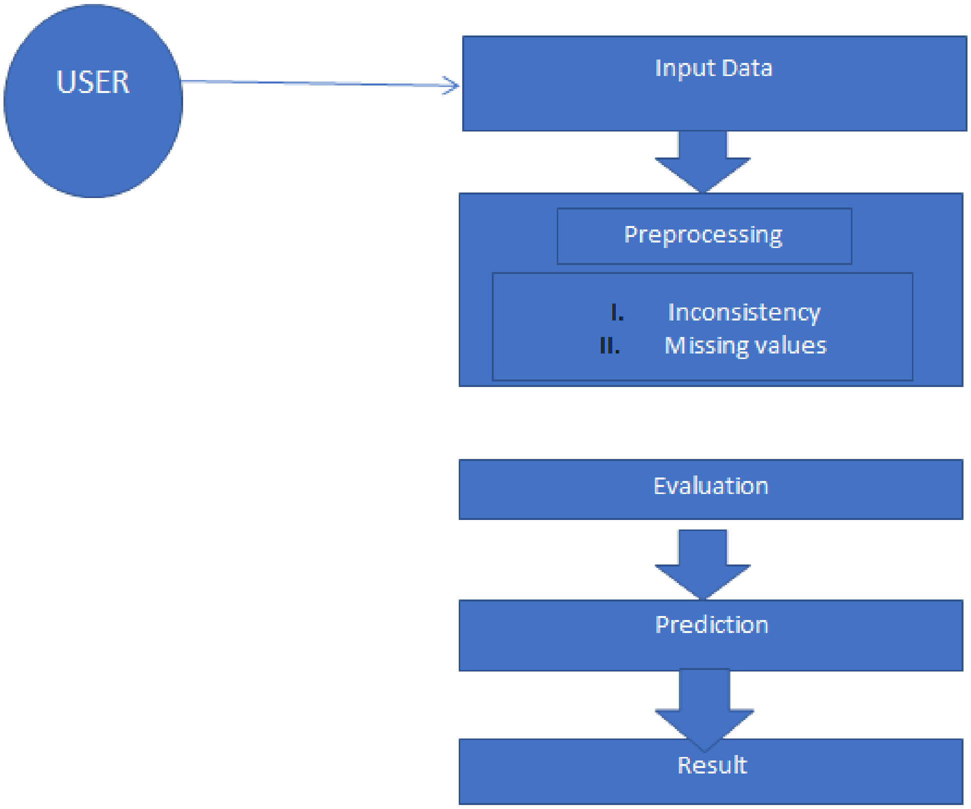
7. **Using XGBoost ensembles.** XG Boost is the implementation of the gradient boosted tree algorithms that's commonly used for classification and regression problems. Gradient boosting is a supervised learning algorithm consisting of an ensemble (set) of weaker models (trees), which sums up their estimates to predict a target variable with more accuracy.

Chapter 4

SYSTEM DESIGN

4.1 System architecture

It defines the structure of the developed system comprising different modules. The externally visible properties and the relationships show the overall architectural design of the system.

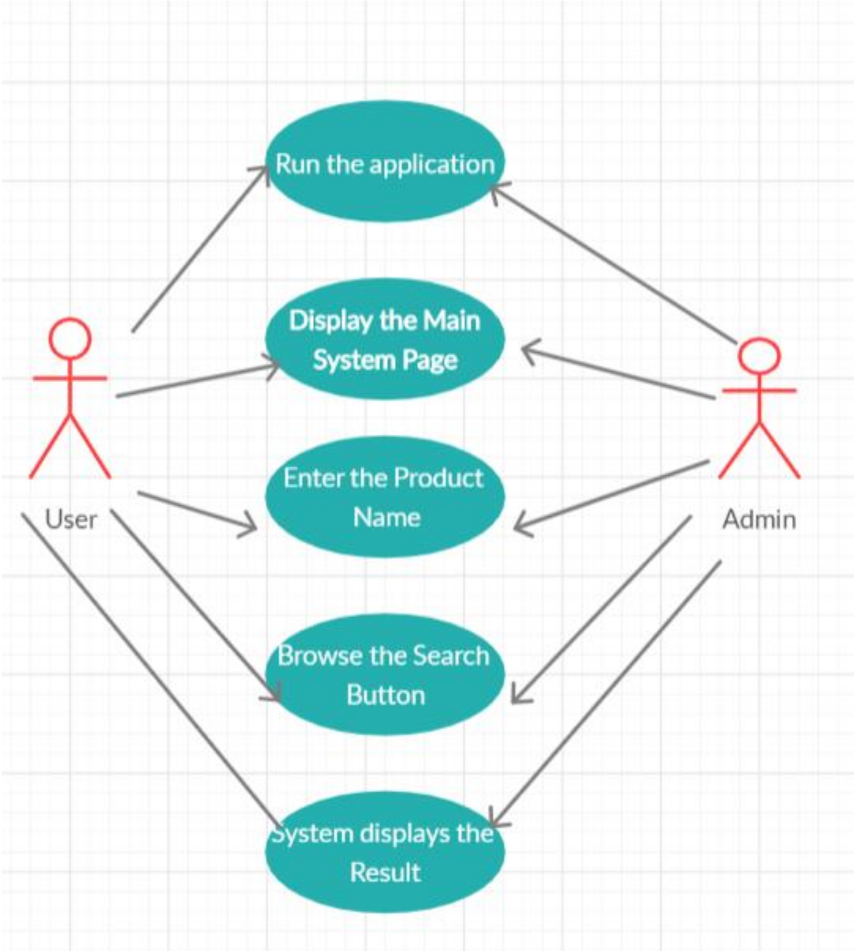


4.2 Uml Diagrams

UML (Unified Modelling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

4.2.1 Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as **actors**.



