

E- RATION SHOP

PROJECT REPORT

Submitted to the Calicut University in partial fulfilment of the requirement
for the award of the degree of

Bachelor of Computer Science (BSc CS)

By

MUHAZEENA FATHIMA

(Reg.No: ARATSCS004)

Under the guidance of

Ms. Fayisa P H

Ms.Sainaba Soonam P K



DEPARTMENT OF COMPUTER SCIENCE

ANSAR WOMEN'S COLLEGE, PERUMPILAVU
2019-2022



ANSAR WOMEN'S COLLEGE, PERUMPILAVU
DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the project report entitled **E -RATION SHOP** is a bonafide record of work done by **MUHAZEENA FATHIMA (Reg.No: ARATSCS004)** of Department of Computer science, under the guidance and supervision of Mrs.Fayisa P H and Sainaba Soonam P k in partial fulfilment of requirements for the award of degree of Bachelor of Computer science under the University of Calicut during the year 2019-2022.

Internal Guide:

Head Of Department:

The presentation and viva-voce examination was conducted on at Ansar Women's college.

Internal Examiner:

External Examiner:

DATE:

DECLARATION

I ,**MUHAZEENA FATHIMA** of sixth semester, Bachelor of Computer Application, do hereby declare that the project report entitled “**E RATION SHOP**”, is the result of my original work done by me along with my Group Members **FASNA BEEVI,RISHMA SHERIN AND HASEENA V R** under the guidance of **Ms Sainaba Soonam P K** and **Ms Fayisa P H** .

I also declare that this work has not been submitted earlier by me too University of Calicut or any other institutions for the fulfilment of the requirement of the course of study. The imperial findings in the report are based on the data collected by ourselves.

Place : PERUMPILAVU

Name : MUHAZEENA FATHIMA

Date:

Reg.No : ARATSCS004

Signature :

ACKNOWLEDGMENT

At the very outset, we would like to give the first honors to almighty God, who gave the wisdom and knowledge to complete this project.

We owe our deep sense of gratitude to **Mr Kamaludheen K.T**, the principal of Ansar Women's College, Perumpilavu for having encouraged us to undertake this project work. We would also like to express sincere thanks to **Mrs CM Sulaikha**, Head of the Department of Computer Science for making necessary arrangements to enable us to undertake our project with keen interest

It is a great pleasure for us to express our profound and sincere thanks to our trainer Mr . **Gopikrishnan P G (Grapesgenix technical solution)**.

We would also like to express our gratitude to our internal guides **Ms Sainaba Soonam P K (Asst.professor)** and **Ms Fayisa P H (Asst.professor)** for their valuable guidance and assistance in completing the project in a determined way within stipulated time .

We express our immense pleasure and thankfulness to our parents, friends and all teaching and nonteaching staffs of the Department of Computer Science, Ansar Women's College, Perumpilavu for their encouragement and support

MUHAZEENA FATHIMA

TBLE OF CONTENT

| CHAPTER | TITLE | PAGE NO |
|----------------|---|----------------|
| 1 | INTRODUCTION | 1-5 |
| | 1.1 SYNOPSIS | 2 |
| | 1.2 ABOUT THE PROJECT | 2 |
| | 1.3 MAIN OBJECTIVE OF THE PROJECT | 3 |
| | 1.4 INTRODUCTION TO PYTHON | 4-5 |
| 2 | DEVELOPMENT ENVIRONMENT &EXECUTION ENVIRONMENT | 6-12 |
| | 2.1DATABASE | 7 |
| | 2.2 PROGRAMMING LANGUAGES | 7-11 |
| | 2.3 SOFTWARE ARCHITECTURE AND DEVELOPMENT TOOLS | 7-12 |
| | 2.4 HARDWARE SPECIFICATIONS | 12 |
| 3 | SYSTEM STUDY | 13-17 |
| | 3.1 EXISTING SYSTEM | 14 |
| | 3.2 PROPOSED SYSTEM | 14 |
| | 3.3 PLANNING AND SCHEDULING | 14-15 |
| | 3.3.1 PERT CHART | 16 |
| | 3.3.2 GANT CHAR | 17 |
| 4 | SYSTEM ANALYSIS | 18-21 |
| | 4.1 OVERVIEW | 19 |
| | 4.1 FEASIBILITY ANALYSIS | 19 |
| | 4.1.1 TECHNICAL FEASIBILITY | 20 |
| | 4.2.2ECONOMICAL FEASIBILITY | 20-21 |
| | 4.2.3OPERATIONAL FEASIBILITY | 21 |
| 5 | SYSTEM DESIGN | 22-30 |

| | | |
|----|-----------------------------------|-------|
| | 5.1 INPUT DESIGN | 23 |
| | 5.2 OUTPUT DESIGN | 23 |
| | 5.3 DATAFLOW DIAGRAM | 24-27 |
| | 5.4 DATABASE DESIGN | 28-30 |
| 6 | SYSTEM TESTING AND IMPLEMENTATION | 31-38 |
| | 6.1 SYSTEM TESTING | 32-35 |
| | 6.2 SYSTEM IMPLEMENTATION | 36 |
| | 6.3 SYSTEM MAINTENANCE | 37-38 |
| 7 | FUTURE ENHANCEMENT | 39 |
| 8 | CONCLUSION | 40 |
| 9 | BIBILOGRAPHY | 41-42 |
| 10 | APPENDIX | 43-77 |
| | 10.1 SAMPLE CODE | 44-70 |
| | 10.2 SCREEN SHOTS | 71-77 |

1. INTRODUCTION

Public Distribution System (PDS) is an Indian food security system. It is established by the Government of India under Ministry of Consumer Affairs, Food, and Public Distribution and managed jointly with state governments in India. The traditional PDS is used to distribute grocery items to India's poor who are valid ration card holders. The validity and the allocation of the ration cards is monitored by the state governments. A ration card holder should be given 35 kg of food grain as per the norms of PDS. However, there are concerns about the efficiency of the distribution process. In order to make it efficient and improve the current system of PDS we are implementing e-Ration Shop. Here we are going to make a website for shopping purpose. Using this website ration card holder can order his/her grocery items from the FPS online. Most of the times Users do not get their rightful entitlement in terms of quantity. What's meant for them or the farm produce procured by the FPS's is diverted to the open market. So in order to avoid all these drawbacks we are going to use the e-Ration Shop which will help us to avoid the corruption in PDS if not eradicate it.

1.1 SYNOPSIS

The main purpose of this site is to help the users to get the details related to their ration shop. Can avoid dealer communication, manual calculation and it's reduces the time spending in the shops. The main reason for using this website is making this process computerized and to remove the drawbacks of the present way of issuing products based on ration card. The main drawback in the current system is that the PDS has been criticized for its urban bias and its failure to serve the poorer sections of the population effectively. Also many retail shopkeepers have large number of bogus cards to sell food grains in the open market. Many FPS dealers resort to malpractice since they acquire less. Salary So, by this project we are solving this problem too.

1.2 ABOUT THE PROJECT

The Indian ration card provides food for the poor people which is distributed by the government along with the fuel. It provides a distinct identity of person which is useful to update with the government record. The basic food items provided by government are rice, sugar, wheat. Ration Card is one of the most important documents which acts as identity proof for any individual. If people are not having their own Ration card they can also apply for the same. The process to apply for ration card has been facilitated to great extent but now a days this process is online which comes as blessing for the applicants who hate standing for long time in queues for filling the application form and then go to the office again to know the status. The network of the ration shops is spread all over in India to provide food security to the people. This distribution of food and fuel is fully controlled by the government. But it has so many limitations. Most of the ration shopkeepers keep fake ration cards with them. Due to availability of all ration items these items are present with the ration shop dealer so he can falsify the records and use the items to sell in the market loosely. The dealer then does not provide these ration items to the customers. Many a times people are not aware that the items have arrived in the shop. The dealer then sells these items in increased rates in the market. In this way, in the current situation we are facing problem due to lack in transparency. There is no such good system yet developed through which government gets message of usage of grains by the people. Hence, we have proposed an e-ration card management system which is based on RFID technology and biometric authentication technology that replaces traditional ration cards.

1.3 MAIN OBJECTIVE OF THE PROJECT

This e-Ration Shop website is to give information related to card holders and to record all transactions. FPS does not open every day, nor do they keep regular hours. Even on the days that the FPS is open, ration card holders have to stand in long queues. Generally, FPS neither opens every day nor do they keep regular hours. This website will provide information about the day and time to report at e – Ration Shop for the food items that they can get on their ration card and several other periodic, cumulative reports, saving a lot of time and avoiding manual errors. This will also send them a notification e-mail providing information about the day and time when ration shop will open. This makes the card holders not to stand in queue for long duration of time. No mechanism to identify inclusion errors such as duplicate, bogus and ineligible. The objective of this site is to avoid dealer communication and reduces the time spending in the shops. The user and shopkeeper can register through the application and after the successful approval from admin they can login to the site. This site provides the details and price of the product, details about the kit.

For the users to use the E- ration shop smoothly we provide QR code facility. So that the card holders can use it to login to the website. They just have to open the phone's native camera app or a QR Code scanning app and hold in front of the QR Code. Within 2-3 seconds, they receive a notification indicating that the QR Code has been scanned.

In the existing system we don't have the choice to purchase anything other than ration items so, as a solution for this we provide grocery items too. The card holders can buy the desired product by using the same credentials. They can arrange them for home delivery directly from the store, or he can pick up his order at the store once an employee has assembled it.

1.4 INTRODUCTION TO PYTHON



Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry. Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0.[32] Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, reference counting, and Unicode support

FEATURES OF PYTHON

❖ SIMPLE

- Looks familiar to existing programmers related to C and C++.
- Contains no go to statement, but break and continue.
- Has no header files and eliminated C pre-processor.

Eliminated much redundancy.

1. Have no pointers which can be used by the programmers.

OBJECT ORIENTED

Java is an object oriented language, which means that you focus on the data in your application and methods that manipulate that data, rather than thinking strictly in terms of procedures. In an Object Oriented system, a class is a collection of data and methods that operate on that data. Taken together, the data and methods describe the state and behavior of an object. Classes are arranged in a hierarchy, so that a subclass can inherit from its super class. Java comes with an extensive set of classes, arranged in package that you can use in your programs.

❖ DISTRIBUTED

1. Now uses mostly TCP-IP based protocols like ftp and http.
2. It has a spring-like transparent RPC system.

Java supports various levels of network connectivity through classes in the java.net package.

❖ INTERPRETED

The Java compiler generated bytes-codes, rather than native machine code. To actually run java program, you use the java interpreter to execute the compiled byte-codes. Java byte-codes provide an architecture-neutral object file format. The code is designed to transport programs efficiently to multiple platforms.

1. Rapid turn-around development.
2. Software author is protected, since binary byte streams are download and not the source code.

❖ ROBUST

Java has been designed for writing highly reliable or robust software.

