RV COLLEGE OF ENGINEERING®, BENGALURU-560059

(Autonomous Institution Affiliated to VTU, Belagavi)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



##### SMART RATION CARD SYSTEM

##### Mini - Project Report

###### *Submitted by*

**Siddharth Raikar USN 1RV17CS119**

**Rishab V Arun USN 1RV17CS122**

***in partial fulfillment for the requirement of 5th Semester***

***DBMS Laboratory Mini Project (16CS52)***

**Under the Guidance of**

**Dr Poonam Ghuli, Associate Professor, CSE Department, RVCE**

**Academic year 2019-2020**

**RV COLLEGE OF ENGINEERING®, BENGALURU - 560059**

**(Autonomous Institution Affiliated to VTU, Belagavi)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

**CERTIFICATE**

Certified that the project work titled ‘Smart Ration Card System’ is carried out by **Rishab V Arun (1RV17CS122), Siddharth Raikar (1RV17CS119),** who are bonafide students of RV College of Engineering®, Bengaluru, in partial fulfillment of the curriculum requirement of 5th Semester Database Design Laboratory Mini Project during the academic year **2019-2020**. It is certified that all corrections/suggestions indicated for the internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in all respect laboratory mini-project work prescribed by the institution.

**Signature of Faculty In-charge Head of the Department**

**Dept. of CSE, RVCE**

**External Examination**

**Name of Examiners Signature with date**

**1**

**2**

**ACKNOWLEDGEMENT**

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this project work. I would like to take this opportunity to thank them all.

I deeply express my sincere gratitude to my guide **Dr Poonam Ghuli,** **Associate Professor,** Department of CSE, RVCE, Bengaluru,for her able guidance, regular source of encouragement and assistance throughout this project

I would like to thank **Dr.Ramakanth Kumar P**, Head of Department, Computer Science & Engineering, R.V.C.E, Bengaluru, for his valuable suggestions and expert advice.

First and foremost I would like to thank **Dr. Subramanya. K. N**, Principal, R.V.C.E, Bengaluru, for his moral support towards completing my project work.

I thank my Parents, and all the Faculty members of Department of Computer Science & Engineering for their constant support and encouragement.

Last, but not the least, I would like to thank my peers and friends who provided me with valuable suggestions to improve my project.

**Abstract**

India’s social commitment for providing food security for the poor and needy is answered through the strong network of more than 4.5 Lac Fair Price Shops (FPS) across the country. More than 150 million families purchase ration shop commodities every year. The Public Distribution system in India is however plagued with several malpractices which prevent the benefits from reaching the intended beneficiaries and also result in revenue loss for the Government. There is leakage and pilferage of food grains at each point of the supply chain - from procurement until distribution. It has also been seen that issuance of multiple ration cards to a single person prohibits the proper allocation of food grains to deserved beneficiary.

The manual maintenance of records for issuance of food grains at the Fair Price Shops helps in creating a supportive environment for the FPS owners to indulge in malpractice. Keeping in mind the above areas, it is crucial to strengthen the PDS to ensure adequate supplies, reasonable subsidies and efficient delivery of subsidized food. The Planning commission had stated in its report that, ‘For every Rs 4 spent on PDS only Rs 1 reaches the poor’,’57 % of PDS food grain does not reach the intended people’.

The main focus of this project is to bring order to the current system by eliminating all these problems and providing a secure environment for transactions to take place. It is also a prospect for Digital India. The automation of the current ration system will speed up the process and thus help cope with increasing population.

**Contents**

1. [Introduction](#_TOC_250028) 7
   1. [Objective](#_TOC_250027) 7
   2. [Scope](#_TOC_250026) 7
2. [Software Requirement Specification](#_TOC_250025) 8
   1. [Hardware Requirements](#_TOC_250024) 8
   2. [Software Requirements](#_TOC_250023) 8
   3. [Functional Requirements](#_TOC_250022) 8
      1. [Login Module](#_TOC_250021) for ration card holders
      2. [Purchase Module for ration card holders](#_TOC_250020) 8
      3. Stock Module 9
      4. [Admin Moduke](#_TOC_250019) 9
      5. [FPS owner Module 9](#_TOC_250018)
3. [Entity-Relationship Diagram](#_TOC_250014) 10
4. [Detailed Design](#_TOC_250013) 11
   1. [Data Flow Diagram Level 0](#_TOC_250012) 11
   2. [Data Flow Diagram Level 1](#_TOC_250011) 12
5. [Normalisation 1](#_TOC_250006)3
   1. [First Normal form 1](#_TOC_250005)3
      1. Functional Dependencies 13
      2. [Schema Diagram 1](#_TOC_250004)4
   2. [Other Normal forms 1](#_TOC_250003)4
6. [NoSQL 1](#_TOC_250000)5
   1. [NoSQL architecture 1](#_TOC_250005)5
7. **Conclusion 16**
   1. Conclusion 16

**References 17**

**Appendix:** Screenshots with description **18**

**List of Figures**

1. ER Diagram 10
2. DFD L0 11
3. DFD L1 ………………………………………………………..12
4. Normalised tables.……………………………………………...14
5. NoSQL tables…….……………………………………………..15

**List of Tables**

1. Hardware Requirements 8
2. Software Requirements 8

**Chapter 1**

**Introduction**

The manual maintenance of records for issuance of food grains at the Fair Price Shops helps in creating a supportive environment for the FPS owners to indulge in malpractice. Keeping in mind the above areas, it is crucial to strengthen the PDS to ensure adequate supplies, reasonable subsidies and efficient delivery of subsidized food.