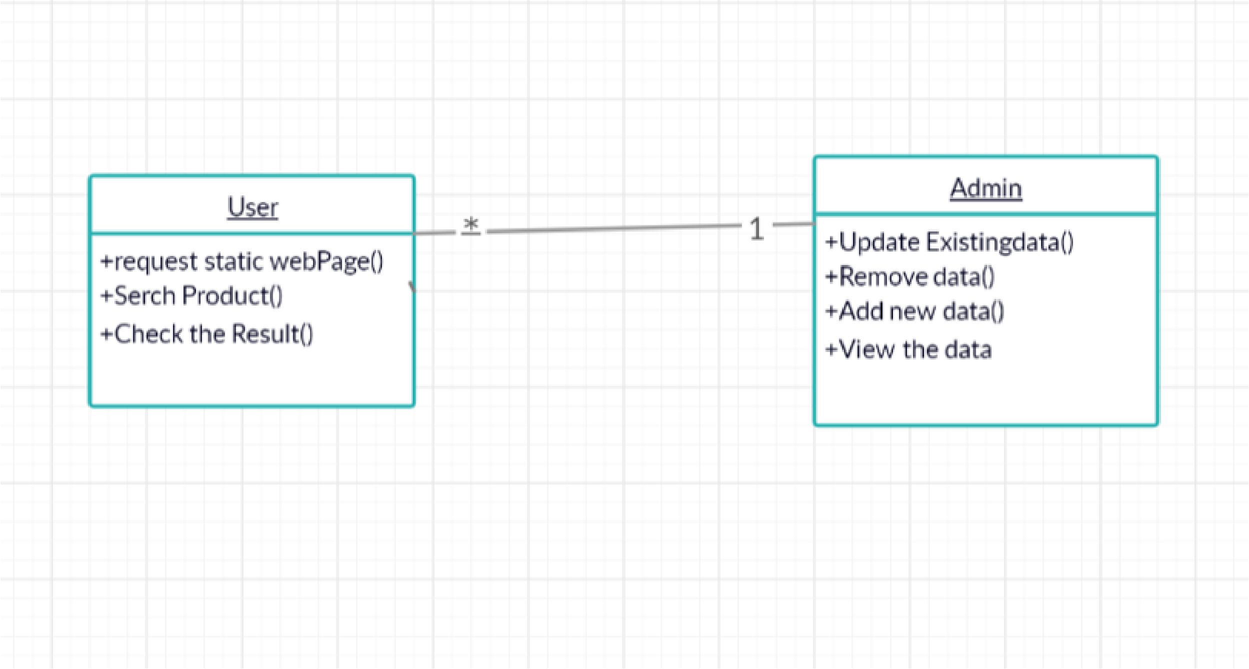
4.2.2 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among the classes.

Class names : User, Admin.

User Operations : Request static web page, Search product, Check the result

Admin Operations : Update existing data, Remove data, Add New data, View the data



USER OPERATIONS

When the user point the cursor in the search bar, he is permitted to view the products that has been explored in the recent past and ready to examine products of their needs.

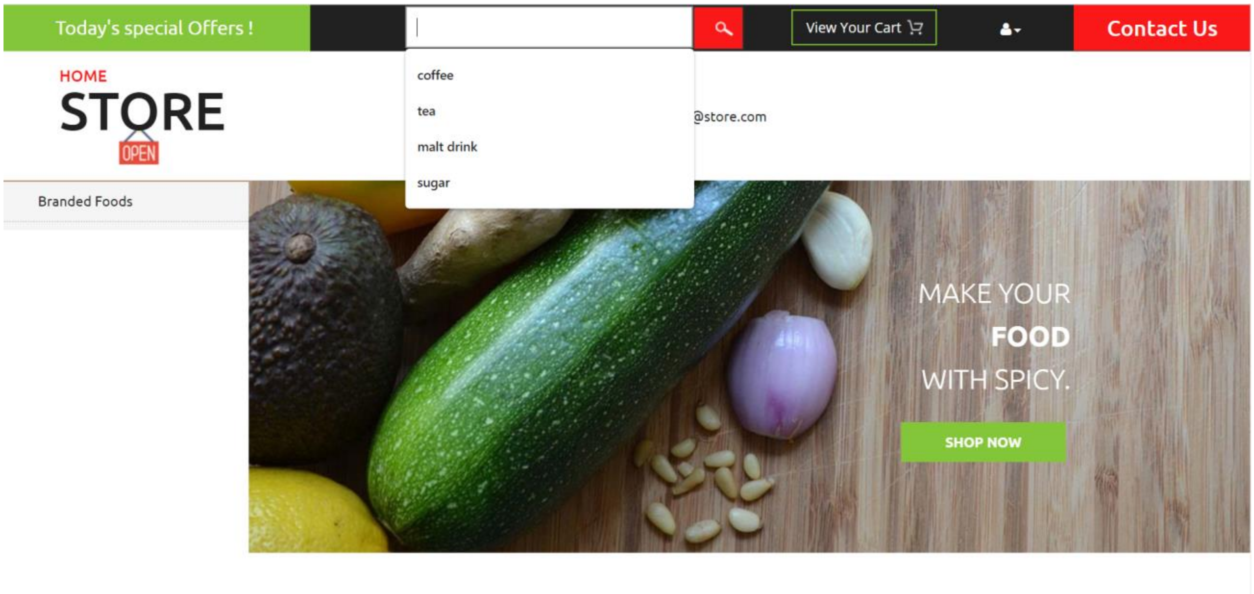


Fig: 4.1 Exploring products

ADMIN OPERATIONS

Admin has the authority to add and pull out products based on the present market situations. He is allowed to provide offers to certain products for the frequent users.