1. Language restrictions to make impossible for applications to smash memory
2. Java does automatic garbage collection, which prevents memory leaks.
3. Extensive compile-time checking so bugs can be found early, that is repeated at runtime for flexibility and to check consistency.

a. SECURE

Security is an important concern, since Java is meant to be used in networked environments. Without some assurance of security, you certainly wouldn't want to download an applet from a random site on the net and let it run on your computer. Java's memory allocation model is one of its main defenses against malicious code.

1. Access restrictions are enforced.
2. Byte codes are verified, which copes with the threat of a hostile compiler.

2. DEVELOPMENT ENVIRONMENT & EXECUTION ENVIRONMENT

2.1 DATABASE

SQLite3

The SQLite3 module provides the Python interface to connect to a SQLite database from a Python program. Once the SQLite3 module is imported within a Python program, the connect method of the module can be invoked by specifying the database file name. The sqlite3 module is a powerful part of the Python standard library; it lets us work with a fully featured on-disk SQL database without installing any additional software. SQLite3 can be integrated with Python using sqlite3 module, which was written by Gerhard Haring. It provides an SQL interface compliant with the DB-API 2.0 specification described by PEP 249. You do not need to install this module separately because it is shipped by default along with Python version 2.5.x onwards. To use sqlite3 module, you must first create a connection object that represents the database and then optionally you can create a cursor object, which will help you in executing all the SQL statements

2.2 PROGRAMMING LANGUAGES

-

The coding step transfers a detailed design of representation of software into a programming language realization. All the software engineering steps that have been presented up to this step are directed towards a final objective. Coding is a process that transforms design into a programming language. The translation process begins when a compiler accepts the source code as input and produces machine dependent object code as output. Compiler output is further translated into machine code. Language characteristics have an impact on the quality and efficiency of translation. Building the application refers to the coding step in software engineering process.

I have done this project by using HTML, CSS, and JavaScript as front-end languages. Since it is a web site it requires the database and I here used Python, SQLite3 Server as back end language to access and perform operations. Also, Here Anaconda Navigator is as an Integrated Development Environment and Django as framework.

PYTHON

Python is an interpreted high-level language general purpose programming language. Its design philosophy emphasizes code readability with its use of significant identification. its language constructs as well as it object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Described as a “batteries included” language due to its comprehensive standard library.

Guido van Rossum began working on python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle detecting garbage collection system (in addition to reference coding). Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020. Python consistently ranks as one of the most popular

programming languages.

Features of Python:

Easy to code

- ❖ Object-Oriented language
- ❖ Free and open source
- ❖ GUI programming support
- ❖ High level language
- ❖ Extensible feature
- ❖ Portable
- ❖ Interpreted
- ❖ Integrated language
- ❖ Large standard library
- ❖ Dynamically typed language

Django

Django is an open-source web framework written in the Python programming language. Named after the jazz guitarist Django Reinhardt, it is used by some of the largest websites in the world including Instagram, Mozilla, and NASA, but also lightweight enough to be a popular choice for weekend side projects and startups. Its "batteries-included" approach means a powerful website can be generated quickly in the hands of a skilled developer. Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

HTML

The Hyper Text Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as `` and `<input />` directly introduce content into the page. Other tags such as `<p>` surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the

behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997. A form of HTML, known as HTML5, is used to display video and audio, primarily using the <canvas> element, in collaboration with JavaScript.

CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable

JAVASCRIPT

JavaScript often abbreviated JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. Over 97% of websites use JavaScript on the client side for web page behavior, often incorporating thirdparty libraries. All major web browsers have a dedicated JavaScript engine to execute the code on the user's device. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based objectorientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

Anaconda Navigator

Anaconda is the distribution of the python and R programming languages for scientific computing(data science, machine learning applications, large-scale data processing, predictive analysis, etc.) that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of

which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages

MACHINE LEARNING (ML)

ML is the study of computer algorithms that can improve automatically through experience and by the use of data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so.¹ Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks. A subset of machine learning is closely related to computational statistics, which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of mathematical optimization, delivers methods, theory and application domains to the field of machine learning. The term *machine learning* was coined in 1959 by Arthur Samuel, an American IBMer and pioneer in the field of computer gaming and artificial intelligence. Also the synonym *self-teaching computers* was used in this time period. A representative book of the machine learning research during the 1960s was the Nilsson's book on Learning Machines, dealing mostly with machine learning for pattern classification. Interest related to pattern recognition continued into the 1970s, as described by Duda and Hart in 1973 In 1981 a report was given on

using teaching strategies so that a neuralnetwork learns to recognize 40 characters (26 letters, 10 digits, and 4 special symbols) from a computer terminal.

Tom M.Michel provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E . This definition of the tasks in which machine learning is concerned offers a fundamentally operational division rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing machinery and intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?".

Modern day machine learning has two objectives, one is to classify data based on models which have been developed, the other purpose is to make predictions for future outcomes based on these

models. A hypothetical algorithm specific to classifying data may use computer vision of moles coupled with supervised learning in order to train it to classify the cancerous moles. A machine learning algorithm for stock trading may inform the trader of future potential predictions.

OVERVIEW OF PL/SQL

PL/SQL is Oracle's procedural language extension to SQL. PL/SQL combines the ease and flexibility of SQL with the procedural functionality of a structured programming language, such as IF ... THEN, WHILE, and LOOP.

When designing a database application, consider the following advantages of using stored PL/SQL:

- PL/SQL code can be stored centrally in a database. Network traffic between applications and the database is reduced, so application and system performance increases. Even when PL/SQL is not stored in the database, applications can send blocks of PL/SQL to the database rather than individual SQL statements, thereby reducing network traffic.
- Data access can be controlled by stored PL/SQL code. In this case, PL/SQL users can access data only as intended by application developers, unless another access route is granted.
- PL/SQL blocks can be sent by an application to a database, running complex operations without excessive network traffic.
- Oracle supports PL/SQL Server Pages, so your application logic can be invoked directly from your Webpages.

The following sections describe the PL/SQL program units that can be defined and stored centrally in a database

PL/SQL Program Units

Program units are stored procedures, functions, packages, triggers, and autonomous transactions.

Procedures and functions are sets of SQL and PL/SQL statements grouped together as a unit to solve a specific problem or to perform a set of related tasks. They are created and stored in compiled form in the database and can be run by a user or a database application. Procedures and functions are identical, except that functions always return a single value to the user. Procedures do not return values. Packages encapsulate and store related procedures, functions, variables, and other constructs together as a unit in the database. They offer increase functionality (for example, global package

variables can be declared and used by any procedure in the package). They also improve performance (for example, all objects of the package are parsed, compiled, and loaded into memory once).

2.3 SOFTWARE ARCHITECTURE AND DEVELOPMENT TOOLS

The software configuration says about the software's and packages used to design the system. The Following specifications includes the software and packages used by the programmer.

| | | |
|------------------|---|---------------|
| platform | : | Visual code |
| front end tool | : | Python-Django |
| back end tool | : | SQLITE |
| operating system | : | windows |
| server | : | c panel |

2.4 HARDWARE SPECIFICATIONS

| | | |
|-----------|---|-----------|
| PROCESSOR | : | PENTIUM13 |
| RAM | : | 4 GB |
| HARD DISK | : | 1 TB |

3. SYSTEM STUDY

3.1 EXISTING SYSTEM

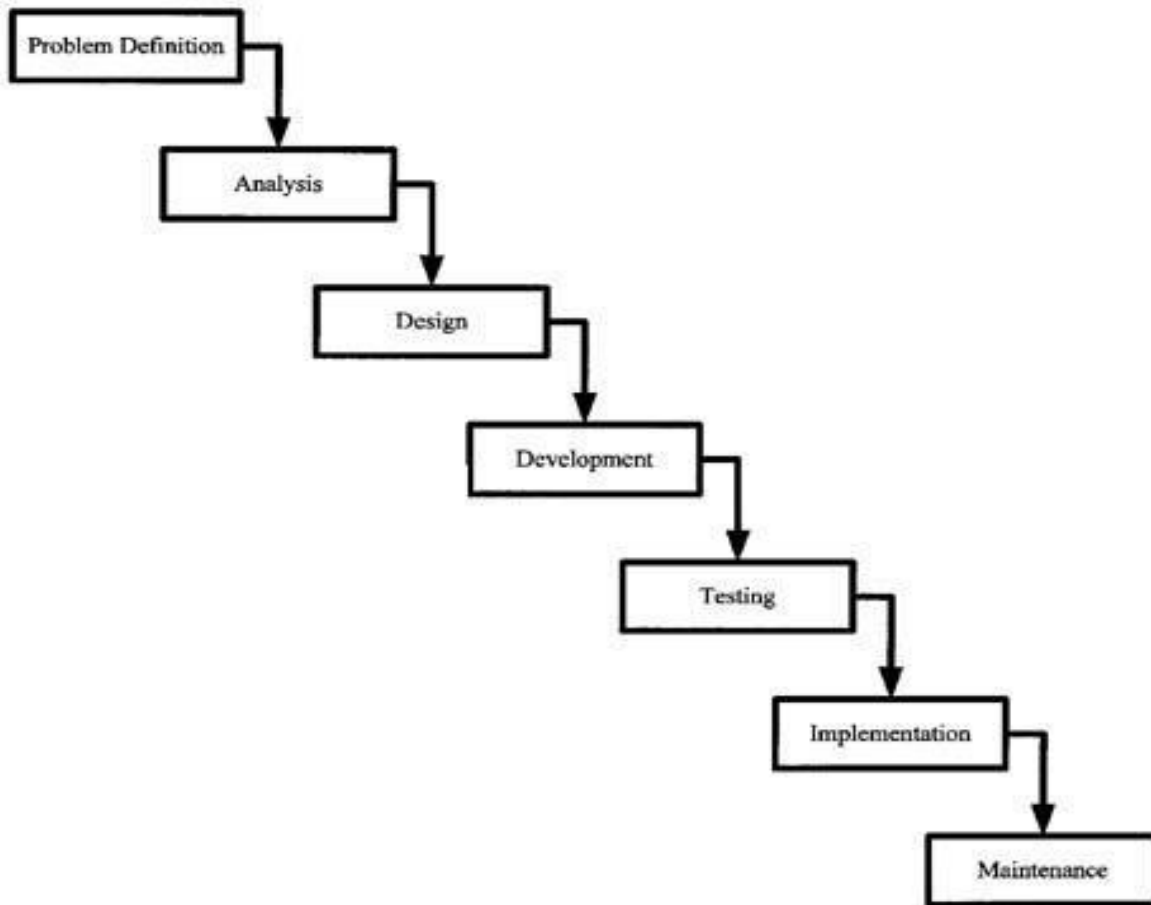
No existing application like this developed until now. The current existing system is offline ration purchase. Presently users have to choose the ration shop manual

