

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JnanaSangama”, Belgaum - 590018



A DBMS MINI PROJECT REPORT ON
“GROCERIES SALES AND INVENTORY MANAGEMENT SYSTEM”

Submitted in the partial fulfillment of the requirement for the fifth semester of
BACHELOR OF ENGINEERING

In
COMPUTER SCIENCE & ENGINEERING

By
VINUTHASHREE N N (1RR18CS159)

Under the guidance of

Mr. Manjunath S R
Assistant Professor,
Dept. of CSE, RRCE



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJARAJESWARI COLLEGE OF ENGINEERING
MYSORE ROAD, BANGALORE-560074

(An ISO 9001:2008 Certified Institute)

(2020-21)

RAJARAJESWARI COLLEGE OF ENGINEERING
MYSORE ROAD, BANGALORE-560074

(An ISO 9001:2008 Certified Institute)

(Affiliated to Visvesvaraya Technological University, Belgaum)



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that mini project work entitled

“GROCERIES SALES AND INOVENTORY MANEGMENT SYSTEM”

Carried out by **VINUTHASHREE N N(1RR18CS159)**

The students of **“RajaRajeswari College of Engineering”** in partial fulfillment for the fifth semester of **Bachelor Of Engineering in Computer Science & Engineering** of the Visvesvaraya Technological University, Belgaum during the year **2020–2021**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the fifth semester.

.....

Signature of Guide

[Mr. Manjunath S R]

Assistant Professor, Dept. of CSE

RRCE, Bangalore

External Viva-Voce

Examiners:

1. _____

2. _____

.....

Signature of HOD

[Dr. Usha S]

Prof. & HOD, Dept. of CSE

RRCE, Bangalore

Signature

PAGE INDEX		
Chapters	Topic	Page No
ACKNOWLEDGEMENT		
1. INTRODUCTION.....		
	1.1 Introduction about DBMS	1
	1.2 Project Background	1
	1.3 Project Need	1
	1.4 Application and Advantages	1
	1.5 Project Implementation	1
2. DESIGN.....		
	2.1 ER diagram	3
	2.2 Relational Schema diagram	5
	2.3 Schema diagram	7
	2.4 Normalisation	8
3. IMPLEMENTATION.....		
	3.1 Software and hardware requirements	11
	3.2 Create table commands	13
	3.3 Table structure	16
	3.4 Functionality	18
4. RESULT.....		
	4.1 Snapshots	21
5. CONCLUSION.....		26

ACKNOWLEDGEMENT

I am truly thankful and convey my sincere gratitude to the honorable principal **Dr. T Chandrashekar**, Rajarajeshwari College of Engineering, Bangalore.

I convey my sincere gratitude to **Dr.UshaSakthivel**,HOD, Department of Computer Science & Engineering for her meticulous support, continuous co-operation, valuable suggestion and encouragement during the development of the project. I also extend my thanks for her valuable guidance to imbibe the requisite knowledge for success of our project.

I convey my sincere gratitude to **Mr. Manjunath S R**, Assistant Professor, Department of Computer Science & Engineering, for his meticulous support continuous co-operation, valuable suggestion and encouragement during the development of the project. I also extend my gratitude for her guidance to imbibe the requisite knowledge for success of my project, as an internal guide. I am very much obliged.

I also thank my parents who have encouraged me and supported me in every stage of development of this project.

Last but not the least, my wishes to the entire Computer Science Department for their help and guidance, encouragement, inspiration and co-operation at all stages of the development of this project without which this project wouldn't have been a success.

Finally, I express my heartfelt gratitude to all those who helped me to complete the project work successfully by providing support, suggestions, advise, guidance and much needed encouragement.

VINUTHASHREE N N(1RR18CS159)

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION ABOUT DBMS

The DBMS manages three important things: “DATA”, “DATABASE ENGINE” that allows data to be accessed, locked and modified and “DATABASE SCHEMA” which defines the database’s logical structure. These three foundational elements help provide concurrency, security, data-integrity and uniform administration procedures.

1.2 PROJECT BACKGROUND

Our sales_and_inoventory project is developed for customers and shop keepers. It is accessible by the cashier and admin. Since it is a database oriented system, it allows insertion, updating, deletion, and joining of orders. User can enroll into the course which he/she interested in by making payment. Admin can insert, modify, delete and generate receipt. Customer can be added either by admin or by the Cashier. Customer may then select there wished products. Finally the receipt is generated by the cahier and admin which can also be printed for the reference. Details of the customer, products buy by the customer And payments are saved in the database.