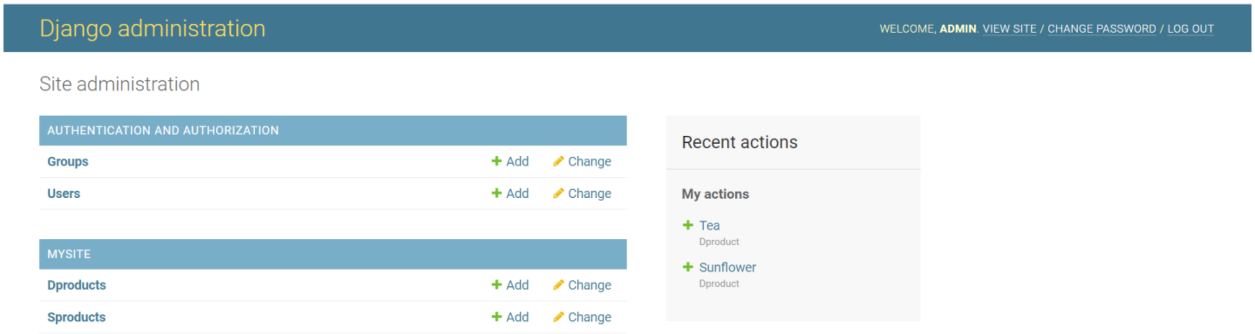
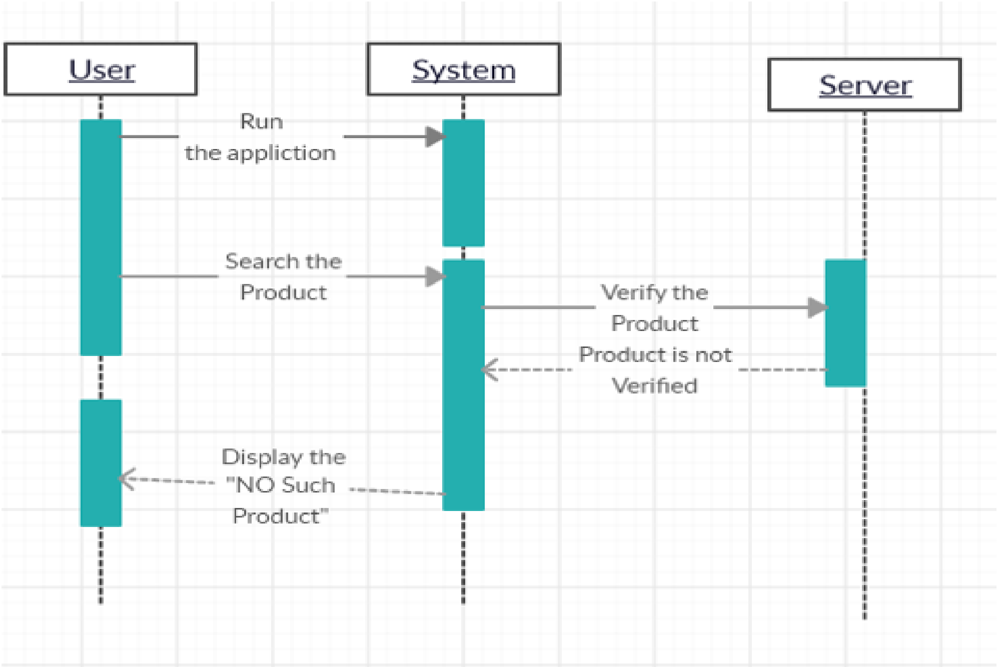


Fig: 4.2 Operations performed by the admin

4.2.3 Sequence Diagram

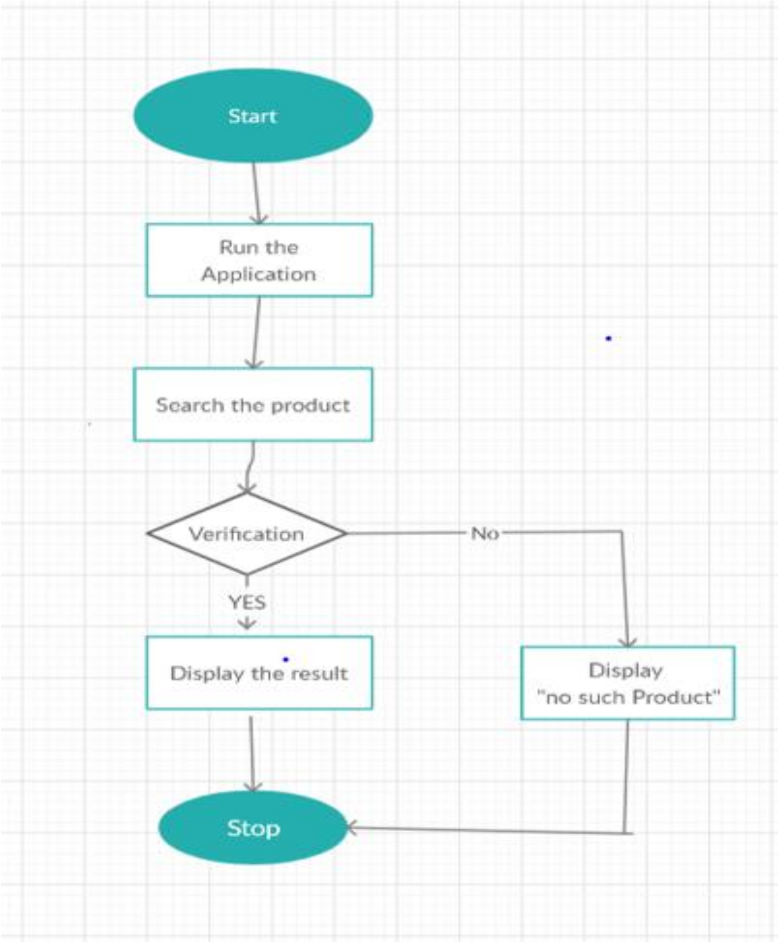
A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. Interaction among the components of a system is very important from implementation and execution perspective.



A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

4.3.4 Activity Diagram

Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system. Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.



Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Chapter 5

SYSTEM IMPLEMENTATION

5.1 Modules:

Data collection: Data collection is a very basic module and the initial step towards the project. It generally deals with the collection of the right dataset. According to the accuracy, we will be using the model with the data to analyze the predictions accurately.