3.2 PROPOSED SYSTEM

We are expected to design and develop any program that works efficiently with-out any delays. For a convenience and effective purpose admin panel is given in the website. Gives all the details about the ration shop and the available items based on their card number. Notify when the product bought. Get an estimate receipt with a token number after selecting the items

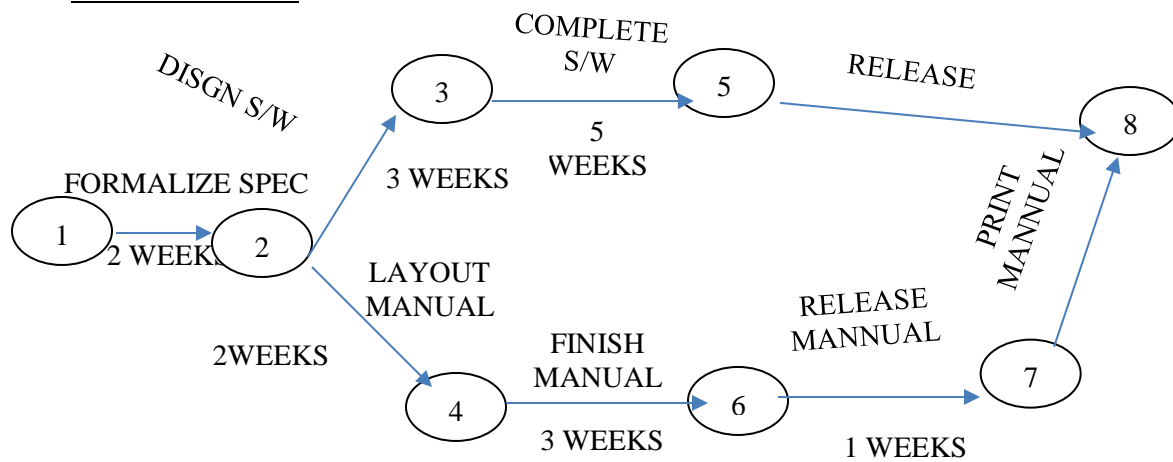
3.3 PLANNING AND SCHEDULING

The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are :

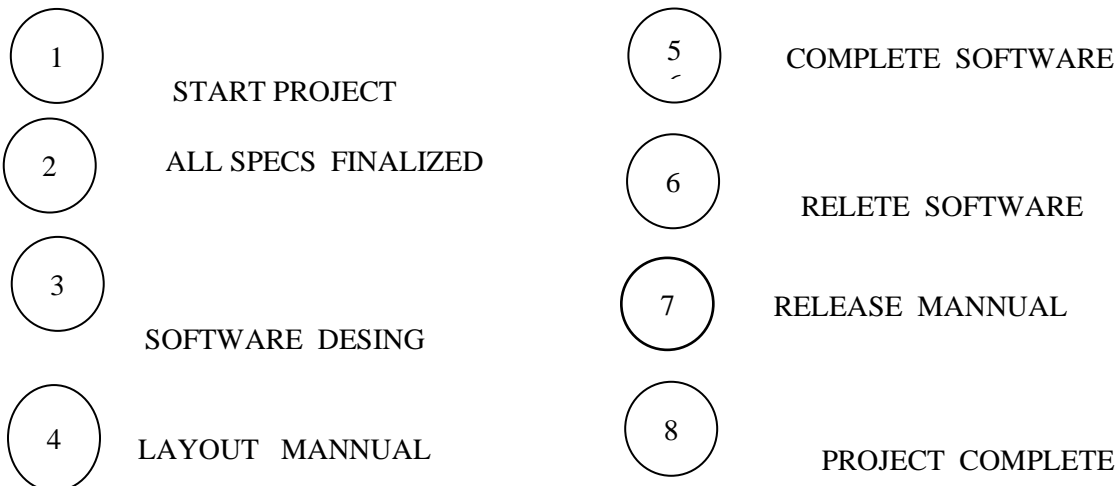


It was a well-planned and well executed job. The details and the requirements for the system was collected from the client itself and all were properly tabulated. Then the next phase was to design all the outputs given by the client to well manageable design which have all the flexibility and which should withstand the current and upcoming technologies. More than one design was made and from that a well acceptable one which got all the functionalities of the user requirements was selected. And we gathered information about the front ends and back ends that will suit for our job.

3.3.1 PERT CHART



STEPS



3.3.2 GANTT CHART

| Tasks | Duration | Month 1 | | | | Month 2 | | | | Month 3 | | | | Month 4 | | |
|--------------------|----------|---------|---|---|---|---------|---|---|---|---------|---|---|---|---------|---|---|
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 |
| Project | 4month | | | | | | | | | | | | | | | |
| System study | 2 weeks | | | | | | | | | | | | | | | |
| System Analysis | 4 weeks | | | | | | | | | | | | | | | |
| Design | 3 weeks | | | | | | | | | | | | | | | |
| Coding and testing | 4 weeks | | | | | | | | | | | | | | | |
| Implementation | 1 weeks | | | | | | | | | | | | | | | |

4. SYSTEM ANALYSIS

System Analysis is conducted with the following objectives.

- Identify the customers need.
- Evaluate the system concept for feasibility.
- Perform economic analysis.
- Technical analysis.
- Allocate function to hardware, software, people, database and other system elements.
- Establish cost and schedule constraints.
- Create system definition that forms the foundation for all subsequent engineering work.

4.1 OVERVIEW

System analysis is concerned with studying the existing system identifying the problems and creation of the requirement specification document. In the system analysis I study about the system to be computerized and the entire problem with the present are pin pointed.

The system analysis was planned and conducted in 3 stages, involving an initial investigation, feasibility and detailed analysis.

4.2 FEASIBILITY ANALYSIS

Feasibility study is the process of determination of whether or not a project is worth doing. Feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report. The contents and recommendations of this feasibility study helped us as a sound basis for deciding how to precede the project. It helped in taking decisions such as which software to use, hardware combinations, etc.

The feasibility analysis starts with the user set of requirements. With this, the existing system is also observed. The next step is to check for the deficiencies in the existing system. By evaluating the above points a fresh idea is conceived to define and quantify the required goals. The user consent is very important for the new plan. Besides that, a set of alternatives and their feasibility is also considered in case of any failure in the proposed system. Thus, feasibility study is an important part in software development. To perform the feasibility study, the software engineer must first analyse the problem at global level. Indeed, the more the problem is understood, the better alternative solutions, the cost and their potential benefits for the user can be identified. Therefore, ideally, one should perform as much analysis of the problem as is needed to do a well-founded feasibility study. Since software developers cannot be sure that the offer will be accepted, they have a limited incentive for investing resource into analyzing the program. On the other hand, if the study produces result that are inaccurate, it may underestimate the resource needed to develop the application and that will result in serious budget problem.

The requirements analysis phase of system and take advantages of the opportunities identified during scope definition and it satisfies the requirement identified in the requirements analysis phase of system development. An estimate is made of whether the identified user needs may be satisfied using current software and hardware technologies. The study will decide if the proposed system will be cost effective from a business point study will decide if the proposed system will be cost effective from a business point of view and if it can be developed given existing budgetary constraints. The key considerations involved in the feasibility analysis are economic, technical, behavioral and operational.

Three tests for project feasibility namely, Technical, Economical, and Operational feasibilities.

4.2.1 TECHNICAL FEASIBILITY

Technical feasibility determines whether the work for the project can be done with the existing equipment, software technology and available personnel. Technical feasibility is concerned with specifying equipment and software that will satisfy the user requirement.

This project is feasible on technical remarks also, as the proposed system is more beneficiary in terms of having a sound proof system with new technical components installed on the system. The proposed system can run on any machines supporting Windows and Internet services and works on the best software and hardware that had been used while designing the

As far as our project is concerned we have the necessary expertise so that the proposed solution can be made feasible. Some projects are initiated with specific deadlines. In our case first we have given three months' time but due to some problems regarding time and the constraints of expertise it has been extended to six months. Now there are some organizational constraints that have not yet given us the opportunity to install the system.

4.2.2 ECONOMICAL FEASIBILITY

Economic feasibility determines whether there are sufficient benefits in sufficient benefits in creating to make the cost acceptable, or is the cost of the system too high. As this signifies cost-benefit analysis and savings. On the behalf of the cost-benefit analysis, the proposed system is feasible and is economical regarding its pre-assumed cost for making a system. Economic feasibility has great importance as it can outweigh other feasibilities because costs affect organization decisions. The concept of Economic Feasibility deals with the fact that a system that can be developed and will be used on installation must be profitable for the Organization. The cost to conduct a full system investigation, the cost of hardware and software, the benefits in the form of reduced expenditure are all discussed during the economic feasibility.

During the economic feasibility test we maintained the balance between the Operational and Economical feasibilities, as the two were the conflicting. For example the solution that provides the best operational impact for the end-users may also be the most expensive and, therefore, the least economically feasible.

As we know that the system development costs are usually one-time costs that will not recur after the project has been completed. For calculating the Development costs we evaluated certain cost categories viz.

- i. Personnel costs
- ii. Computer usage
- iii. Training
- iv. Supply and equipment costs
- v. Cost of any new computer equipment and software.

In order to test whether the Proposed System is cost-effective or not we evaluated it through three techniques viz.

- Payback analysis
- Return on Investment:
- Net Present value

4.2.3 OPERATIONAL FEASIBILITY

Operation feasibility is a measure of how people feel about the system. Operational Feasibility criteria measure the urgency of the problem or the acceptability of a solution. Operational Feasibility is dependent upon determining human resources for the project. It refers to projecting whether the system will operate and be used once it is installed.

If the ultimate users are comfortable with the present system and they see no problem with its continuance, then resistance to its operation will be zero. Behaviorally also the proposed system is feasible. A particular application may be technically and but may fail to produce the forecasted benefits, because the company is not able to get it to work. For the system, it is not necessary that the user must be a computer expert, but any computer operator given a little bit of knowledge and training can easily operate. Our Project is operationally feasible since there is no need for special training of staff member and whatever little instructing on this system is required can be done so quite easily and quickly as it is essentially This project is being developed keeping in mind the general people who one have very little knowledge of computer operation, but can easily access their required database and other related information. The redundancies can be decreased to a large extent as the system will be fully automat

5.SYSTEM DESIGN

5.1 INPUT DESIGN

The input design is the link between the computers and the users. It composes developing specification and procedure for data preparation and those steps that are necessary to put transaction data into a usable form for processing data entry. Input design is one of the most expensive phase of the development of a system. A large number of problem with a system can usually be raised due to the fault input design and method. Therefore needed to say, the input data is the life block of a system and has to be analysed and designed with the most consideration.

- The decision made during input design must help.
- To provide cost effective method of input.
- To achieve highest level of accuracy.