1.1 Objective

Our purpose is to automate the ration card system so that the details are of the transactions that take place in the FPS are available to the government (monitoring body) as well as the ration card user. By automatically updating the stock based on the transactions that take place in the FPS, the distributor has no power to change the stock and sell it to non-ration card holders illegaly.

1.2 Scope

The drawbacks of the existing system include no proper maintenance of the transactions that take place in the FPS, which provides an easy way for the distributor to indulge in malpractice. The ration card user does not have a printed bill of his FPS transactions and the ration card user does not have access to the amount of quota left after withdrawal of ration. In the present system, it is difficult for the government to send circulars/notices to ration card users/FPS owners.

The proposed system eliminates all these drawbacks by providing a secure environment for transactions to take place in the FPS. The ration card user will receive a bill of his transactions. The details of ration stock in the FPS is automatically updated when the ration card user withdraws ration. This provides no scope for the FPS owner to indulge in malpractice. The proposed system also allows the government to monitor the distributors easily. The ration card user is also allowed to submit reviews on FPS, he can also complain against any malpractice in FPS.

**Chapter 2**

**Software Requirement Specification**

A Software Requirement Specification (SRS) is a description of a software system to be developed. Software requirements specification (SRS) is important for developers because it minimizes the amount of time and effort developers have to expend to achieve desired software goals. Software requirements specifications establish the basis for an agreement between customers and contractors or suppliers on how the software product should function.

2.1 Hardware requirements

For the running of the application in a system the following hardware requirements have to be met.

Table 1: Hardware Requirements

|  |  |
| --- | --- |
| Basic | Internet access |
| Core | Dual/Quad core recommended |
| RAM | 2GB or more |
| CPU | Intel core i-series or equivalent AMD processor |
| Hard Drive | 1GB and above |

2.2 Software requirements

For the running of the application in a system the following software requirements have to be met.

Table 2: Software Requirements

|  |  |
| --- | --- |
| Operating System | Windows 7 and above, Linux, Mac OS |
| Programming Language | Python, JavaScript |
| User Interface (Front End) | HTML, CSS, |
| Database (Back End) | MySQL, MongoDB, Django |

2.3 Functional requirements

## 2.3.1 Login module for Ration card holders

The concept of a user id and password is a cost effective and efficient method of authentication between a user and a computer system. In this module, the system verifies the user’s login details with the details stored in the database when the user was registered.

The user will login by providing the user-id (Aadhar number) and password, if the details are correct then he can proceed to main page else an error message will pop up.

## 2.3.2 Purchase module for Ration card holders

The ration distribution system is being made automatic by maintaining the database at one main control station and updating the database so that the shopkeeper does not cheat the poor people.

Users have to enter the amount of ration to be withdraw. The system checks the user’s account, if the user has sufficient balance (ration quota) left he will get the required ration and the transaction details will be updated in the database. The user’s quota will reduce if he withdraws ration. The user will get a bill detailing the transaction made.

2.3.3 Stock module

The FPS owner has a stock of food grains and other ration items to be distributed to the ration card holder’s family. The FPS owner will be able to modify the stock details. As ration card holders withdraw ration, the FPS stock will get modified.

2.3.4 Admin module

In this module the officials working in the PDS department are able update/modify the details of ration card holders. The admin is able to add/remove distributors and ration card holders. The admin can access the reviews given by the users regarding the fair price shops. The admin can send circulars to ration card holders and FPS regarding any new announcements and changes to the system.

2.3.5 FPS owner module

The FPS will login by providing the user-id (FPS number) and password. After successful login, the FPS owner will be able to view the details of the stock present in the FPS. The FPS owner will be able to view the transaction details of the shop. The FPS will be able to modify the stock details.

**Chapter 3**

**Entity relationship diagram**

ER (Entity Relationship) model is a popular high level conceptual data model used for conceptual design of database applications. The ER model describes data as entities, relationships and attributes. The basic object that the ER model represents is an entity, which is a thing in the real world with an independent existence. The ER model has attributes for entities and relationship between entities. ER diagram is the diagrammatic representation of the ER model