1.3 PROJECT NEED

This project is needed to keep record of all users that enroll into the system. It also keeps record of all the transaction and roles enrolled by the users. It has important details such as the Id to identify the transaction, date and amount of sales.

1.4 APPLICATION AND ADVANTAGES

Managing an Sales_and_inoventory website has lot of responsibilities. It is not easy to manage users, provide quality products, great offers with qualified products and user-friendly at the same time. A great deal of attention has to be paid in ensuring quality, revenue and profit. To ensure the system will have a long run, equal importance and investment has to be given to technology. Having arecord of sales will let you know about the success rate of the Sales_and_inoventory website. It will help you find the best products with best brand in Sales_and_inoventory website

1.5 PROJECT IMPLEMENTATION

We have implemented our project using MYSQL, Sublime Text3 and XAAMP.

We have used MYSQL and PHP as it is backward compatible i.e. it allows businesses to upgrade their systems without a complete change of database systems. It is efficient and has low-cost updates. Sublime Text3 is a powerful Frontend tool. It supports sophisticated yet simplified for developing Application Framework and also for binding which has been used in our project.

CHAPTER 2

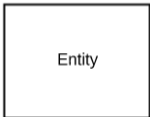
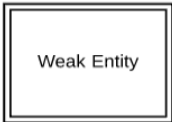




DESIGN

2.1 ER DIAGRAM

An **entity relationship** model, also called an **entity-relationship (ER) diagram** that displays the relationship of entity sets stored in a database. In other words, ER diagrams help to explain the logical structure of databases.

ER diagram of Online learning management system is shown in figure number 2.1.2.

The symbols used in ER-Diagram is as follows.

Symbols	Name
	Strong entity
	Weak entity
	Relationship
	Weak relationship
	Attribute
	Multivalued attribute