System analysis decide the following input design details like, what data item to input, how the data should be arranged, coded data items and transactions needing validations to detect errors and finally dialogue to guide the users in providing input. The design of input involves identifying the data needed, specifying the characteristic of each data item, capturing and for computer processing and ensuring correctness of data.

5.2 OUTPUT DESIGN

The output design refers to the result and information that are generated by the system. For many users output is the main reason for developing a system and the basis on which they evaluate the usefulness of an application.

The objective of a system leads to the determination of output. The analysis of a system leads to the determination of output. Output of a system can take various forms. The most common reports are, screen displays, printed forms, graphical forms etc. the output also varies in terms of their contents, frequency, timing and format. The users of the output, its purpose and sequence of details to be printed are all considered. The output form a system is the justification for its existence. If the output is inadequate in anyway, the system itself in inadequate. The basics requirements of the output are that it should be accurate, timely and appropriate, in terms of content, medium and layout for its intended purpose. Hence it is necessary to design output so that the objective of the system is met in the best possible manner

5.3 DATAFLOW DIAGRAM

A data flow diagram (DFD) is graphical representation of the “flow” of data through an information system. DFD can also be used for the visualization of the data processing (structured design). A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. Data-flow diagrams can be used to provide the end user with physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to record. How many systems are developed can be determined through a data flow diagram. The circle or bubble represents a transformation process and the label inside the bubble describes the process, using an active verb. Data flows are directed lines that identify the input data flow and output data flows at each process bubble. Data storage is represented by an open-ended rectangle with a label that identifies the data store or file. The square is labelled to identify an external entity that is a source or destination of a data flow.

Data flow diagrams are used to define the flow of the system and its resources such as information's. DFD are way of expressing system requirements in a graphical manner. DFD represents one of the most ingenious tools used for the structured analysis. It has the purpose of clarifying system requirements and identifying major transformation that will become programs in the system design. It is the major point in the design phase that functionality decomposes the requirements specification down the lowest level of details.

Flow diagrams in general are usually designed using designed using simple symbols such as a rectangle, an oval or a circle depicting a processes, data stored or an external entity, and arrows are generally used to depicts the data flow one step to another. Data flow diagrams present the logical flow of information through a system in graphical or pictorial form.

BASIC DATAFLOW DIAGRAM SYMBOLS

- **DATAFLOW**

Used to represent functions or process Packets of data to travel from one point to another. Data may flow from a source to process and from data store or process. An arrow line depicts the flow, with arrow head pointing in the direction of the flow. Arrowhead Indicating the direction in which the data is flowing. Can have two arrow heads when a process is altering existing records in a data stores.

- **PROCESS**

Circle stands for processes that convert data into information. A process represents

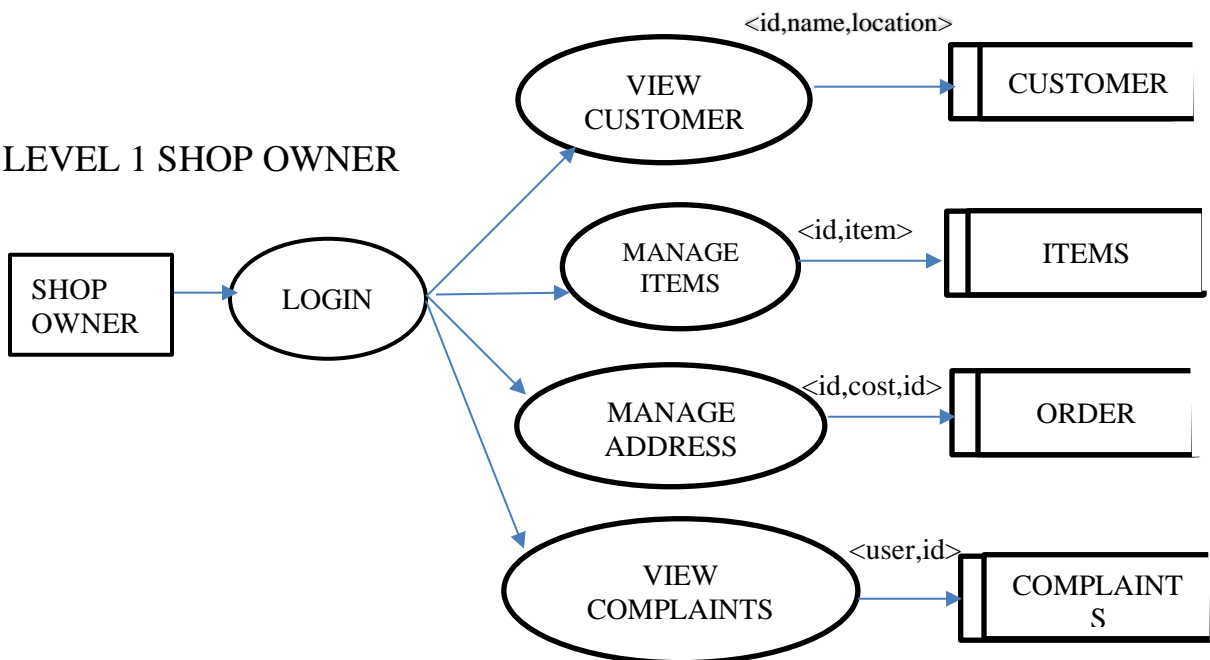
transformation where incoming data flows. A data flow is a route, which enables are changed into outgoing data flows. Must have at least one input and at least one output. When naming process, avoid glossing over them, without really understanding their roles.

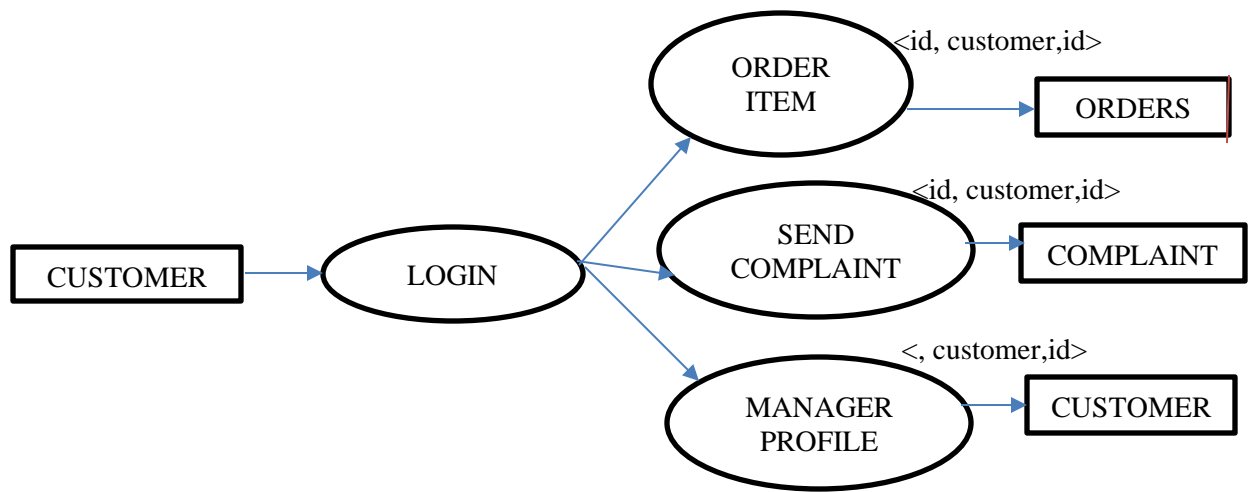
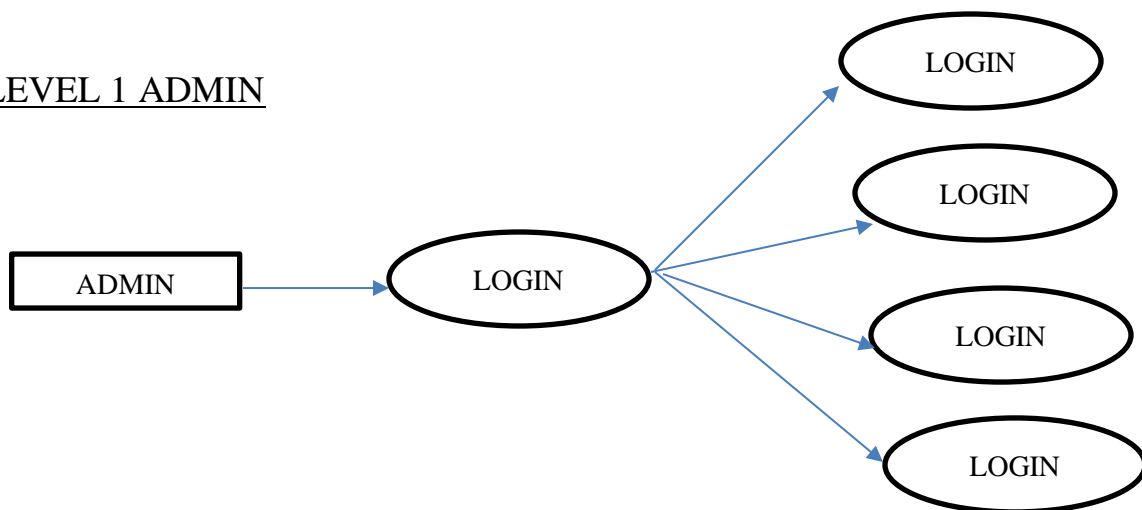
- **DATA STORE**

A data store is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear name. If a process merely uses the content of store and does not alter it.

- **Source/Destination**

A source or sink is a person or part of an organization which enters or receives information from the system, but is considered to be outside the content of data flow model. It is normal for all the information represented with a system to have been obtained from or to be passed onto an external source or recipient. These external entities may be duplicated on a diagram, to avoid crossing dataflow lines, they are duplicated a stripe is drawn across the left hand corner

LEVEL OF DFDLEVEL 1 SHOP OWNER

LEVEL 1 CUSTOMERLEVEL 1 ADMIN

5.4 **DATABASE DESIGN**

A database is a collection of inter related data. A database system provides the enterprise centralized control of its operational data. ORACLE XE is chosen for developing the relevant databases. A database is a collection of stored data organized in such a way that all the user data requirements are satisfied by the database. In order to design an application, it is necessary to design database and the tables used in the system. The overall objective in the development of database technology has been to treat data as an organizational resource and as an integrated whole. Database management system allows data to be protected and organized separately from other sources. During database design the major issues considered are:

- Ease of connectivity to existing database.
- Redundancy.
- Data independence.
- Accuracy.
- Recovery from failure.