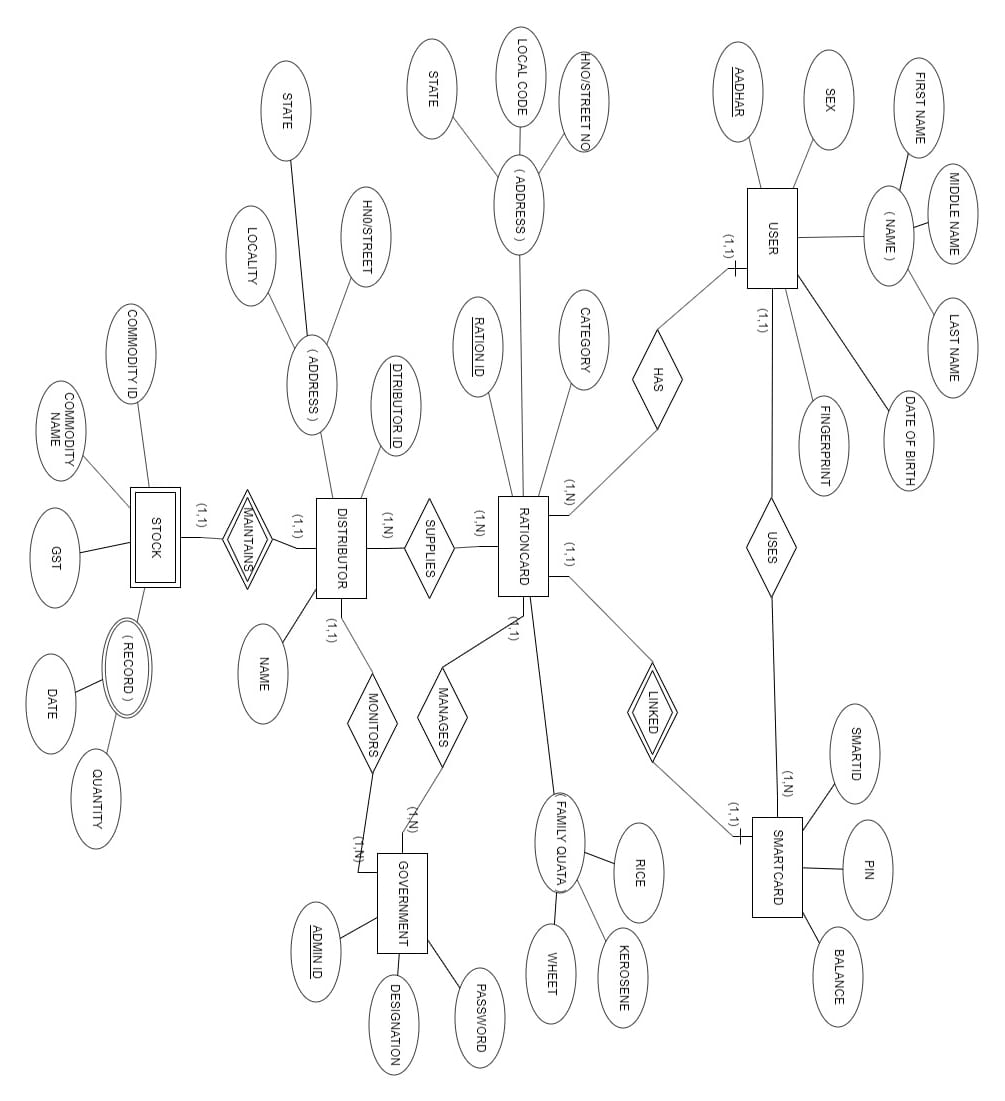


Figure 1: ER Diagram

**Chapter 4**

**Detailed design**

The detailed design of the project involves Data Flow Diagrams which show the detailed flow of data and information flow in the whole system. The DFD consists of 3 levels-

● DFD Level 0 - General Information Flow

● DFD Level 1 - Listing the major processes in the whole system.

● DFD Level 2 - Detailed flow of each process in the system.

4.1 DFD Level 0

The DFD at level 0 gives the flow of data at a highly abstract level. As shown in Figure 2 there are three types of users Admin (government), ration card user and the distributor. The details of data flow is show by level 1 DFD.

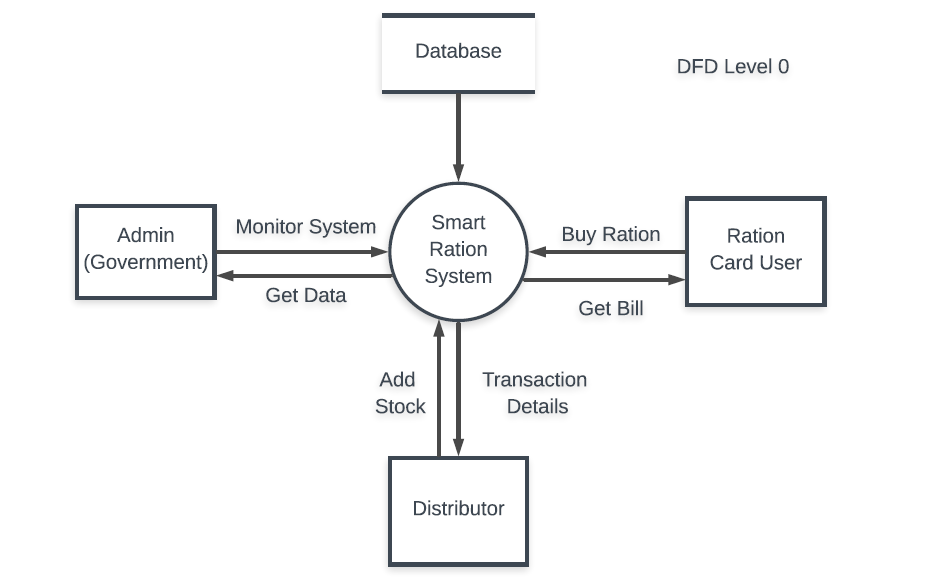


Figure 2: DFD Level 0

4.2 DFD Level 1

The DFD Level 1 shows the different modules in the system. It shows a detailed design of the system. The ration card user logs in using the login module by providing the login details (which will be verified with the database). The ration card user makes the transaction in the FPS and gets the bill of the transaction. The transaction details are reflected in the distributor’s database. The distributor can add stock and view the transactions of his shop. The admin can add/remove distributors and also monitor the distributors.