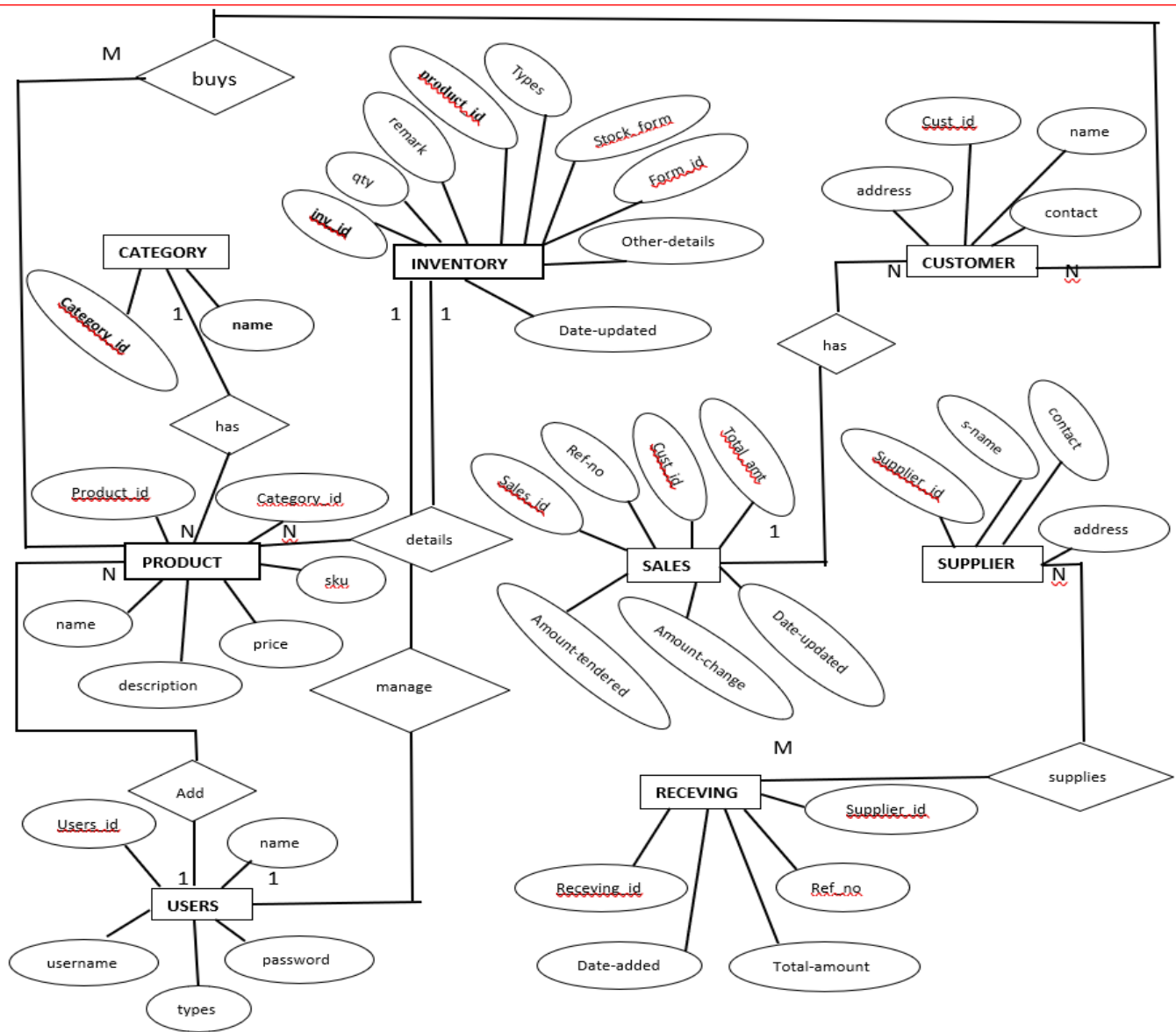


Figure 2.1.1 Entity Relatinship diagram for Groceries Sales_And_Inventory_System DBMS

2.2RELATION-SCHEMA DIAGRAM

2.2.1 STEP 1: MAPPING OF REGULAR ENTITY TYPES

In this step, we draw the schema for strong entities.

USERS

<u>User_id</u>	name	username	password	types
----------------	------	----------	----------	-------

CATEGORY

<u>Category_id</u>	name
--------------------	------

CUSTOMER

<u>Customer_id</u>	name	contact	address
--------------------	------	---------	---------

INVENTORY

<u>Inv_id</u>	Product_id	qty	types	Stock_from	Form_id	Other_details	remarks	Date_updated
---------------	------------	-----	-------	------------	---------	---------------	---------	--------------

PRODUCT

<u>Product_id</u>	Category_id	sku	price	name	description
-------------------	-------------	-----	-------	------	-------------

RECEIVING

<u>Rec_id</u>	Ref_no	Supplier_id	Total_amount	Date_added
---------------	--------	-------------	--------------	------------

SALES

<u>Sales_id</u>	Ref_no	Customer_id	Total_amount	Amount_tendered	Amount_change	Date_updated
-----------------	--------	-------------	--------------	-----------------	---------------	--------------

SUPPLIER

<u>Supplier_id</u>	Supplier_name	contact	address
--------------------	---------------	---------	---------

2.2.2 STEP 2: MAPPING OF WEAK ENTITY RELATIONSHIP TYPES.

We do not have weak entity in our ER diagram.

2.2.3 STEP 3: MAPPING OF BINARY 1:1 RELATIONSHIP TYPE INVENTORY

<u>Inv_id</u>	Product_id	qty	types	Stock_from	Form_id	Other_details	remarks	Date_updated
---------------	------------	-----	-------	------------	---------	---------------	---------	--------------

Figure 2.2.3: Mapping of 1:1 Binary Relationship Entity Types.

The primary key of USERS(USER ID) is included in the INVENTORY table.

2.2.4 STEP 4: MAPPING OF BINARY 1: N RELATIONSHIP TYPE

PRODUCT

<u>Product_id</u>	Category_id	sku	price	name	description
-------------------	-------------	-----	-------	------	-------------

Figure 2.2.4: Mapping of 1:N Binary Relationship Entity Types.

The primary key of CATEGORY(CATEGORY ID) is included in the PRODUCT table.

2.2.5 STEP 5: MAPPING OF BINARY M: N RELATIONSHIP TYPE.

CUSTOMER

<u>Customer_id</u>	name	contact	address
--------------------	------	---------	---------

Figure 2.2.5: Mapping of M:N Binary Relationship Entity Types.