TABLE STRUCTURES:SHOP OWNER

| | | |
|-------------|----------|---------|
| id | Not null | Int 3 |
| Name | Not null | Varchar |
| Age | Not null | Int |
| Gender | Not null | varchar |
| Address | Not null | Varchar |
| Location | Not null | varchar |
| Email | Not null | Varchar |
| Password | Not null | Varchar |
| License no. | Not null | varchar |

Customer

| | | |
|----------|----------|---------|
| id | Not null | Int 3 |
| Name | Not null | Varchar |
| Age | Not null | Int |
| Gender | Not null | varchar |
| Address | Not null | Varchar |
| Location | Not null | varchar |
| Email | Not null | Varchar |
| Password | Not null | Varchar |

ADMIN

| | | |
|-----------------|----------------|--------------------|
| <u>username</u> | <u>pk</u> | <u>Varchar(30)</u> |
| <u>password</u> | <u>notnull</u> | <u>Varchar(30)</u> |

STOCK

| | | | |
|----------|-----------------------------|---------|----|
| id | Primary | Int | 3 |
| Itemcode | Notnull | varchar | 30 |
| Itemname | notnull | Varchar | 30 |
| Shop id | Fk(reference shop owner id) | int | 3 |
| quantity | notnull | int | 30 |

CUSTOMER TYPE

| | | |
|-------------------|-------------------------|---------|
| Id | | |
| Userid | Fk(references customer) | |
| Cardtype(APL/BPL) | varchar | notnull |

STOCK

| | | | |
|----------|------------------------------|---------|----|
| Id | Primary | Int | 3 |
| Itemcode | Notnull | Varchar | 30 |
| Itemname | notnull | Varchar | 30 |
| Shop id | Fk(references shop owner id) | Int | 3 |
| quantity | notnull | int | 30 |

CUSTOMERTYPE

| | | |
|-------------------|---------|---------|
| Id | | |
| Userid | FK | |
| Cardtype(APL/BPL) | varchar | notnull |

ORDERS

| | |
|-------------|----|
| Id | pk |
| Customer id | fk |
| Shop id | fk |
| Quantity | |
| Datetime | |
| price | |

COMPLAINTS

| | | |
|------------|--------------|-----|
| User id | Fk(customer) | 100 |
| complaints | varchar | |

6. SYSTEM TESTING AND IMPLEMENTATION

6.1 SYSTEM TESTING

Software testing is defined as the process by which one detects the defects in the software. It is considered as the final opportunity for covert/ rectify and us to detect any defects that were in the software. Testing is a process which is done with the explicit intention of finding errors that make the program fail .In short, system and quality assurance is a review of the software products and related documentation for completion, correctness, reliability and maintainability. The first step in system testing is to prepare a plan that will test all aspects of the system.

System testing is an expensive, but critical process that may take as much as 50% of the budget for program development. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. In a software development project, errors can be injected at any stage during development. Testing is the phase where the errors remaining from the earlier phases also must be detected. The common view of testing is that it is performed to prove that there are no errors in the program. But this is quite difficult since the analyst cannot prove that software is free from all sorts of errors. Therefore, the most useful and practical approach is with the understanding that testing is the process of executing program with the explicit intension of finding errors that is to make the program fail. A successful test can be therefore, one that finds an error.

The philosophy behind testing is finding errors. Test cases are devised with this purpose in mind. A test case is a set of data that the system will process as normal input. However, the data is created with the express intent of determining whether the system will process it correctly. Each test case is designed with the intent of finding errors in the way the system will process it. There are two general strategies for testing software, code testing and specification testing.

CODE TESTING

The code testing strategy examines the logic of the program. To follow this testing method, the analyst develops test cases that result in executing every instruction in the program or module, that is, path through the program is tested. A path is specific combination of conditions that is handled by the program. On the surface, code testing seems to be an ideal method for testing software. However, even if code testing can be performed in its entirety, it does not guarantee against software failures.

This testing strategy does not indicate whether the code meets its specification nor does it determine whether all aspects are even implemented. Code testing also does not check the range of data that the program will accept, even though, when software failures occur in actualcase, it is frequently because users submitted data outside of expected rang

SPECIFICATION TESTING

To perform specification testing, the analyst examines the specifications stating what the program should do, and how it should perform under various conditions. Then test cases are developed for each condition or combination of conditions and submitted for processing. By examining the results, the analyst can determine whether the program performs according to its specified requirements. Software testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before the live operation commences. It is the critical element of software quality assurance and ultimate review of specification, design and coding. The debugging process is the most unpredictable part of the testing procedure.

Syntax Testing

We use syntax testing to eliminate errors in the software. In the system, we have input fields like text, numeric fields. We should allow only numeric fields.

Unit Testing

Unit testing comprises the set of tests performed by an individual programmer prior to integration of the unit into a layer system. These are four categories of tests that can be performed on a program unit.

- Functional Unit
- Performance Unit
- Stress Unit
- Structure Unit

Integration Testing

Integration testing involves bottom up integration, top-down integration and sandwich integration strategy. Bottom-up integration starts with the traditional strategy used to integrate the components of software system into a whole functioning. Top-down integration starts with the main routine and one or two immediate subroutine in the system structure. Sandwich integration is predominantly top-down, bottom-up techniques are used in some modules and systems.

Acceptance Testing

Acceptance testing involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate the implemented system satisfies its requirements. Functional test cases involve exercising the code with nominal files for which the expected outputs are known. Thus the software system developed in the above manner is one that satisfies the user needs, confirms to its requirements and the design satisfactions and exhibits an absence of errors.

Validation Testing

Validation refers to the process of using the new software for the developed system live environment. i.e. new software inside the organization, in order to find out the error the validation phase reveals the failure and the bugs in the developed system. It will come to know about the practical difficulties the system focus. When operated in the true environment by testing the code of the implement software, the logic of the program can be examined. A specifications test is conducted to check whether and specifications starting the program are performing under various conditions. Apart from these test are some special test conducted which are given below:

- **Peak load test:** This determines whether the new system will handle the volume of activities demand. The test has revealed that, the new software for the college is capable of handling demands at the peak time.
- **Storage testing:** This determines the capacity of the new system to store transaction data on a disk or on other files. The proposed software has the required storage space available, because of the use of a number of hard disks.
- **Performance time testing:** This test determines the length of the time used by the system to process transaction data.

WHITE BOX TESTING

White box testing is a test case design method that causes the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

- a. All independent paths within a module have been exercised at least once.

- b. Exercise all logical decisions on their true and false sides.
- c. Execute all loops at their boundaries and within their Operational bounds
- d. Exercise internal data structures to ensure their validity.

BLACK BOX TESTING

Black box testing methods focus on the functional requirement of the software. That is, Black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing attempts to find errors in the following categories.

- a. Incorrect or missing functions
- b. Interface errors
- c. Errors in data structures or external database access.
- d. Performance errors and
- e. Initialization and termination errors

Black box testing is not an alternative to White box testing. Rather it is a complimentary approach that is likely to uncover a different class of errors than white box methods. Unlike white box testing, that is performed early in the testing process. Black box testing to be applied during later stages of testing.

6.2 SYSTEM IMPLEMENTATION

Implementation is a process of ensuring that the information system is operational. Implementation allows the users to take over its operation for use and evaluation. It involves training the users to handle the system and plan for a smooth conversion.

The personnel in the system must know in detail what their roles will be, how they can use the system, and what the system will or will not do. The success or failure of well designed and technically elegant systems can depend on the way they are operated and used.

Systems operators must be trained properly such that they can handle all possible operations, both routine and extraordinary. The operators should be trained in what common malfunctions may occur, how to recognize them, and what steps to take when they come.

Training involves creating troubleshooting lists to identify possible problems and remedies for them, as well as the names and telephone numbers of individuals to contact when unexpected or unusual problems arise.

6.3 **SYSTEM MAINTENANCE**

Software is always evolving and it is never finished as long as it is used; partly to accommodate for the ever changing world we live in. The evolution of your software might be motivated by a variety of reasons; to keep the software up and running, to upgrade to the latest release, enhance features or to rework the system for future maintainability. No matter the motivation, software change is vital for the evolution and success of it. Therefore, software will have to undergo changes, and understanding the different types of changes your software can go through is important to realize that software maintenance is more than just *bug fixing*. In fact, a study suggests that over 80% of software change is attributed to non bug related changes.

There are four categories of software maintenance:

- Corrective maintenance
- Adaptive maintenance
- Perfective maintenance
- Preventive maintenance

CORRECTIVE MAINTENANCE

Corrective change, most commonly referred to as “bugs,” is the most typical change associated with maintenance work. Corrective changes address errors and faults in your software that could affect various areas of your software; design, logic or code. Most commonly, these changes are sprung by bug reports created by users. It is important to note that sometimes problem reports submitted by users are actually enhancements of the system not bugs.

ADAPTIVE MAINTENANCE

Adaptive change is triggered by changes in the environment your software lives in. An adaptive change can be triggered by changes to the operating system, hardware, software dependencies and even organizational business rules and policies. These modifications to the environment can trigger changes within other parts of your software. For example, updating the server, compilers, etc or modifications to shipping carriers and payment processors can affect functionality in your software.

PERFECTIVE MAINTENANCE

Perfective changes refers to the evolution of requirements and features in your existing system. As your software gets exposed to users they will think of different ways to expand the system or suggest new features that they would like to see as part of the software, which in turn can become future enhancements to the system. Perfective changes also includes removing features from a system that are *not effective and functional to the end goal of the system*. Surprisingly, 50-55% of most maintenance work is attributed to perfective changes.

PREVENTIVE MAINTENANCE

Preventive changes refer to changes made to increase the understanding and maintainability of your software in the long run. Preventive changes are focused in decreasing the deterioration of your software in the long run. Restructuring, optimizing code and updating documentation are common preventive changes. Executing preventive changes reduces the amount of unpredictable effects a software can have in the long term and helps it become scalable, stable, understandable and maintainable.

7. FUTURE ENHANCEMENT

The project has very vast scope in future. This project can be similarly integrated with Supplyco, Maveli stores, and other Public distribution system and make them completely online in our technical world

8. CONCLUSION

Being the Final year students, we were able to implement what we learnt in our programming, System Analysis & Software engineering classes into practice. We were able to attain our set objectives, and this helped us gain confidence in writing our own code and our own applications.

We worked as a team, and gained some experience on how professional programmers work in the industry. There is always room for improvement, and this application we created can also be improved. This is especially because we had to create it within a limited time along with other projects, quizzes and Exams. With more time, the software can be improved to include security and cater for different types of users.