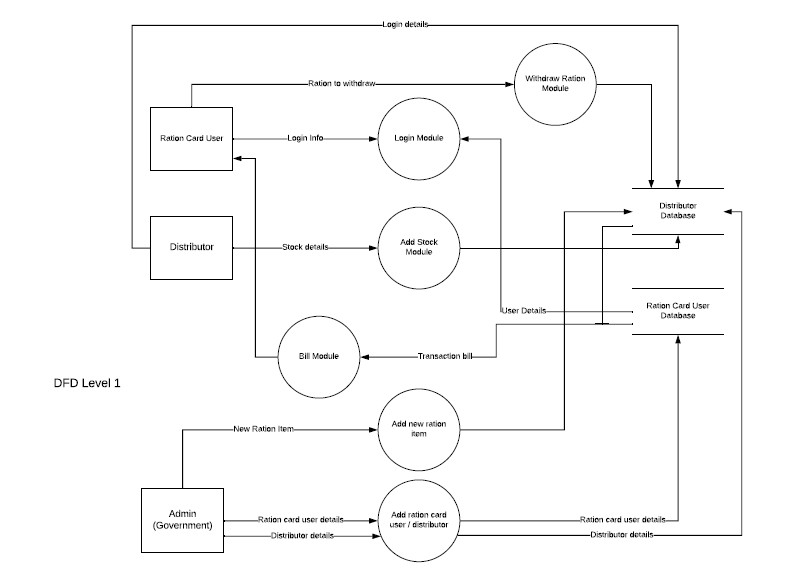


Figure 3: DFD Level 1

**Chapter 5**

**Normalisation**

**Normalisation** is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly. The most commonly used normal forms are:

* First normal form(1NF)
* Second normal form(2NF)
* Third normal form(3NF)
* Boyce & Codd normal form (BCNF)

5.1 First Normal Form

First normal form (1NF) is a property of a relation in a relational database. A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain.

The conditions to be met are:

1. Each table cell should contain a single value.
2. Each record needs to be unique.

5.1.1 Functional dependencies

A functional dependency is a [constraint](https://en.wikipedia.org/wiki/Relational_database#Constraints) between two sets of attributes in a [relation](https://en.wikipedia.org/wiki/Relation_(database)) from a database.

Given a relation R, a set of attributes X in R is said to functionally determine another set of attributes Y, also in R, (written X → Y) if, and only if, each X value in R is associated with precisely one Y value in R

**USER**

User\_Id-> Username, Password

**PERSON**

Person\_Id-> First name, Middle name, Last name, Aadhar number, Date of birth, Age, Sex, Photo, Rationcard\_id, User\_id.

**RATIONCARD**

Rationcard\_no-> Pin, Locality code, Street, House\_no, Category, Email, Password

**TRANSACTION**

Transaction\_id-> Distributor\_id, Rationcard\_id, Rice quantity, Wheat quantity, Kerosene quantity, Date, Amount, Time

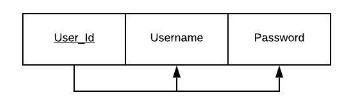
**CIRCULAR**

Circular\_id-> Name, File, date

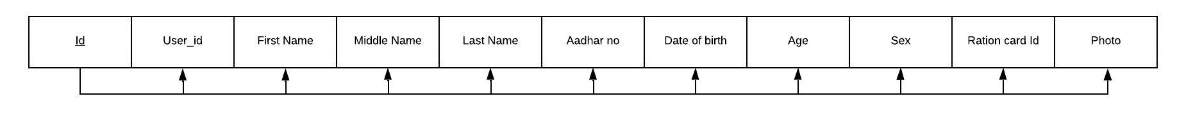
5.1.2 Schema diagram

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.

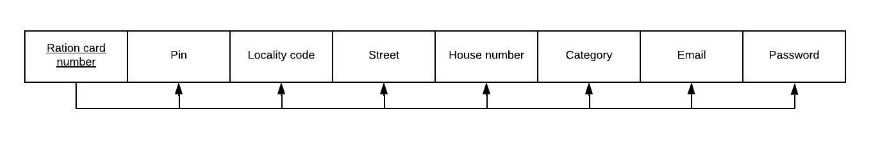
User



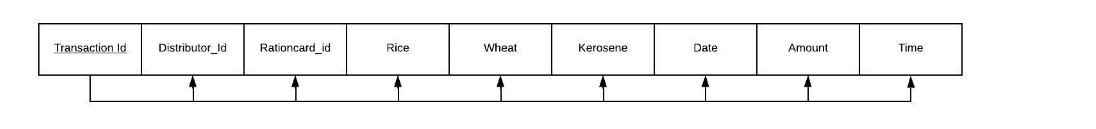
Person



Ration-card



Transaction



Circular

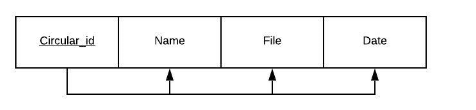


Figure 4: Normalised tables

5.2 Other Normal Forms

The tables are already in 2NF, 3NF and BCNF. Since there are no multi valued dependencies it is also in 4NF.