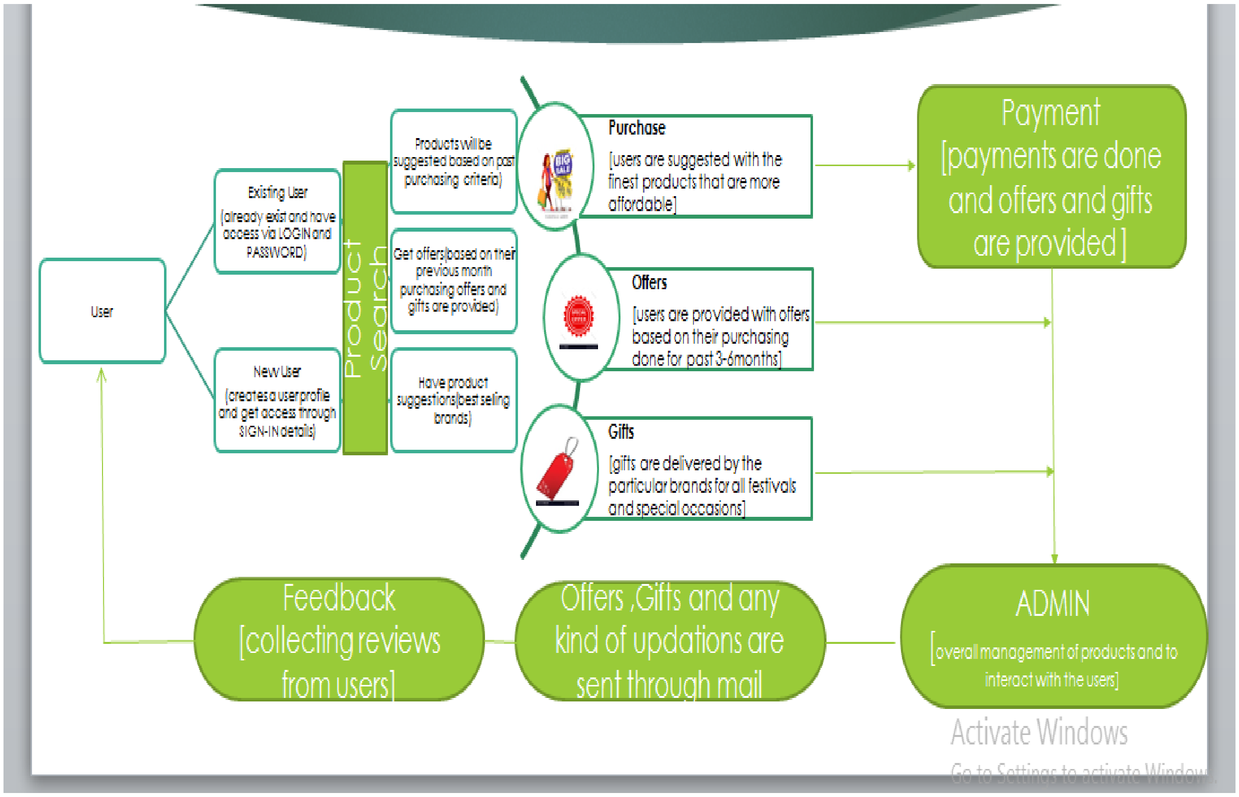

User
Existing User
(already exist and have access via LOGIN and PASSWORD)
New User
(creates a user profile and get access through SIGN-IN details)
Product Search
Products will be suggested based on past purchasing criteria)
Get offers(based on their previous month purchasing offers and gifts are provided)
Have product suggestions(best seling brands)
Purchase
[users are suggested with the finest products that are more affordable]
Offers
[users are provided with offers based on their purchasing done for past 3-6months]
Gifts
[gifts are delivered by the particular brands for all festivals and special occasions]
Payment [payments are done and offers and gifts are provided]
ADMIN [overall management of products and to interact with the users]
Offers ,Gifts and any kind of updations are sent through mail
Feedback [collecting reviews from users]
Activate Windows
Go to Settings to activate Windows

Fig: 5.1 Data collection

All the details regarding products purchased and products searched are being collected in the database. Based on the data collected the admin provides offers and gifts to the users. The feedback given by the users are also concerned and the respective actions are taken.

To gather all the data that required for the web application we use SQlite database. SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file.

Preprocessing: Raw data is usually inconsistent or incomplete and usually contains many error.

The data pre-processing involves checking out for missing values, looking for categorical values, splitting the data-set into training and test set. We use Logistic regression for processing the data.

Training the Machine: Training the machine is similar to feeding the data to the algorithm to touch up the test data. The training sets are used to tune and fit the models. The test sets are untouched, as a model should not be judged based on unseen data. We use Logistic Regression to train the model.

Logistic regression is a fundamental classification technique. It belongs to the group of **linear classifier** and is somewhat similar to polynomial and Logistic regression is fast and relatively uncomplicated, and it’s convenient for you to interpret the results. Although it’s essentially a method for binary classification, it can also be applied to multiclass problems.

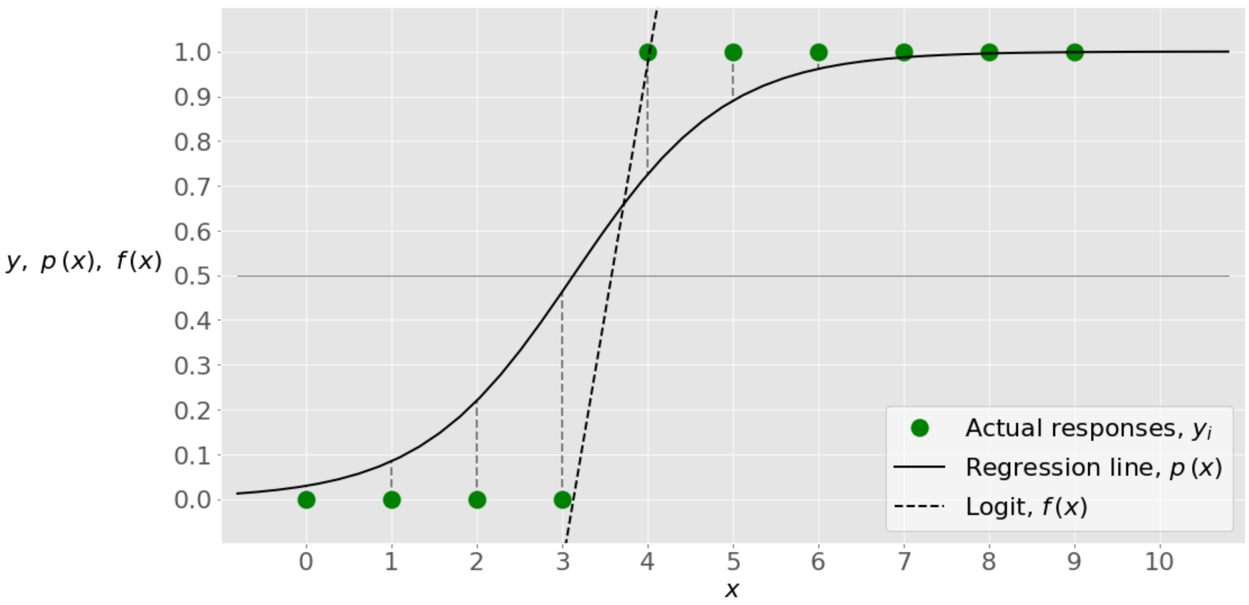


Fig: 5.2 Logistic Regression

This model lists the products that are lately purchased under each category, if the frequency reaches 0.5.