The primary key of PRODUCT(PRODUCT ID) is included in CUSTOMER a new table.

2.2.6 STEP 6: MAPPING OF MULTIVALUED ATTRIBUTES.

We do not have such attributes in our ER diagram.

2.2.7 STEP7: MAPPING OF N-ARY RELATIONSHIP TYPE.

We do not have such attributes in our ER diagram.

2.3 SCHEMA DIAGRAM

A database schema is the skeleton structure that represents the logical view of the entire database. SALES AND INVENTORY SYSTEM is shown in figure 2.3.1

USERS

<u>User_id</u>	name	username	password	types
----------------	------	----------	----------	-------

CATEGORY

<u>Category_id</u>	name
--------------------	------

CUSTOMER

<u>Customer_id</u>	name	contact	address
--------------------	------	---------	---------

INVENTORY

<u>Inv_id</u>	Product_id	qty	types	Stock_from	Form_id	Other_details	remarks	Date_updated
---------------	------------	-----	-------	------------	---------	---------------	---------	--------------

PRODUCT

<u>Product_id</u>	Category_id	sku	price	name	description
-------------------	-------------	-----	-------	------	-------------

RECEIVING

<u>Rec_id</u>	Ref_no	Supplier_id	Total_amount	Date_added
---------------	--------	-------------	--------------	------------

SALES

<u>Sales_id</u>	Ref_no	Customer_id	Total_amount	Amount_tendered	Amount_change	Date_updated
-----------------	--------	-------------	--------------	-----------------	---------------	--------------

SUPPLIER

<u>Supplier_id</u>	Supplier_name	contact	address
--------------------	---------------	---------	---------

Figure 2.3.1: Relation Schema diagram for Sales and inventory system DBMS.

2.4 NORMALISATION

First Normal Form (1NF)

A relation is in 1NF if and only if the domain of each attribute contains only atomic values, and the value of each attribute contains only a single value from that domain.

Second Normal Form (2NF)

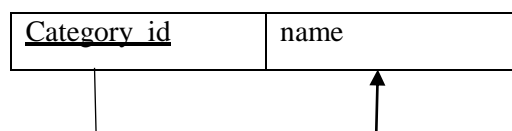
A relation is in 2NF if it is in 1NF and no non-prime attribute is dependent on any proper subset of any candidate key of the relation. A non-prime attribute of a relation is an attribute that is not a part of any candidate key of the relation.

Third Normal Form (3NF)

A relation is in 3NF if and only if both of the following conditions hold:

- The relation R is in 2NF and
- Every non-prime attribute of R is non-transitively dependent on every key of R.

2.4.1 CATEGORY



The above relation does not have any multivalued attributes .Hence it satisfies 1NF.

The above relation is in 1NF and also all the nonprime attributes are fully functionally dependent on the prime attribute (Category_id).

The above relation is in 2NF and also no nonprime attribute is transitively dependent on the primary key. Hence it satisfies 3NF.

Figure 2.4.1:Normalisation of CATEGORY table.

2.4.2 CUSTOMER

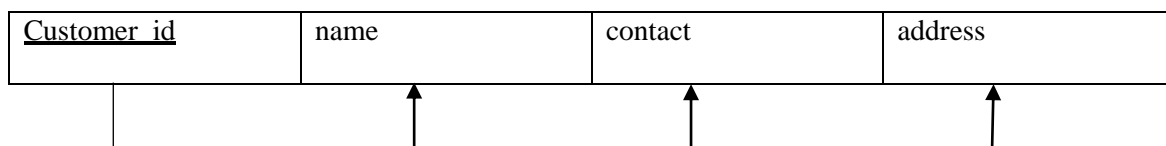


Figure 2.4.2:Normalisation of CUSTOMER table.

The above relation does not have any m+u+ivalued attributes .Hence it satisfies 1NF.

The above relation is in 1NF and also all the nonprime attributes are fully functionally dependent on the prime attribute (CUST_ID).

The above relation is in 2NF and also no nonprime attribute is transitively dependent on the primary key. Hence it satisfies 3NF.