**Chapter 6**

**No SQL**

NqSQL is NoSQL, which stands for "not only SQL," is an alternative to traditional relational databases in which data is placed in tables and data schema is carefully designed before the database is built. NoSQL databases are especially useful for working with large sets of distributed data.

This project uses MongoDB to for the distributor side of the database. All the transactions in the FPS and the distributor side of the database including details of stock are stored in the NoSQL database.

The NoSQL part is integrated with the system using Djongo (Django + MongoDB). MongoDB is used for the distributor side of the database because MongoDB supports nesting of collections (tables) which is very convenient in our project to represent the distributor database. The ration card user’s reviews on the FPS are also stored in NoSQL database because NoSQL has the provision of providing extra fields (columns) to certain objects (tuples) of a collection.

6.1 NoSQL database architecture

In this project we have used array field of NoSQL to implement one to many relationships. The array field and embedded documents are used useful for nesting relations. The primary document is distributor in which stock, input entry, complain and record.

Figure 5: NoSQL architecture

**Chapter 7**

**Conclusion**

7.1 Conclusion

The project consists of a database system to manage the public distribution system. This helps in solving the problems that exist in the current ration card system which include improper maintenance of records and transactions that occur in the FPS and illegal selling of ration to non-ration card holders by the distributors. Our project solves these problems by providing a secure environment for the transactions in the FPS to take place as well as help the government to monitor the activities of FPS.

It was successfully implemented and the results were produced. The complete project was as per software requirement specifications and it completed the tasks which were there in synopsis and software requirements specification of the project.

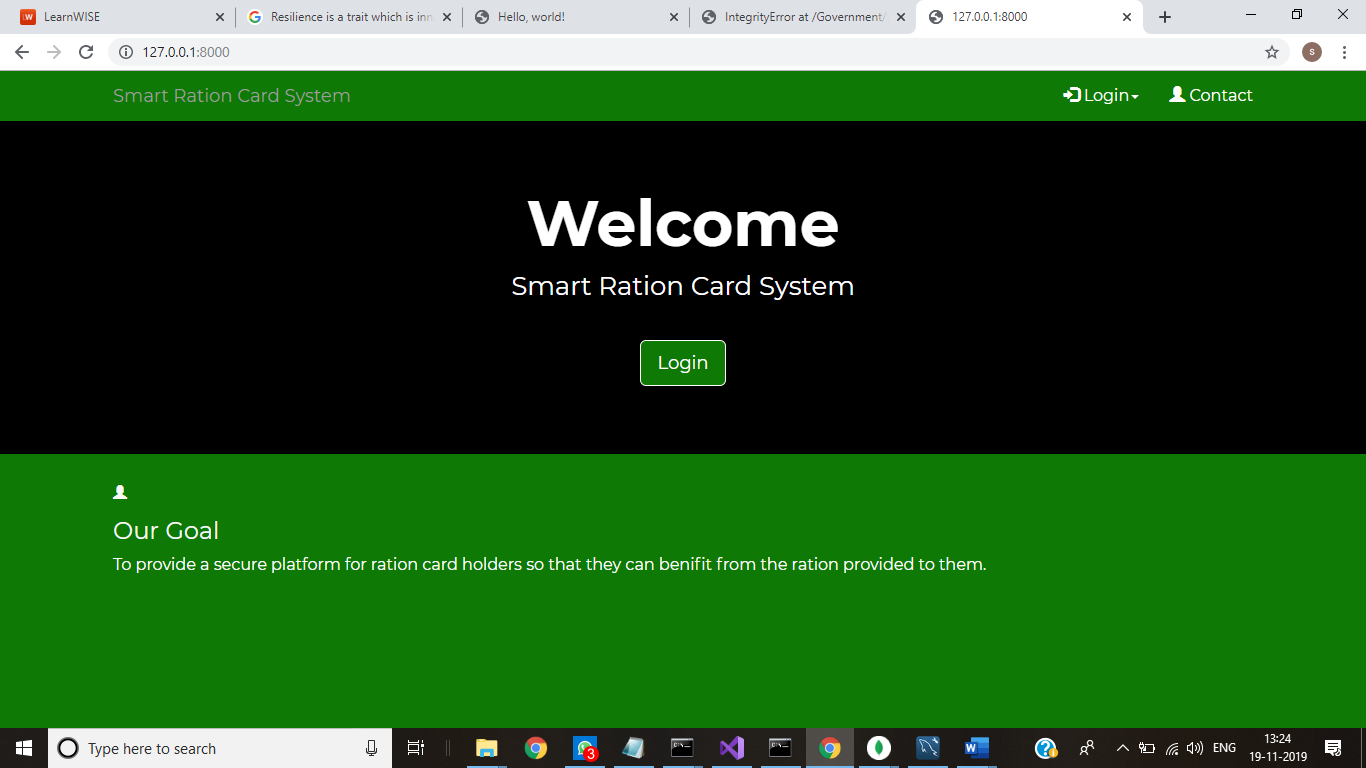
Future scope for this project includes including the biometric details of the ration-card holders in the database for more secure authentication. The godown system of PDS can also be atomized, the stock of ration in the godown can be automatically updated in the database using sensors which monitor stock (weight sensors) in the godown. This lowers the probability of illegal distribution of grains and adulteration of food grains.