Here conclude, this site help user to know all details about the ration shops. So this android application will reduce the dealer communication, manual calculation and it's reduces the time spending in the shops

9. BIBILOGRAPHY

References

1. X. Chen, M. Vorvoreanu, and K. P. C. Madhavan, "Mining social media data for understanding students' learning experiences," *IEEE Transactions on Learning Technologies*, vol. 7, no. 3, pp. 246–259, 2014. View at: [Publisher Site](#) | [Google Scholar](#)
2. L. H. Son and H. Fujita, "Neural-fuzzy with representative sets for prediction of student performance," *Applied Intelligence*, vol. 49, no. 1, pp. 172–187, 2019. View at: [Publisher Site](#) | [Google Scholar](#)
3. S. K. Mohamad and Z. Tasir, "Educational data mining: a review," *Procedia-Social and Behavioral Sciences*, vol. 97, pp. 320–324, 2013. View at: [Publisher Site](#) | [Google Scholar](#)
4. C. Romero and S. Ventura, "Educational data mining: a review of the state of the art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 40, no. 6, pp. 601–618, 2010. View at: [Publisher Site](#) | [Google Scholar](#)
5. S. Bharara, S. Sabitha, and Bansal, "Application of learning analytics

URLS

1. http://delivery.acm.org/10.1145/640000/638180/s2.pdf?ip=115.249.53.74&id=638180&acc=OPEN&key=045416EF4DDA69D9%2E0B5B0253775BA20F%2E4D4702B0C3E38B35%2E6D218144511F3437&_acm_=1525156531_876a762e556fd775eba267a77d75e7e0
2. https://ac.els-cdn.com/S1877050915036182/1-s2.0-S1877050915036182-main.pdf?_tid=44012f66-b408-473b-a41b-69790c3b51e2&acdnat=1525166323_ac1e1a0715d091a4ed3f38939666bfd4

10. APPENDIX

10.1 SAMPLE CODE

Views.py

```

from multiprocessing import context
from os import closerange
from django.http.response import Http404
from django.shortcuts import render, redirect
from django.http import HttpResponseRedirect
from . models import *
from . forms import *
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required
from django.urls import reverse
from django.contrib.auth.forms import PasswordChangeForm
from django.contrib.auth import update_session_auth_hash
from django.contrib import messages
from django.shortcuts import render
from .models import complaints
from .forms import complaintForm
# Create your views here.
def index(request):
    return render(request, 'index.html')
def shop_register(request):
    reg=False
    if request.method=='POST':
        user_form=UserForm(data=request.POST)
        shop_form=ShopRegisterForm(request.POST,request.FILES)
        if user_form.is_valid() and shop_form.is_valid():
            user=user_form.save()
            user.save()
            profile=shop_form.save(commit=False)
            profile.user=user
            profile.save()

            reg=True
            return redirect('user_login')
        else:
            HttpResponseRedirect("invalid form")
    else:
        user_form=UserForm()
        shop_form=ShopRegisterForm()
    return
render(request, 'shop_register.html', {'register':reg, 'user_form':user_form, 'shop_form':shop_form})

```

```

def customer_register(request):
    reg=False
    if request.method=='POST':
        user_form=UserForm(data=request.POST)
        customer_form=CustomerRegisterForm(request.POST,request.FILES)
        if user_form.is_valid() and customer_form.is_valid():
            user=user_form.save()
            user.save()
            profile=customer_form.save(commit=False)
            profile.user=user
            profile.save()

            reg=True
            return redirect('user_login')
        else:
            HttpResponse("invalid form")
    else:
        user_form=UserForm()
        customer_form=CustomerRegisterForm()
    return
render(request,'customer_register.html',{'register':reg,'user_form':user_form,'customer_form':customer_form})
def volunteer_register(request):
    reg=False
    if request.method=='POST':
        shop=Shop.objects.get(user=request.user)
        print(shop)
        user_form=UserForm(data=request.POST)
        volunteer_form=VolunteerRegisterForm(request.POST,request.FILES)
        if user_form.is_valid() and volunteer_form.is_valid():
            user=user_form.save()
            user.save()
            profile=volunteer_form.save(commit=False)
            profile.user=user
            profile.shop=shop
            profile.save()
            reg=True
            return redirect('user_login')
        else:
            HttpResponse("invalid form")
    else:

```

```

user_form=UserForm volunteer_form=VolunteerRegisterForm()
    return
render(request, 'volunteer_register.html', {'register':reg, 'user_form':user_form, 'volunteer_form':volunteer_form})

def registrations(request):
    return render(request, 'registrations.html')

def user_login(request):
    if request.method=="POST":
        username=request.POST.get('username')
        password=request.POST.get('password')

        user=authenticate(username=username,password=password,status=True)
        if user:
            if user.is_active:
                login(request,user)
                return HttpResponseRedirect(reverse('dashboard'))
            else:
                return HttpResponse("admin not approved")
        else:
            return HttpResponse("Invalid username or password")
    else:
        return render(request, 'login.html')

# def user_login(request):
#     if request.method=="POST":
#         username=request.POST.get('username')
#         password=request.POST.get('password')

#         user=Customer.objects.filter(username=username,password=password,status=True)
#         if user:

```

```

#         return render(request, 'dashboard.html')
#     else:
#         return HttpResponse("Invalid username or password")
#     else:

#         return render(request, 'login.html')

def dashboard(request):
    #list=Todo.objects.all()
    return render(request, 'dashboard.html')

@login_required
def user_logout(request):
    logout(request)
    return redirect('index')

def volunteer(request):
    our_volunteer=Volunteer.objects.filter(shop=request.user.shop,availability_status='available')
    print(our_volunteer)
    return render(request, 'our_volunteer.html', {'our_volunteer':our_volunteer})

def update_volunteer(request, volunteer_id):
    my_volunteer=Volunteer.objects.get(id=volunteer_id)
    if request.method=="POST":
        my_volunteer=Volunteer.objects.get(id=volunteer_id)
        update_form=UpdateForm(request.POST, instance=my_volunteer.user)

        update_profile_form=UpdateProfileForm(request.POST, instance=my_volunteer)

        if update_form.is_valid() and update_profile_form.is_valid():
            update_form.save()
            update_profile_form.save()
            messages.success(request, f'Your Account has been Updated')
            return redirect('dashboard')

```

```

else:
    update_form=UpdateForm(instance=my_volunteer.user)
    update_profile_form=UpdateProfileForm(instance=my_volunteer)
    context={
        'update_form':update_form,
        'update_profile_form':update_profile_form
    }
    return render(request,'update_volunteer.html',context)

def delete_volunteer(request,id):
    deleteemp = Volunteer.objects.get(id=id)
    deleteemp.delete()

    messages.success(request,'Record deleted succefully')
    return redirect('volunteer')

def home_servicers(request):
    category=HomeServicerCategory.objects.all()
    if request.method=="GET":
        category=HomeServicerCategory.objects.all()
        location=request.GET.get('location')
        category1=request.GET.get('category')
        try:
            servicer=HomeServicer.objects.filter(location__icontains=location) and
            HomeServicer.objects.filter(category=category1,availability_status='available')
            print(servicer)
            return
        render(request,'home_servicers.html',{'servicer':servicer,'category':category})
        except:
            pass

    return render(request,'home_servicers.html',{'category':category})

def servicer_booking(request,servicer_id):
    servicer=HomeServicer.objects.get(id=servicer_id)

```

```

if request.method=="POST":
    servicer=HomeServicer.objects.get(id=servicer_id)
    booking_form=BookingForm(request.POST)
    if booking_form.is_valid():
        bookings=Booking(customer=request.user.customer,wanted_date=booking_form.cleaned_data['wanted_date'],place=booking_form.cleaned_data['place'])
        bookings.servicer=servicer
        bookings.save()
        return redirect('view_customer_bookings')
    else:
        return HttpResponse("Invalid Form")
    booking_form=BookingForm()
    return render(request,'booking_form.html',{'booking_form':booking_form})

def view_customer_bookings(request):
    customer_bookings=Booking.objects.filter(customer=request.user.customer)

    print(customer_bookings)
    return
render(request,'view_bookings.html',{'customer_bookings':customer_bookings})

def view_servicer_bookings(request):
    servicer_bookings=Booking.objects.filter(servicer=request.user.homeservicer)
    print(servicer_bookings)
    return
render(request,'view_bookings.html',{'servicer_bookings':servicer_bookings})

def delete_booking(request,id):
    delete_booking = Booking.objects.get(id=id)
    delete_booking.delete()
    messages.success(request,'Record deleted successfully')
    return redirect('view_customer_bookings')

def update_booking(request,booking_id):
    booking=Booking.objects.get(id=booking_id)

    print(booking)
    update_booking_form=UpdateBookingForm(instance=booking)
    if request.method=="POST":

update_booking_form=UpdateBookingForm(request.POST,request.FILES,instance=booking)
    update_booking_form.save()
    return redirect('view_servicer_bookings')

```

```

    return
render(request, 'update_bookings.html', {'update_booking_form': update_booking_form})

def update_availability_volunteer(request):
    my_volunteer=Volunteer.objects.get(user=request.user)
    print(my_volunteer.availability_status)
    if request.method=="POST":
        my_volunteer=Volunteer.objects.get(user=request.user)
        update_form=VolunteerAvailabilityForm(request.POST,instance=my_volunteer)
        print(my_volunteer.availability_status)
        if update_form.is_valid():
            update_form.save()

            messages.success(request,f'Your Account has been Updated')
            return redirect('dashboard')
    else:
        update_form=VolunteerAvailabilityForm(instance=my_volunteer.user)

    context={
        'update_form':update_form,

    }
    return render(request, 'update_volunteer_availability.html', context)

def update_availability_servicer(request):
    my_servicer=HomeServicer.objects.get(user=request.user)
    print(my_servicer.availability_status)
    if request.method=="POST":
        my_servicer=HomeServicer.objects.get(user=request.user)
        update_form=ServicerAvailabilityForm(request.POST,instance=my_servicer)
        print(my_servicer.availability_status)
        if update_form.is_valid():
            update_form.save()

            messages.success(request,f'Your Account has been Updated')

    return redirect('dashboard')
    else:
        update_form=VolunteerAvailabilityForm(instance=my_servicer.user)

```



```

context={
    'update_form':update_form,
}
return render(request,'update_servicer_availability.html',context)

def service_workers(request, category):
    try:
        hsc, _ = HomeServicerCategory.objects.get_or_create(servicer_category_name
= category)
        data = HomeServicer.objects.filter(category = hsc)
        return render(request, 'account/service_workers.html', {'data': data})
    except Exception as e:
        print(e)
        raise Http404

def customercomplaints(request):
    if request.method=='POST':
        shopname = request.POST.get('shopname')
        location = request.POST.get('location')
        description = request.POST.get('description')

        complaints(shopname=shopname,location=location,description=description).save()

        return redirect("dashboard")
    return render(request,'customer_complaints.html')

def afterlog(request):
    return render(request,'afterlog.html')

```

Cart_detail.html

```

<!doctype html>
<html lang="zxx">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <title>Ration @ Shop
    </title>