Chapter 6

SOFTWARE DESCRIPTION

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

- **Extendable −** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases −** Python provides interfaces to all major commercial databases.
- **GUI Programming −** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable −** Python provides a better structure and support for large programs than shell scripting.

Python - Environment Setup

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Local Environment Setup

Open a terminal window and type "python" to find out if it is already installed and which version is installed.

- Unix (Solaris, Linux, FreeBSD, AIX, HP/UX, SunOS, IRIX, etc.)
- Win 9x/NT/2000
- Macintosh (Intel, PPC, 68K)
- OS/2
- DOS (multiple versions)
- PalmOS
- Nokia mobile phones
- Windows CE
- Acorn/RISC OS
- BeOS
- Amiga
- VMS/OpenVMS
- QNX
- VxWorks

- Psion
- Python has also been ported to the Java and .NET virtual machines

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python https://www.python.org/

You can download Python documentation from https://www.python.org/doc/. The documentation is available in HTML, PDF, and PostScript formats.

Integrated Development Environment

You can run Python from a Graphical User Interface (GUI) environment as well, if you have a GUI application on your system that supports Python.

- **Unix** − IDLE is the very first Unix IDE for Python.
- **Windows** − PythonWin is the first Windows interface for Python and is an IDE with a GUI.
- **Macintosh** − The Macintosh version of Python along with the IDLE IDE is available from the main website, downloadable as either MacBinary or BinHex'd files.

If you are not able to set up the environment properly, then you can take help from your system admin. Make sure the Python environment is properly set up and working perfectly fine.

We already have set up Python Programming environment online, so that you can execute all the available examples online at the same time when you are learning theory. Feel free to modify any example and execute it online.

panda

pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.

Key Features of Pandas

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of date sets.
- Label-based slicing, indexing and subsetting of large data sets.
- Columns from a data structure can be deleted or inserted.
- Group by data for aggregation and transformations.
- High performance merging and joining of data.
- Time Series functionality.

Standard Python distribution doesn't come bundled with Pandas module.

NumPy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.

Audience

This tutorial has been prepared for those who want to learn about the basics and various functions of NumPy. It is specifically useful for algorithm developers. After completing this tutorial, you will find yourself at a moderate level of expertise from where you can take yourself to higher levels of expertise.

Prerequisites

You should have a basic understanding of computer programming terminologies. A basic understanding of Python and any of the programming languages is a plus. umPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open source project.

Operations using NumPy

Using NumPy, a developer can perform the following operations −

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

NumPy – A Replacement for MatLab

NumPy is often used along with packages like **SciPy** (Scientific Python) and **Mat−plotlib** (plotting library). This combination is widely used as a replacement for MatLab, a popular platform for technical computing. However, Python alternative to MatLab is now seen as a more modern and complete programming language.

It is open source, which is an added advantage of NumPy. tandard Python distribution doesn't come bundled with NumPy module. A lightweight alternative is to install NumPy using popular Python package installer, **pip**.

```
pip install numpy
```

The best way to enable NumPy is to use an installable binary package specific to your operating system. These binaries contain full SciPy stack (inclusive of NumPy, SciPy, matplotlib, IPython, SymPy and nose packages along with core Python).

Chapter 7

SYSTEM CONFIGURATION

7.1 SOFTWARE REQUIREMENTS

Operating System	:	Windows 7
Technology	:	python - Django
Web Technologies	:	Html, JavaScript, CSS
IDE	:	Python IDE
Web Server	:	Django Server
Database	:	SQLITE

7.2 HARDWARE REQUIREMENTS

Hardware	:	Pentium Dual Core
Speed	:	2.80 GHz
RAM	:	4GB
Hard Disk	:	500 GB
Key Board	:	Standard Windows Keyboard
Mouse	:	Two or Three Button Mouse
Monitor	:	SVGA

Chapter 8

SYSTEM STUDY

8.1 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ♦ ECONOMICAL FEASIBILITY
- ♦ TECHNICAL FEASIBILITY
- ♦ SOCIAL FEASIBILITY