**References**

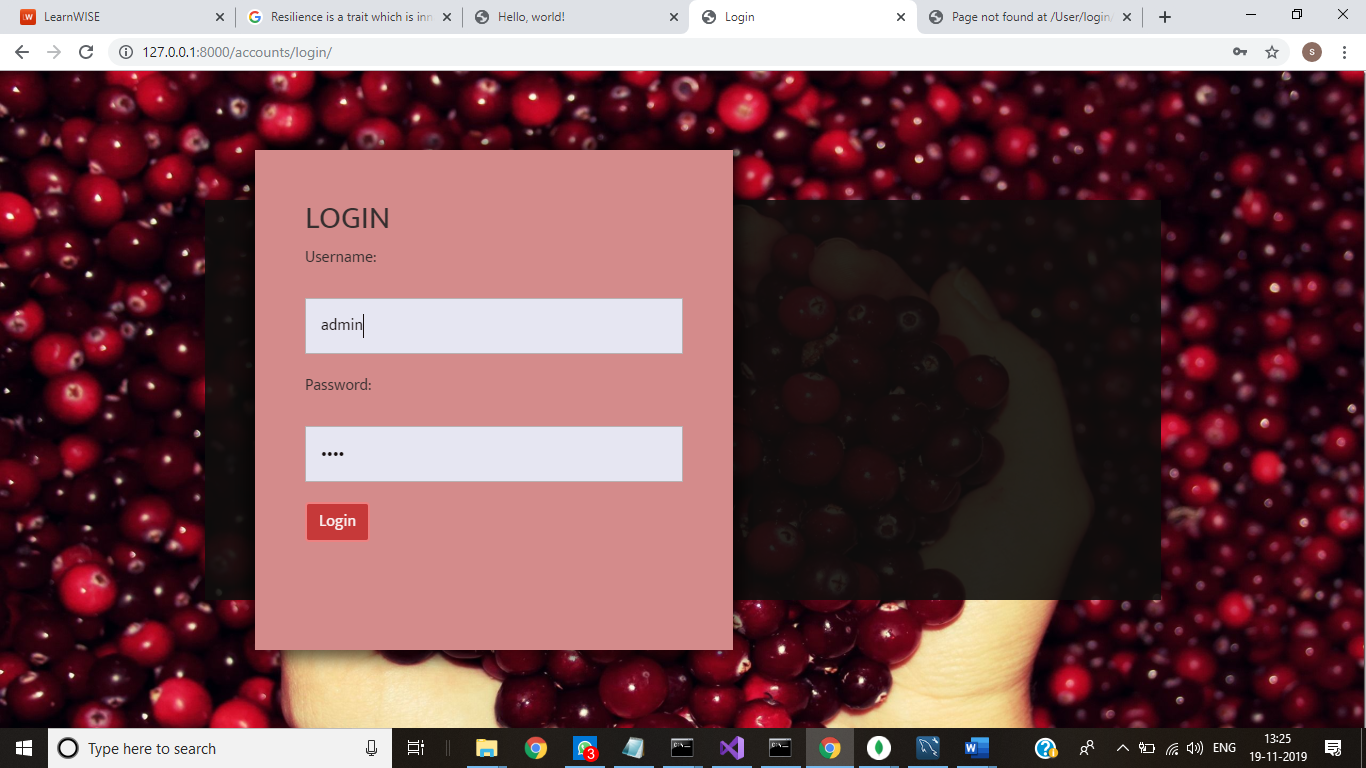
* <https://docs.djangoproject.com/>
* <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django>
* h[ttps://www.elastic.co/what-is/elasticsearc](http://www.elastic.co/what-is/elasticsearch)h
* h[ttps://www.elastic.co/guide/en/elasticsearch/reference/curren](http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-bulk.html)t/do[cs-bulk.h](http://www.elastic.co/guide/en/elasticsearch/reference/current/docs-bulk.html)tml
* h[ttps://www.elastic.co/guide/en/elasticsearch/reference/current/full-text-](http://www.elastic.co/guide/en/elasticsearch/reference/current/full-text-) queries.html
* h[ttps://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-](http://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-)simple-query-string-query.html
* <https://www.w3schools.com/bootstrap4/default.asp>