2.4.3 INVENTORY

<u>Inv_id</u>	Product_id	qty	types	Stock_from	Form_id	Other_details	remarks	Date_updated
	↑	↑	↑	↑	↑	↑	↑	↑

Figure 2.4.3:Normalisation of INVENTORY table.

The above relation does not have any multivalued attributes .Hence it satisfies 1NF.

The above relation is in 1NF and also all the nonprime attributes are fully functionally dependent on the prime attribute (Inv_id).

The above relation is in 2NF and also no nonprime attribute is transitively dependent on the primary key. Hence it satisfies 3NF.

2.4.4 PRODUCT

<u>Product_id</u>	Category_id	sku	price	name	description
	↑	↑	↑	↑	↑

Figure 2.4.4:Normalisation of PRODUCT table.

The above relation does not have any multivalued attributes .Hence it satisfies 1NF.

The above relation is in 1NF and also all the nonprime attributes are fully functionally dependent on the prime attribute (Product_id).

The above relation is in 2NF and also no nonprime attribute is transitively dependent on the primary key. Hence it satisfies 3NF.

2.4.5 RECEIVING

<u>Rec_id</u>	Ref_no	Supplier_id	Total_amount	Date_added
	↑	↑	↑	↑

Figure 2.4.5:Normalisation of RECEIVING table.

The above relation does not have any multivalued attributes .Hence it satisfies 1NF.

The above relation is in 1NF and also all the nonprime attributes are fully functionally dependent on the prime attribute (Rec_id).

The above relation is in 2NF and also no nonprime attribute is transitively dependent on

the primary key. Hence it satisfies 3NF.

2.4.6 SALES

<u>Sales_id</u>	Ref_no	Customer_id	Total_amount	Amount_tendered	Amount_change	Date_updated
	↑	↑	↑	↑	↑	↑

Figure 2.4.6:Normalisation of SALES table.

The above relation does not have any multivalued attributes .Hence it satisfies 1NF.

The above relation is in 1NF and also all the nonprime attributes are fully functionally dependent on the prime attribute (Sales_id).

The above relation is in 2NF and also no nonprime attribute is transitively dependent on the primary key. Hence it satisfies 3NF.

2.4.7 SUPPLIER

<u>Supplier_id</u>	Supplier_name	contact	address
	↑	↑	↑

Figure 2.4.7:Normalisation of SUPPLIER table.

The above relation does not have any multivalued attributes .Hence it satisfies 1NF.

The above relation is in 1NF and also all the nonprime attributes are fully functionally dependent on the prime attribute (SUPPLIER_id).

The above relation is in 2NF and also no nonprime attribute is transitively dependent on the primary key. Hence it satisfies 3NF.

2.4.8 USERS

<u>User_id</u>	name	username	password	types
	↑	↑	↑	↑

Figure 2.4.8:Normalisation of USERS table.

The above relation does not have any multivalued attributes .Hence it satisfies 1NF.

The above relation is in 1NF and also all the nonprime attributes are fully functionally dependent on the prime attribute (USER_id).

The above relation is in 2NF and also no nonprime attribute is transitively dependent on the primary key. Hence it satisfies 3NF.

CHAPTER 3

IMPLEMENTATION

3.1 SOFTWARE AND HARDWARE SPECIFICATION

3.1.1 HARDWARE REQUIREMENTS

Mysql8.0 is required for a current system to run smoothly.

The minimum hardware requirements are:

CPU: Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU

Cores: Single (Dual/Quad Core is recommended)

RAM: 2 GB (6 GB recommended)

Graphic Accelerators: ATI with support of OpenGL 1.5 or higher

Display Resolution: 1280×1024 is recommended, 1024×768 is minimum.

The minimum hardware requirements for Mysql8.0 to run smoothly are:

OS: Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional/Windows 10

Processor: Intel® Core(TM)2 Duo CPU T6670 @ 2.20GHz

Memory: 512 MB Express Edition

Disk space: 650 MB of free disk space

The minimum hardware requirements for XAAMP to run smoothly are:

OS: Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional/Windows 10

Processor: Intel® Core(TM)2 Duo CPU T6670 @ 2.20GHz

Memory: 128MB

3.1.2 SOFTWARE REQUIREMENTS

Operating System: Windows 7/8/10

Software's used: Mysql8.0 , XAAMP , Sublime Text3

Libraries: JQuery, Bootstrap and Fontawesome

3.2 CREATE TABLE COMMANDS

3.2.1 CATEGORY

```
CREATE TABLE CATEGORY(  
CATEGORY_ID INT PRIMARY KEY AUTO_INCREMENT,  
CATEGORY_NAME VARCHAR(225));
```