```

```

<!-- Template CSS -->
<link rel="stylesheet" href="/static/assets/css/style-starter.css">
<!-- Template CSS -->
<link
href="//fonts.googleapis.com/css?family=Josefin+Sans:400,400i,500,600,700&display=
swap"
rel="stylesheet">
<link
href="//fonts.googleapis.com/css?family=Lato:300,400,400i,700&display=swap"
rel="stylesheet">
<!-- Template CSS -->
</head>

<body>
<!--w3l-header-->
<section class="w3l-top-header-content">
  <div class="hny-top-menu">
    <div class="container">

      </div>
    </div>
  </section>
  <!--//top-header-content-->
  <header class="w3l-header-nav">
    <!--/nav-->
    <nav class="navbar navbar-expand-lg navbar-light fill px-lg-0 py-0 px-3">
      <div class="container">
        <h1><a class="navbar-brand" href="{% url 'index'
%}">ration<span>Shop</span></a></h1>
        <!-- if logo is image enable this
          <a class="navbar-brand" href="#index.html">
            
          </a> -->
        <button class="navbar-toggler collapsed" type="button" data-
toggle="collapse"
          data-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false"
          aria-label="Toggle navigation">
          <!-- <span class="navbar-toggler-icon"></span> -->

<span class="fa icon-expand fa-bars"></span>
      <span class="fa icon-close fa-times"></span>
    </button>

```

```

<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav ml-auto">
    <li class="nav-item active">
      <a class="nav-link" href="index.html">Home</a>
    </li>
    {% if user.is_authenticated %}
    {% if user.is_superuser %}
    <li class="nav-item">
      <a class="nav-link" href="{% url 'admin:index'
%}">ADMIN</a>
    </li>
    {% endif %}
    <li class="nav-item">
      <a class="nav-link" href="{% url 'dashboard'
%}">Dashboard</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="{% url 'user_logout'
%}">Logout</a>
    </li>
    {% if user.shop %}
    <li class="nav-item">
      <a class="nav-link" href="{% url 'volunteer_register'
%}">ADD VOLUNTEER</a>
    </li>
    {% endif %}
    {% else %}
    <li class="nav-item">
      <a class="nav-link" href="{% url 'registrations'
%}">Register</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="{% url 'user_login'
%}">Login</a>
    </li>
    {% endif %}
  </ul>

</div>
</div>

```

```

        </nav>
        <!--//nav-->
    </header>

{% block content %}
<link href="//maxcdn.bootstrapcdn.com/font-awesome/4.1.0/css/font-awesome.min.css"
rel="stylesheet">

<table id="cart" class="table table-hover table-condensed">
    <thead>
        <tr>
            <th style="width:50%">Product</th>
            <th style="width:10%">Price</th>
            <th style="width:8%">Quantity</th>
            <th style="width:22%" class="text-center">Subtotal</th>
            <th style="width:10%"></th>
        </tr>
    </thead>

    <tbody>
        {% for item in cart.items.all %} {% with product=item.product %}
        <tr>
            <td data-th="Product">
                <div class="row">
                    <div class="col-sm-2 hidden-xs">

                        <a href="{% url 'product_detail' product.id %}">
                            
                        </a>

                    </div>
                    <div class="col-sm-10">
                        <h4 class="nomargin">{{ product.name }}</h4>
                        <p>{{ product.description }}</p>
                    </div>
                </div>
            </td>

            <td data-th="Price"> ₹ {{ product.price }}</td>
            <td data-th="Quantity">

```

```

        <form action="{% url 'cart:cart_add_q' product.id %}">
            <input style="color: brown;" type="number" name="q" value="{{
item.quantity }}" class="form-control text-center" onchange="this.form.submit()">
{% csrf_token %}

        </form>

    </td>
    <td data-th="Subtotal" class="text-center">{{item.get_cost}}</td>
    <td class="actions" data-th="">

        <a href="{% url 'cart:cart_remove' product.id %}" class="btn btn-
danger btn-sm"><i class="fa fa-trash-o"></i></a>
    </td>
</tr>
{% endwith %} {% endfor %}
</tbody>

<tfoot>
    <tr class="visible-xs">
        <!-- <td class="text-center"><strong>cart.</strong></td> -->
    </tr>
    <tr>
        <td><a href="{% url 'product' %}" class="btn btn-warning"><i class="fa
fa-angle-left"></i> Continue Shopping</a></td>
        <td colspan="2" class="hidden-xs"></td>
        <td class="hidden-xs text-center"><strong>Total Rs.
{{cart.get_total_cost}}</strong></td>
        <td><a href="{% url 'orders:order_create' %}" class="btn btn-success
btn-block">Checkout <i class="fa fa-angle-right"></i></a></td>

    </tr>

</tfoot>
</table>

{% endblock %}
<!-- footer-66 -->
<footer class="w3l-footer-66">

<div class="footer-28-main">
    <div class="container">
        <div class="row footer-hny-top">

```

```

        <div class="col-lg-6 footer-logo pl-lg-0">
            <h2><a class="navbar-brand"
href="index.html">Ration<span>Shop</span></a></h2>
            <!-- if logo is image enable this
            <a class="navbar-brand" href="#index.html">
                
            </a> -->
        </div>
        <div class="col-lg-6 main-social-footer-28 pr-lg-0">
            <a href="#facebook"><span class="fa fa-facebook"></a>
            <a href="#twitter"><span class="fa fa-twitter"></a>
            <a href="#google"><span class="fa fa-google-plus"></a>
            <a href="#linkedin"><span class="fa fa-linkedin"></a>
            <a href="#instagram"><span class="fa fa-instagram"></a>
        </div>
    </div>
    <div class="row footer-top-28">
        <div class="col-lg-2 col-md-6 footer-list-28 pl-lg-0">

        </div>
        <div class="col-lg-5 col-md-6 footer-list-28">
            <h6 class="footer-title-28">list of items</h6>
            <ul class="d-grid footer-column-2">
                <li><a href="services.html">Rice</a></li>
                <li><a href="services.html">wheat</a></li>
                <li><a href="services.html">wheat flour(atta)</a></li>
                <li><a href="services.html">sugar</a></li>
                <li><a href="services.html">kerosene</a></li>

            </ul>
        </div>
        <div class="col-lg-2 col-md-6 footer-list-28">

        </div>
        <div class="col-lg-3 col-md-6 footer-list-28 pr-lg-0">
            <h6 class="footer-title-28">Working Hours</h6>
            <ul class="working-hours">
                <li><span>Monday</span> : 8AM - 12 PM |4PM - 8PM</li>

                <li><span>Tuesday</span> : 8AM - 12 PM |4PM - 8PM</li>
                <li><span>Wednesday</span> : 8AM - 12 PM |4PM - 8PM</li>

```

```

            <li><span>Thursday - Friday </span> : 8AM - 12 PM | 4PM -
8PM</li>
            <li><span>Sunday</span> : Closed</li>
        </ul>
    </div>
</div>
<div class="midd-footer-28 align-center">
    <p class="copy-footer-28">© 2022 homespun. All rights reserved |
Designed by <a
        href="https://w3layouts.com">W3layouts</a></p>
    </div>
</div>
<!-- move top -->
<button onclick="topFunction()" id="movetop" title="Go to top">
    <span class="fa fa-angle-up"></span>
</button>
<script>
    // When the user scrolls down 20px from the top of the document, show
the button
    window.onscroll = function () {
        scrollFunction()
    };

    function scrollFunction() {
        if (document.body.scrollTop > 20 ||
document.documentElement.scrollTop > 20) {
            document.getElementById("movetop").style.display = "block";
        } else {
            document.getElementById("movetop").style.display = "none";
        }
    }

    // When the user clicks on the button, scroll to the top of the
document
    function topFunction() {
        document.body.scrollTop = 0;
        document.documentElement.scrollTop = 0;
    }
</script>
<!-- /move top -->

</footer>
<!--//footer-66 -->

```

```

</body>

</html>

<script src="/static/assets/js/jquery-3.3.1.min.js"></script>
<!-- disable body scroll which navbar is in active -->

<!--//-->
<script>
    $(function () {
        $('.navbar-toggler').click(function () {
            $('body').toggleClass('noscroll');
        })
    });
</script>
<!--/scroll-down-JS-->
<!-- for blog carousel slider -->
<script src="/static/assets/js/owl.carousel.js"></script>
<!-- script for banner slider-->
<script>
    $(document).ready(function () {
        $('.owl-one').owlCarousel({
            loop: true,
            margin: 0,
            nav: false,
            responsiveClass: true,
            autoplay: false,
            autoplayTimeout: 5000,
            autoplaySpeed: 1000,
            autoplayHoverPause: false,
            responsive: {
                0: {
                    items: 1,
                    nav: false
                },
                480: {
                    items: 1,
                    nav: false
                },
                667: {
                    items: 1,

```



```

        nav: true
      },
      1000: {
        items: 1,
        nav: true
      }
    }
  })
})
</script>
<!-- //testimonials owlcarousel -->
<script>
  $(document).ready(function () {
    $('.owl-two').owlCarousel({
      loop: true,
      margin: 30,
      nav: false,
      responsiveClass: true,
      autoplay: false,
      autoplayTimeout: 5000,
      autoplaySpeed: 1000,
      autoplayHoverPause: false,
      responsive: {
        0: {
          items: 1,
          nav: false
        },
        480: {
          items: 1,
          nav: false
        },
        667: {
          items: 1,
          nav: false
        },
        1000: {
          items: 1,
          nav: false
        }
      }
    })
  })
</script>
<!-- //script for Testimonials-->

```

```

<!-- //script -->

<script src="/static/assets/js/bootstrap.min.js"></script>

allpayment.html

{% load crispy_forms_tags %}
<!--
Author: W3layouts
Author URL: http://w3layouts.com
License: Creative Commons Attribution 3.0 Unported
License URL: http://creativecommons.org/licenses/by/3.0/
-->
<!DOCTYPE HTML>
<html>
<head>
<title>Product Create</title>

<!--//Metis Menu -->
</head>
<body class="cbp-spmenu-push">
    <div class="main-content">
        <!--left-fixed -navigation-->
        {% include 'dashboard/dashboard_left_nav.html' %}
        <!-- //header-ends -->
        {% include 'dashboard/top_nav.html' %}
        <!-- main content start-->
        <div id="page-wrapper">
            <div class="main-page">
                <div class="forms">
                    <h3 class="title1"></h3>

                    <h4>Sale Table </h4>
                    <div>{% if messages %} {% for mess in messages %}
                        <b style="color: brown;">{{mess}}</b> {% endfor %} {%
endif %}</div>

                </div>
                <table class="table">
                    <thead>
                        <tr>

```

```

        <th>Customer Name</th>
        <th>Amount</th>
        <th>Payment ID</th>
        <th>Order ID</th>
        <th>Total Amount </th>
        <th>Date</th>
        <th>All Total</th>
        <th>Paid Or Not</th>

    </tr>
</thead>
{% for l in payall %}
<tbody>
    <tr>

        <td>{{l.user}}</td>
        <td>{{l.amount}}</td>
        <td>{{l.payment_id}}</td>
        <td>{{l.order_id}}</td>
        <td>{{l.total_amount}}</td>

        <td>{{l.date}}</td>
        <td>{{l.all_total}}</td>
        <td>{{l.is_paid}}</td>

    </tr>

</tbody>
{% endfor %}
</table>
</div>
</div>

</div>
</div>
</div>
</div>

{% include 'dashboard/dashboard_footer.html' %}