**Appendix: Screenshots with description**

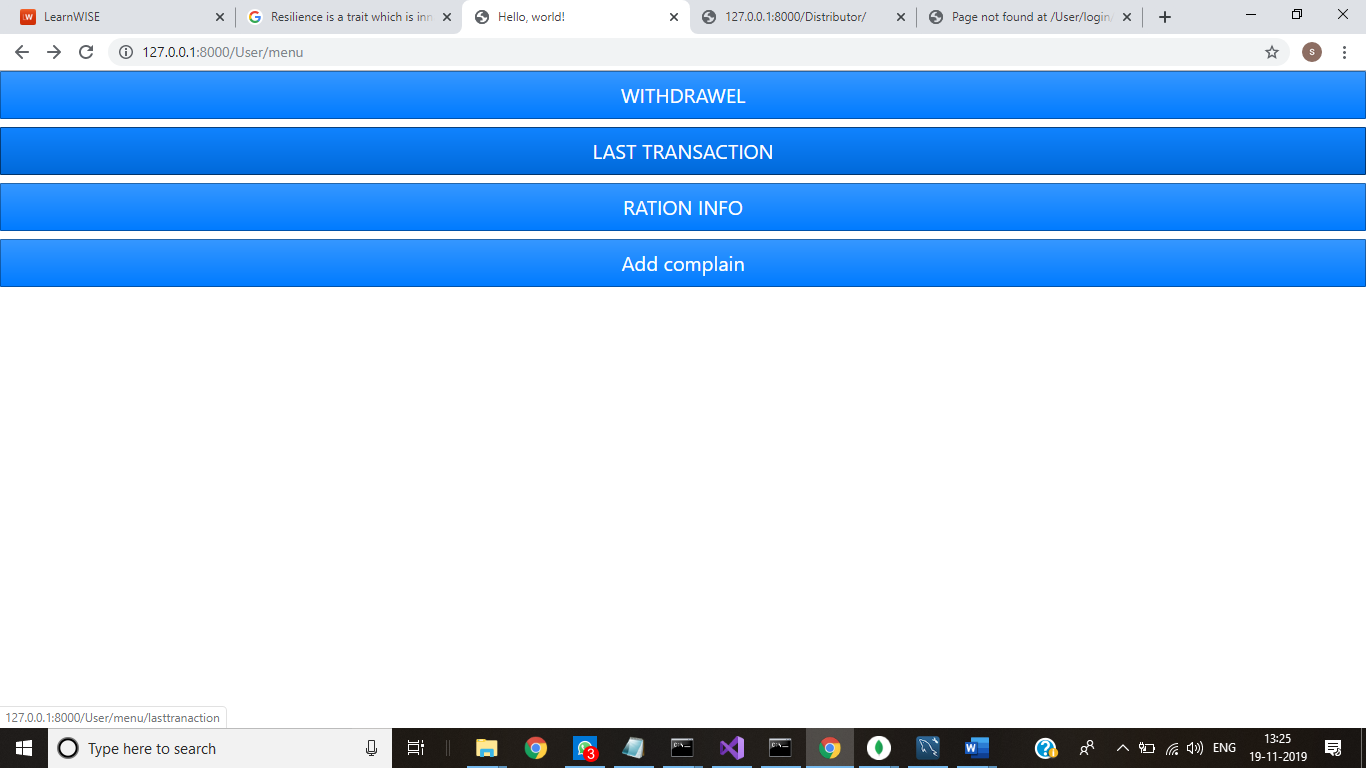
1 Home Page

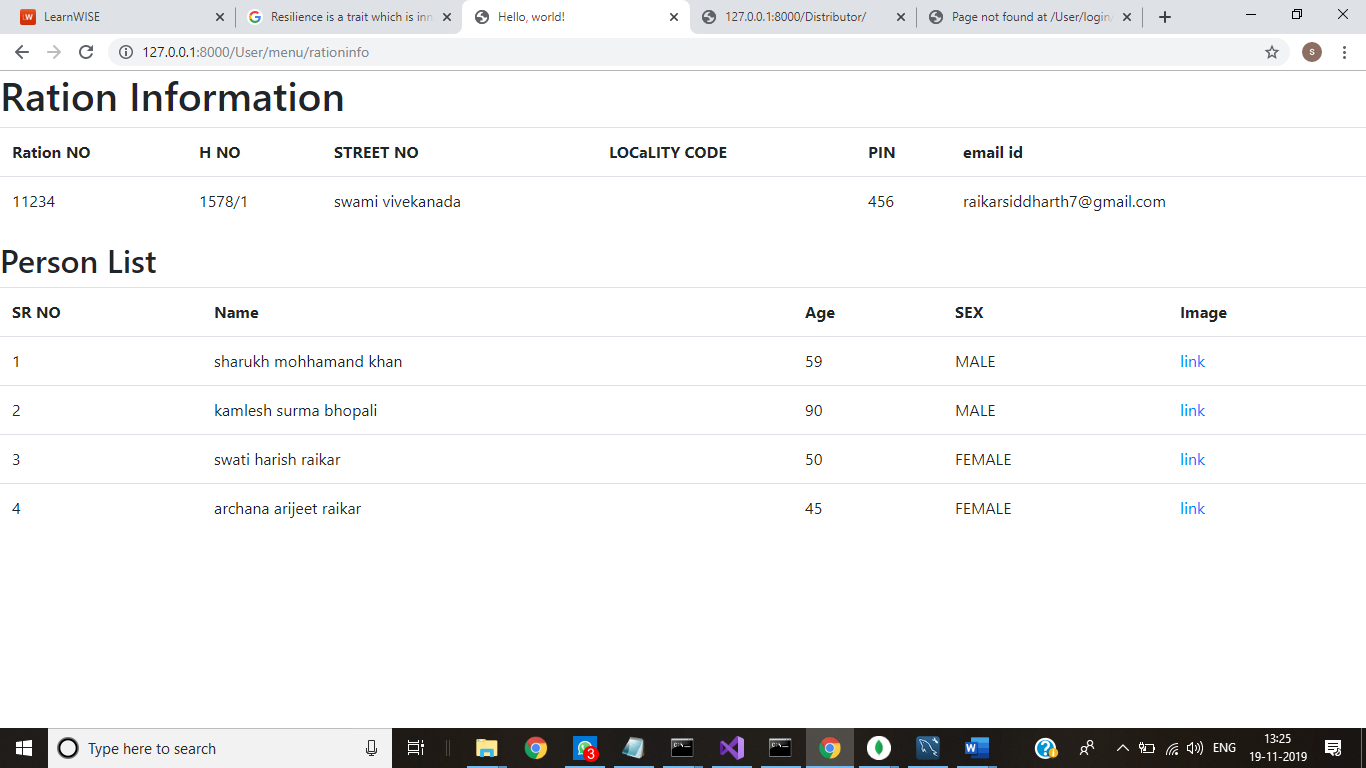