8.2 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

8.3 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

8.4 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

chapter 9

SYSTEM TESTING

9.1 Types of testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

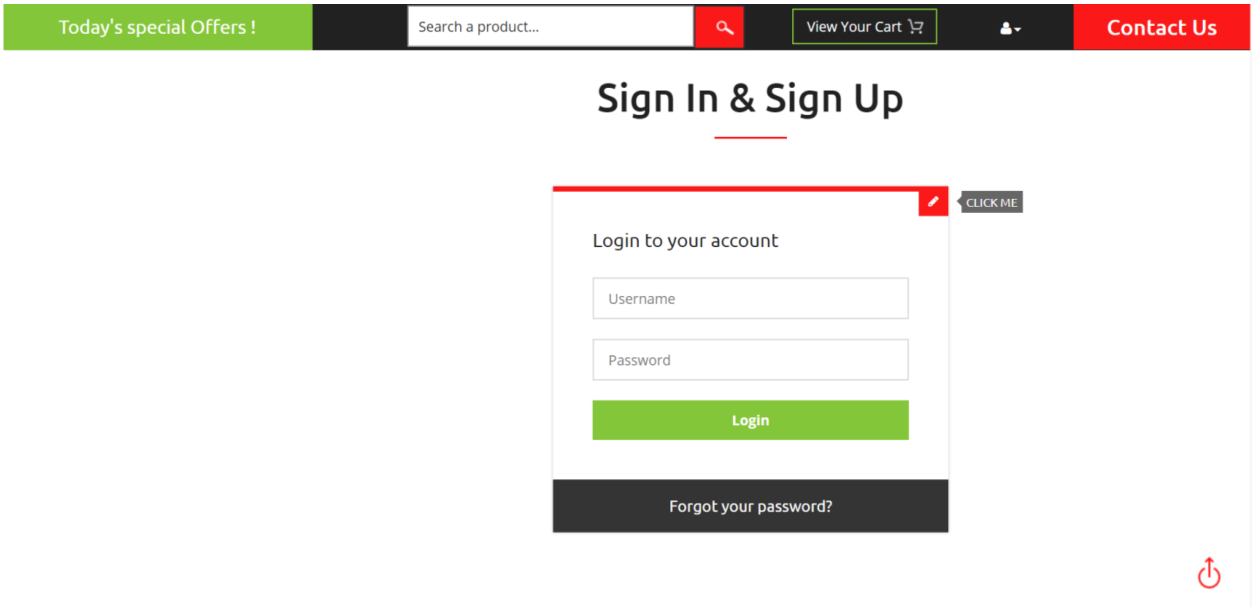


Fig: 9.1 Testing of Sign In and Sign Up details

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

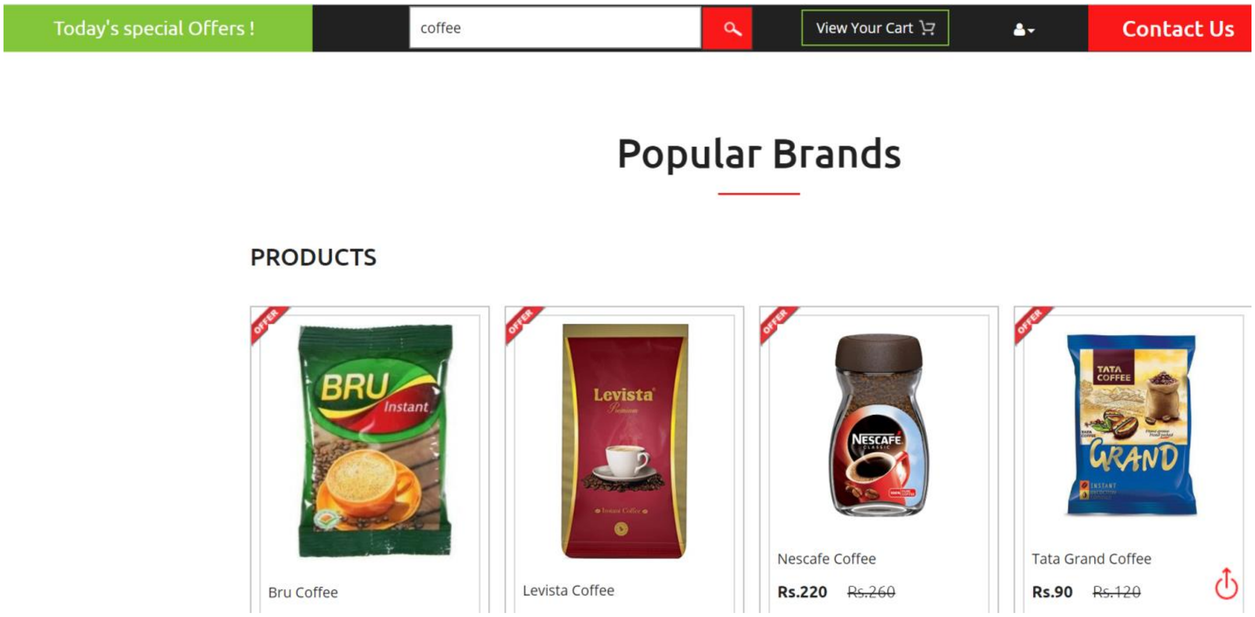


Fig: 9.2 Output displayed based on the search

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format

- No duplicate entries should be allowed
- All links should take the user to the correct page.

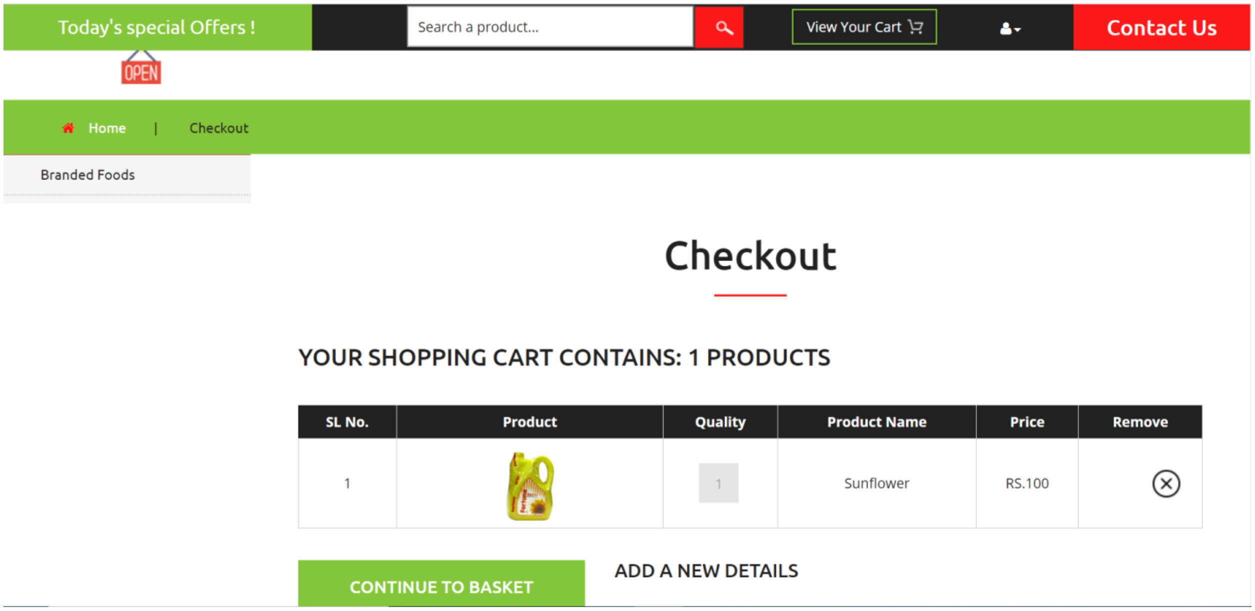
Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.