3.2.2 CUSTOMER

```
CREATE TABLE CUSTOMER(  
CUSTOMER_ID INT PRIMARY KEY AUTO_INCREMENT,  
CUSTOMER_NAME VARCHAR(225),  
CONTACT VARCHAR(10),  
ADDRESS TEXT );
```

3.2.3 PRODUCT

```
CREATE TABLE PRODUCT (  
PRODUCT_ID INT,  
SKU VARCHAR(110),  
PRICE DOUBLE,  
NAME VARCHAR(150),  
DESCRIPTION TEXT,  
CATEGORY_ID INT,  
FOREIGN KEY(CATEGORY_ID) REFERENCES CATEGORY(CATEGORY_ID));
```

3.2.4 INVENTORY

```
CREATE TABLE INVENTORY(  
INV_ID INT PRIMARY KEY AUTO_INCREMENT,  
QTY INT,  
TYPE TINYINT(1) NOT NULL COMMENT '1 = stockin , 2 = stockout',  
STOCK_FROM VARCHAR(100) NOT NULL COMMENT 'sales/receiving',  
FORM_ID INT,  
OTHER_DETAILS TEXT,  
REMARKS TEXT,  
DATE_UPDATED DATETIME NOT NULL DEFAULT current_timestamp( ) on UPDATE current_timestamp( ),  
PRODUCT_ID INT,  
FOREIGN KEY(PRODUCT_ID) REFERENCES PRODUCT(PRODUCT_ID));
```

3.2.5 SUPPLIER

```
CREATE TABLE SUPPLIER(  
SUPPLIER_ID INT PRIMARY KEY AUTO_INCREMENT,  
SUPPLIER_NAME TEXT,  
CONTACT VARCHAR(10),  
ADDRESS TEXT );
```

3.2.6 RECEIVING

```
CREATE TABLE RECEIVING (  
RECEIVING_ID INT,  
REF_NO VARCHAR(20),  
TOTAL_AMOUNT DOUBLE,  
DATE_ADDED DATETIME NOT NULL DEFAULT current_timestamp( ),  
SUPPLIER_ID INT,  
FOREIGN KEY (SUPPLIER_ID)REFERENCES SUPPLIER(SUPPLIER_ID));
```

3.2.7 SALES

```
CREATE TABLE SALES (  
    SALES_ID INT PRIMARY KEY AUTO_INCREMENT,  
    REF_NO VARCHAR(30),  
    TOTAL_AMOUNT DOUBLE,  
    AMOUNT_TENDERED DOUBLE,  
    AMOUNT_CHANGE DOUBLE,  
    DATE_UPDATED DATETIME NOT NULL DEFAULT current_timestamp( ) ON UPDATE current_timestamp( ),  
    CUSTOMER_ID INT,  
    FOREIGN KEY(CUSTOMER_ID) REFERENCES CUSTOMER(CUSTOMER_ID));
```

3.2.8 SYSTEM_SETTINGS

```
CREATE TABLE SYSTEM_SETTINGS (  
    SYS_ID INT PRIMARY KEY AUTO_INCREMENT,  
    NAME TEXT,  
    EMAIL VARCHAR(200),  
    CONTACT VARCHAR(20),  
    COVER_IMG TEXT,  
    ABOUT_CONTENT TEXT ));
```

3.2.9 USERS

```
CREATE TABLE USERS (  
    USER_ID INT PRIMARY KEY AUTO_INCREMENT,  
    NAME VARCHAR(200),  
    USERNAME VARCHAR(100),  
    PASSWORD VARCHAR(200),  
    TYPE TINYINT(1) NOT NULL DEFAULT 2 COMMENT '1 = admin ,2 = cashier' );
```

3.3 TABLE STRUCTURE

3.3.1 CATEGORY RELATION

COLUMN NAME	DATATYPE
Category_id	INT
Category_name	VARCHAR (225)