```

```

    </body>
</html>

    <!--footer-->

```

category.html

```

{% extends 'index.html' %}{% load static %}{% block title %}product pag{% endblock
%}{% load crispy_forms_tags %} {% block content %}
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<link rel="stylesheet" type="text/css" href="static/css/bootstrap.css">
<link rel="stylesheet" type="text/css" href="/static/css/font-awesome.css">
<link rel='stylesheet' id='camera-css' href='/static/css/camera.css'
type='text/css' media='all'>

<link rel="stylesheet" type="text/css" href="/static/css/slicknav.css">
<link rel="stylesheet" href="/static/css/prettyPhoto.css" type="text/css"
media="screen" title="prettyPhoto main stylesheet" charset="utf-8" />
<link rel="stylesheet" type="text/css" href="/static/css/style.css">
<link rel="stylesheet" type="text/css" href="/static/css/card.css">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
<script type="text/javascript" src="static/js/jquery-1.8.3.min.js"></script>

<link
href='http://fonts.googleapis.com/css?family=Roboto:400,300,700|Open+Sans:700'
rel='stylesheet' type='text/css'>
<script type="text/javascript"
src="static/js/jquery.mobile.customized.min.js"></script>
<script type="text/javascript" src="static/js/jquery.easing.1.3.js"></script>
<script type="text/javascript" src="static/js/camera.min.js"></script>
<script type="text/javascript" src="static/js/myscript.js"></script>
<script src="static/js/sorting.js" type="text/javascript"></script>
<script src="static/js/jquery.isotope.js" type="text/javascript"></script>
<!--script type="text/javascript" src="static/js/jquery.nav.js"></script-->
<!-- Required meta tags -->
<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

```

```

<!-- Bootstrap CSS -->
<!--link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
crossorigin="anonymous">
    <link rel="stylesheet" href="static/css/login_style.css"-->

<script>
    jQuery(function() {
        jQuery('#camera_wrap_1').camera({
            transPeriod: 500,
            time: 3000,
            height: '490px',
            thumbnails: false,
            pagination: true,
            playPause: false,
            loader: false,
            navigation: false,
            hover: false
        });
    });
</script>

<div class="container-fluid mt-50">
    <div class="row">{% if listings %} {% for listing in listings %}
        <div class="col-md-4">
            <div class="product-card mb-30">
                <div class="product-badge bg-secondary border-default text-
body">Stock : {{listing.stock}}</div>
                <a class="product-thumb" href="#" data-abc="true"></a>
                <div class="product-card-body">
                    <div class="product-category"><a href="#" data-
abc="true">{{listing.name}}</a></div>
                    <h3 class="product-title"><a href="#" data-
abc="true">Available Branch: {{listing.branch.location}}</a></h3>
                    <h3 class="product-title"><a href="#" data-abc="true">Contact
Number: {{listing.branch.phonenumber}}</a></h3>

                    <h4 class="product-price"><i class="fa fa-cart">Rs.{{listing.price}} </i></h4>

```

```

        </div>

        <div class="product-button-group">
            <a class="product-button btn-wishlist" href="{%
url 'cart:cart_add' listing.id %}" data-abc="true"><i class="fa fa-shopping-cart"
style="color:blue;"></i><span>Cart</span></a>

            <a class="product-button" href="{% url 'product_detail'
listing.id %}" data-abc="true"><i class="fa fa-angle-
right"></i><span>Details</span></a>
        </div>

    </div>
</div>
{% endfor %} {% else %}<i class="fa fa-heart-o"></i>
<div class="col-md-12">
    <p>No Listings Available</p>
</div>
{% endif %}
</div>
</div>
<div class="row ">
    <div class="col-md-12 ">
        {% if listings.has_other_pages %}
        <ul class="pagination ">
            {% if listings.has_previous %}
            <li class="page-item ">
                <a href="?page={{listings.previous_page_number}} " class="page-
link ">&laquo;</a>

                </li>
            {% else %}
            <li class="page-item disabled ">
                <a class="page-link ">&laquo;</a>
            </li>
            {% endif %} {% for i in listings.paginator.page_range %} {% if
listings.number == i %}

            <li class="page-item active ">

```

```

        <a class="page-link ">{{i}}</a>
    </li>
    {% else %}
    <li class="page-item ">
        <a href="?page={{i}} " class="page-link ">{{i}}</a>
    </li>
    {% endif %} {% endfor %} {% if listings.has_next %}
    <li class="page-item ">
        <a href="?page={{listings.next_page_number}} " class="page-link
">&raquo;</a>
    </li>
    {% else %}
    <li class="page-item disabled ">
        <a class="page-link ">&raquo;</a>
    </li>
    {% endif %}
</ul>
{% endif %}
</div>
</div>{% endblock %}}

```

Index.html

```

<!--
Author: W3layouts
Author URL: http://w3layouts.com
-->

{% load static %}
<!doctype html>
<html lang="zxx">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <title>
        {% block title %}
        RationShop
        {% endblock title %}

    </title>
    <!-- Template CSS -->

```

```

    <link rel="stylesheet" href="/static/assets/css/style-starter.css">
    <!-- Template CSS -->
    <link
href="//fonts.googleapis.com/css?family=Josefin+Sans:400,400i,500,600,700&display=
swap"
    rel="stylesheet">
    <link
href="//fonts.googleapis.com/css?family=Lato:300,400,400i,700&display=swap"
rel="stylesheet">
    <!-- Template CSS -->
</head>

<body>
    <!--w3l-header-->
    <section class="w3l-top-header-content">
        <div class="hny-top-menu">
            <div class="container">

                </div>
            </div>
        </section>
    <!--//top-header-content-->
    {% include 'includes/nav.html' with user=user %}
    {% block content %}
    <!-- //w3l-header -->
    <!--banner-slider-->
    <!-- main-slider -->
    <section class="w3l-main-slider" id="home">
        <div class="banner-content">
            <div class="owl-one owl-carousel owl-theme">
                <div class="item">
                    <li>
                        <div class="slider-info banner-view bg bg2">
                            <div class="banner-info">
                                <div class="container">
                                    <div class="banner-info-bg">
                                        <h6>Fast and efficient</h6>
                                        <h5>The superior choice for home made products
                                            </h5>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </li>
                </div>
            </div>
        </div>
    </section>
</div>

```



```

        </li>
    </div>

    <div class="item">
        <li>
            <div class="slider-info banner-view banner-top2 bg bg2">
                <div class="banner-info">
                    <div class="container">
                        <div class="banner-info-bg">
                            <h6>Fast and efficient</h6>
                            <h5>The superior choice for organic items
                            </h5>

                                </div>
                            </div>
                        </div>
                    </div>
                </li>
            </div>
            <div class="item">
                <li>
                    <div class="slider-info banner-view banner-top3 bg bg2">
                        <div class="banner-info">
                            <div class="container">
                                <div class="banner-info-bg">
                                    <h6>Fast and efficient</h6>
                                    <h5>Quality products for quality
homes</h5>

                                    <a class="btn mt-sm-5 mt-4"
href="services.html">Read More</a>
                                </div>
                            </div>
                        </div>
                    </div>
                </li>
            </div>
        </div>
    </div>
</section>
<!-- /main-slider -->
<!-- //banner-slider-->
<!-- /bottom-grids-->
<section class="w3l-bottom-grids-6">
    <div class="container">

<div class="grids-area-hny main-cont-wthree-fea row">

```

```

        <div class="col-lg-4 col-md-6 grids-feature">
            <div class="area-box">
                <span class="fa fa-bath"></span>
                <h4><a href="#feature" class="title-head">Best
Quality</a></h4>
                <p>the homespun provides best quality home made products
which makes people likes you</p>
            </div>
        </div>
        <div class="col-lg-4 col-md-6 grids-feature">
            <div class="area-box active">
                <span class="fa fa-cogs active"></span>
                <h4><a href="#feature" class="title-head">Learning
corner</a></h4>
                <p>we including a lerning section to create homemades
through uploading tutorial videos</p>
            </div>
        </div>
        <div class="col-lg-4 col-md-6 grids-feature">
            <div class="area-box">
                <span class="fa fa-users"></span>
                <h4><a href="#feature" class="title-head">Labour
Expertise</a></h4>
                <p>we employe a lot of people . Our delivery partners are
innocent and accurate </p>
            </div>
        </div>
    </div>
</div>
</section>
<!-- //bottom-grids-->
<!-- /content-with-photo-1-->

<!-- //content-with-photo-1-->

<!-- /content-6-->

<!-- //content-6-->

<!-- /quote-6-->

<!-- //quote-6-->
<!--/testimonials-->

```

```

    <!--//testimonials-->
    {% endblock %}
    <!-- footer-66 -->
    {% include 'includes/footer.html' %}
    <!--//footer-66 -->
</body>

</html>

<script src="static/assets/js/jquery-3.3.1.min.js"></script>
<!-- disable body scroll which navbar is in active -->

<!--//-->
<script>
    $(function () {
        $('.navbar-toggler').click(function () {
            $('body').toggleClass('noscroll');
        })
    });
</script>
<!--/scroll-down-JS-->
<!-- for blog carousel slider -->
<script src="static/assets/js/owl.carousel.js"></script>
<!-- script for banner slider-->
<script>
    $(document).ready(function () {
        $('.owl-one').owlCarousel({
            loop: true,
            margin: 0,
            nav: false,
            responsiveClass: true,
            autoplay: false,
            autoplayTimeout: 5000,
            autoplaySpeed: 1000,
            autoplayHoverPause: false,
            responsive: {
                0: {
                    items: 1,
                    nav: false
                },
                480: {
                    items: 1,

                    nav: false
                },
                667: {

```

```

        items: 1,
        nav: true
    },
    1000: {
        items: 1,
        nav: true
    }
}
}))
})
</script>
<!-- //testimonials owlcarousel -->
<script>
    $(document).ready(function () {
        $('.owl-two').owlCarousel({
            loop: true,
            margin: 30,
            nav: false,
            responsiveClass: true,
            autoplay: false,
            autoplayTimeout: 5000,
            autoplaySpeed: 1000,
            autoplayHoverPause: false,
            responsive: {
                0: {
                    items: 1,
                    nav: false
                },
                480: {
                    items: 1,
                    nav: false
                },
                667: {
                    items: 1,
                    nav: false
                },
                1000: {
                    items: 1,
                    nav: false
                }
            }
        })
    })

```

```
}  
    })  
  })  
</script>  
<!-- //script for Testimonials-->  
<!-- //script -->  
  
<script src="static/assets/js/bootstrap.min.js"></script>
```

10.2 SCREEN SHOTS