Checkout

YOUR SHOPPING CART CONTAINS: 1 PRODUCTS

SL No.	Product	Quality	Product Name	Price	Remove
1		1	Sunflower	RS.100	⊗

Fig: 9.3 selected products added to the cart

Chapter 10

CONCLUSION

We will do research on following area of recommendation as well as pricing. We will try to consider both user and providers concerns of changing demand and its cost. This will ensure both provider and customers benefit. Apart from this we will consider competitive prices and its result on pricing. We will study best fit auction based pricing to support optimized fine grained scheme. Also partial waste issue is a area of study which can result in reduced prices using precise scheduling of users' job. User scheduling behaviors and partial usage waste will be brainstormed to find an effective solution.

REFERENCES

[1] Liu Qingyuan, and Wang Tianchuan, “Market price theory and practice”. Dalian: Northeast University of Finance and Economics Press, 1998, pp. 15-20.

[2] R. S. Pindyck, and D. L. Rubinfeld, “Econometric Models and Economic Forecasts”, The McGwar-Hill Companies, Inc, 1998, pp. 251-260.

[3] Yao Xia, Peng Hangen, and Zhu Yan, “ARIMA Time Series Modeling and Applying on Fresh Agricultural Products”, System Sciences and Comprehensive Studies in Agriculture, vol. 23, Feb. 2007 , pp. 88-94.

[4] Nie Rong, Qin Keming, and Zhang Xiaohong, “Stochastic Model and Risk Measuring on the Farm-produce′s Price”, Mathematics In Practice and Theory, vol. 34, Nov. 2004, pp. 108-112.

[5] Zhang Xiaioshuan, “Research of Aquatic Products Price Forecasts Support System”, Beijing: China Agricultural University, 2003, pp. 10-15.

[6] Yu Shouhua, Huang Haoran, and Ou Jingying, “Vegetables Prices Prediction Research of Regional Agricultural Products Wholesale Market”, Research of Agricultural Modernization, 27 monog, Dec. 2006, pp. 118-120.

[7] Cheng Xianlu, “Vegetables Prices Forecast and Predict System Research of Beijing's Agricultural Products Wholesale Market”, Beijing Agricultural Sciences, Feb. 2002, pp. 1-10.

[8] Yuan Zenren, “Artificial neural network and applications”, Beijing: Qinghua University Press, 1998, pp. 13-14.

[9] MT Hagan, and MB Menhaj, “Training Feed Forward Networks with Marquart Algorithm”, IEEE Trans.on Neural Networks, vol. 5, June 1994, pp. 989-993.

[10] Xu Dong, and Wu Zhen, “System analysis and design-neural network based on the MATLAB6.X” (the 2nd version), Xi’an: Xi’an Electronic Technology University Press, 2002, pp. 30-33.

[11] Li Xiaofeng, “New improvement of BP neural network and its application”, Journal of Jilin Institute of Chemical Technology, vol. 17, Dec. 2000, pp. 48-51.

[12] Gao Ling, “The crop pest situation forecasting and achieving in MATLAB based on BP neural network”, Hefei: Anhui Agricultural University, 2003, pp. 45-52.

[13] Jiang Shaofei, “The method of data processing of artificial neural networks in civil engineering problems”, Journal of Harbin University of Civil Engineering and Architecture, vol. 32, Oct. 1999, pp. 24-28.

[14] Liu Yongjian, and Liu Yijian, “The improving methods of arificial neural network in Geo-technical engineering”, Journal of Guangdong University of Technology, vol. 19, Mar. 2002, pp. 21-25.

[15] Liu Wenxi, “New Keynesian price rigidity micro-theoretical foundation”, Economic Science, May 1997, pp. 60-8

APPENDICES

SAMPLE CODE

```
from django.shortcuts import
render,HttpResponse,redirect,get_object_or_404
from .models import Dproduct,Sproduct
from django.contrib.auth.decorators import
login_required,user_passes_test
from django.contrib import messages
from django.contrib.auth.models import User,auth
from django.db.models import Sum
from random import randint
from django.db.models import Q
from django.http import Http404
# Create your views here.
def user_is_not_logged_in(user):
    return  not user.is_authenticated

def index(request):
    count=Dproduct.objects.count()
    dproduct=Dproduct.objects.all()[:randint(1,count)]
    return render(request,"index.html",{'dproduct':dproduct})
@user_passes_test(user_is_not_logged_in)
def register(request):
    if request.method=="POST":
        username=request.POST.get("Username")
        password=request.POST.get("Passwrd")
        email=request.POST.get("Email")
        phone=request.POST.get("Phone")
        if User.objects.filter(username=username).exists():
            messages.info(request,"User Already Exists!")
        else:
            from email.mime.text import MIMEText
            import smtplib
            from django.template.loader import render_to_string
            fromaddr = "your email"
            toaddr = email

            html = render_to_string('smail.html', {'username':
username,"e":email,"p":password,"m":phone})
            msg = MIMEText(html, 'html')
            msg['From'] = "no-reply@homestore.com"
            msg['To'] = toaddr
            msg['Subject'] = "Home Store Registration
Successfull"
            debug = False
            if debug:
                pass
            else:
                server = smtplib.SMTP('smtp.gmail.com',587)
                server.starttls()
                server.login("your email", "your password")
                text = msg.as_string()
                server.sendmail(fromaddr, toaddr, text)
                server.quit()

register=User.objects.create_user(username=username,password=pas
sword,email=email,is_staff='1',first_name=username)
            register.save()
            messages.info(request,"User Created Successfully!")
        return render(request,'login.html')
    else:
        return redirect('index')
def login(request):
    return render(request,"login.html")
@user_passes_test(user_is_not_logged_in)
def signin(request):
    if request.method=="POST":
        username=request.POST.get("Username")
        password=request.POST.get("Password")
        import hashlib
        password_to_enc=password
        result = hashlib.sha1(password_to_enc.encode())
        enc_password=result.hexdigest()

user=auth.authenticate(username=username,password=password)
        if user is not None:
            auth.login(request,user)
            request.session['username'] = username
            return redirect('index')
        else:
            messages.info(request,"Wrong Username and 
Password!")
            return render(request,'login.html')
@login_required
def product(request):
    dproduct=Dproduct.objects.all()
    return render(request,"products.html",{'dproduct':dproduct})
l=[]
@login_required
def addtocart(request,pk):
    l.append(pk)
    request.session['cart'] = l
    return redirect('index')
@login_required
def checkout(request):
    obj=Dproduct.objects.filter(id__in=l)
    counts=Dproduct.objects.filter(id__in=l).count()

total=Dproduct.objects.filter(id__in=l).aggregate(Sum('new_price
'))
    context={"item":obj,"counts":counts,"total":total}
    return render(request,'checkout.html',context)
def mail(request):
    return render(request,"mail.html")
def mailus(request):
    if request.method=="POST":
        name=request.POST.get("Name")
        email=request.POST.get("Email")
        tel=request.POST.get("Telephone")
        sub=request.POST.get("Subject")
        msg=request.POST.get("Message")
        import smtplib
        sender = 'your email'
        receivers = [email]
        message = """From: no-reply@homestore.com
        To: <email>
        Subject: Query For Me!!

        SUB :  """+sub+"""\n
        """+"""Name: """+name+"""\n"""+"""Email:
"""+email+"""\n""""""Tel: """+tel+"""\n"""+"""Sub:
"""+sub+"""\n"""+"""Message: """+msg+""""""
        try:
            server = smtplib.SMTP('smtp.gmail.com', 587)
            server.starttls()
            server.login("your emaill", "your password")
            server.sendmail(sender, receivers, message)
            server.quit()
        except SMTPException:
            pass
    return redirect("index")

def logout(request):
    auth.logout(request)
    l.clear()
    try:
        del request.session['username']
        del request.session['cart']
    except:
        pass
    return redirect("index")

def remove(requext,pk):
    l.remove(pk)
    return redirect('checkout')

def search(request):
    q=request.POST.get("Product")
    if q is None:
        messages.info(request,"No Product found!")
        return render(request,'products.html')
    else:
        result=Dproduct.objects.filter(Q(name__icontains=q))
    return render(request,'products.html',{"dproduct":result})

def orders(request,pk):
    result=Sproduct.objects.filter(user_id=pk)

result1=Dproduct.objects.filter(id__in=result.values('product_id
'))
    import datetime
    date=datetime.datetime.now()
    dates=Sproduct.objects.filter(ordered_date__year=2020)
    return
render(request,'orders.html',{"dproduct":result1,"date":date,"id
":pk,"dates":dates})
```

```
def payment(request):
    return render(request,'payment.html')


def placeorder(request,pk):
    liss=request.session['cart']
    name=request.user.first_name
    ids=request.user.id
    import datetime
    for s in liss:

orda=Sproduct.objects.create(user_id=ids,product_id=s,ordered_date=datetime.datetime.now())
        orda.save()
    return render(request,'Done.html')
```