2 Login Page

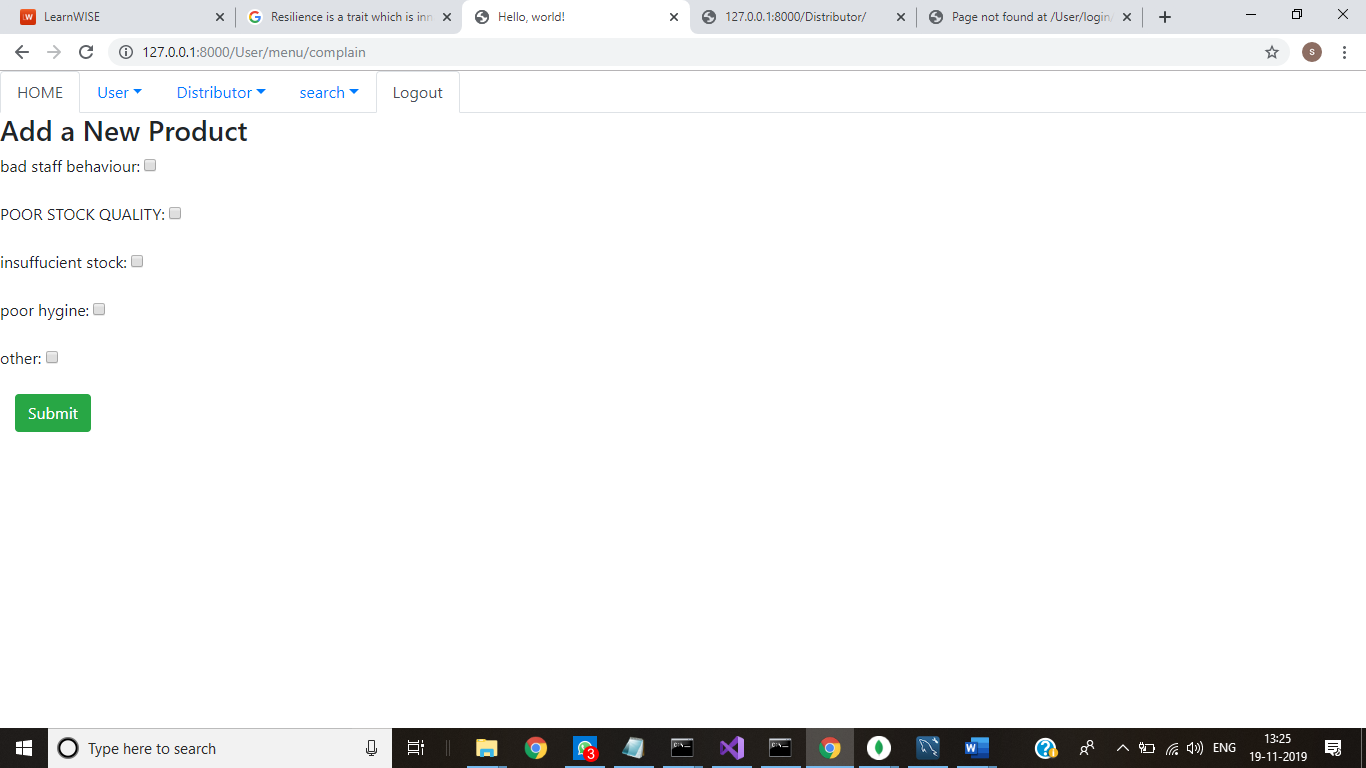


3 Ration card user module





4 Distributor Module



5 Admin Module