SETTINGS

```
import os
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
SECRET_KEY = '+=4v-c%w7=fkogm@%a5evo$d(c=&zedlx-1tc*vhzsk%z%o@^0'
DEBUG = True

ALLOWED_HOSTS = []
INSTALLED_APPS = [
    'mysite.apps.MysiteConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'project.urls'
```

```
1
TEMPLATES = [
    {
        'BACKEND':
'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',

'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'project.wsgi.application'

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarity
Validator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator'
,
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator
',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidato
r',
    },
]
STATIC_URL = '/static/'

MEDIA_URL = '/mysite/static/media/'

MEDIA_ROOT = os.path.join(BASE_DIR,'media')

LOGIN_URL = '/login'
```

URL

```
from django.urls import path,include

from django.conf.urls import url

from . import views

urlpatterns = [

   url(r"^$",views.index,name="index"),

   url(r"^login/$",views.login,name="login"),

   url(r"^signin/$",views.signin,name="signin"),

   url(r"^register/$",views.register,name="register"),

   url(r'^products/$',views.product,name='product'),

   url(r"^checkout/$",views.checkout,name="checkout"),

    url(r"^addtocart/(?P<pk>[0-

9]+)/$",views.addtocart,name="addtocart"),

   url(r'^mail/$',views.mail,name='mail'),

   url(r'^mailus/$',views.mailus,name='mailus'),

   url(r'^logout/$',views.logout,name='logout'),

   url(r'^remove/(?P<pk>[0-9]+)/$',views.remove,name='remove'),

   url(r'^search/$',views.search,name='search'),

   url(r'^orders/(?P<pk>[0-9]+)/$',views.orders,name='orders'),

   url(r'^payment/$',views.payment,name='payment'),

   url(r'^placeorder/(?P<pk>[0-

9]+)/$',views.placeorder,name='placeorder')
```

MODELS

```
from django.db import models

from django.contrib.auth.models import User

import datetime

class Dproduct(models.Model):

    img=models.ImageField(upload_to="pics")

    name=models.CharField(max_length=100)

    new_price=models.IntegerField()

    old_price=models.IntegerField()

    gift=models.BooleanField(default=False)

    offer=models.BooleanField(default=False)


    def __str__(self):

        return self.name


class Sproduct(models.Model):

    user=models.ForeignKey(User, on_delete=models.CASCADE)

    product=models.ForeignKey(Dproduct,on_delete=models.CASCADE)


ordered_date=models.DateTimeField(default=datetime.datetime.now(
))

    def __str__(self):

        return self.user.username+" - "+self.product.name
```

TECHNICAL BIOGRAPHY

U. DIVYALAKSHMI (RRN 171421601006) was born on 25th FEBRUARY 2000.Currently she is pursuing BCA degree which is specialized in Data Science in the **Department of Computer Application** of **B.S ABDUR RAHMAN CRESCENT INSITITUE OF SCIENCE AND TECHNOLOGY VANDALUR, CHENNAI-600048**

Email ID: divyadiv0510@gmail.com

Mobile : 9962254461

J. RAMYA (RRN 171421601026) was born on 12th JUNE 2000.Currently she is pursuing BCA degree which is specialized in Data Science in the **Department of Computer Application** of **B.S ABDUR RAHMAN CRESCENT INSITITUE OF SCIENCE AND TECHNOLOGY VANDALUR, CHENNAI-600048**

Email ID: ramyarajalakshmi12@gmail.com

Mobile : 9840271409

V. SNEHA (RRN 171421601029) was born on 19th AUGUST 2000. Currently she is pursuing BCA degree which is specialized in Data Science in the **Department of Computer Application** of **B.S ABDUR RAHMAN CRESCENT INSITITUE OF SCIENCE AND TECHNOLOGY VANDALUR, CHENNAI-600048.**

Email ID: shakthisneha19@gmail.com

Mobile : 9123597325