Figure 3.3.1: CATEGORY RELATION

3.3.2 CUSTOMER RELATION

COLUMN NAME	DATATYPE
Customer_id	INT
Customer_name	VARCHAR (225)
Contact	VARCHAR (10)
Address	TEXT

Figure 3.3.2: CUSTOMER RELATION

3.3.3 PRODUCT RELATION

COLUMN NAME	DATATYPE
Product_id	INT
Sku	VARCHAR (110)
Price	DOUBLE
Name	VARCHAR (150)
Description	TEXT
Category_id	INT

Figure 3.3.3: PRODUCT RELATION

3.3.4 INVENTORY RELATION

COLUMN NAME	DATATYPE
Inv_id	INT
Qty	INT
Type	TINYINT (1)
Stock_from	VARCHAR (100)
From_id	INT
Other_details	TEXT
Remarks	TEXT
Date_Updated	DATETIME
Product_id	INT

Figure 3.3.4: INVENTORY RELATION

3.3.5 SUPPLIER RELATION

COLUMN NAME	DATATYPE
Supplier_id	INT
Supplier_name	TEXT
Contact	VARCHAR (10)
Address	TEXT

Figure 3.3.5: SUPPLIER RELATION

3.3.6 RECEIVING RELATION

COLUMN NAME	DATATYPE
Receivinig_id	INT
Ref_no	VARCHAR (20)
Total_amount	DOUBLE
Date_added	DATETIME
Supplier_id	INT

Figure 3.3.6: RECEIVING RELATION

3.3.7 SALES RELATION

COLUMN NAME	DATATYPE
Sales_id	INT
Ref_no	VARCHAR (30)
Total_amount	DOUBLE
Amount_tendered	DOUBLE
Amount_change	DOUBLE
Date_updated	DATETIME
Customer_id	INT

Figure 3.3.7: SALES RELATION

3.3.8 SYSTEM_SETTINGS RELATION

COLUMN NAME	DATATYPE
Sys_id	INT
Name	TEXT
Email	VARCHAR (200)
Contact	VARCHAR (20)
Cover_img	TEXT
About_content	TEXT

Figure 3.3.8: SYSTEM_SETTINGS RELATION

3.3.9 USERS RELATION

COLUMN NAME	DATATYPE
User_id	INT
Name	VARCHAR (200)
Username	VARCHAR (100)
Password	VARCHAR (200)
Type	TINYINT(1)

Figure 3.3.9: USERS RELATION

3.4 FUNCTIONALITY

3.4.1 CONNECTION TO DATABASE

```
<?php

$conn= new mysqli('localhost','root','','sales_inventory_db')
or die("Could not connect to mysql".mysqli_error($con));
```

3.4.2 INSERTING VALUES TO USERS

```
function save_user(){
    extract($_POST);
    $data = " name = '$name' ";
    $data .= ", username = '$username' ";
    $data .= ", password = '$password' ";
    $data .= ", type = '$type' ";
    if(empty($id)){
        $save = $this->db->query("INSERT INTO users set ".$data);}
```

3.4.3 INSERTING, DELETING AND UPDATING THE VALUES OF PRODUCT

```
function save_product(){
    extract($_POST);
    $data = " name = '$name' ";
    $data .= ", sku = '$sku' ";
    $data .= ", category_id = '$category_id' ";
    $data .= ", description = '$description' ";
    $data .= ", price = '$price' ";

    if(empty($id)){
        $save = $this->db->query("INSERT INTO product_list set ".$data);
    }else{
        $save = $this->db->query("UPDATE product_list set ".$data." where id=".$id);
    }
    if($save)
        return 1;
}

function delete_product(){
    extract($_POST);
    $delete = $this->db->query("DELETE FROM product_list where id = ".$id);
    if($delete)
        return 1;
}
```

3.4.4 INSERTING, DELETING AND UPDATING THE VALUES OF CUSTOMER

```
function save_customer(){
    extract($_POST);
    $data = " name = '$name' ";
    $data .= ", contact = '$contact' ";
    $data .= ", address = '$address' ";
    if(empty($id)){
        $save = $this->db->query("INSERT INTO customer_list set ".$data);
    }else{
        $save = $this->db->query("UPDATE customer_list set ".$data." where id=".$id);
    }
    if($save)
        return 1;
}

function delete_customer(){
    extract($_POST);
    $delete = $this->db->query("DELETE FROM customer_list where id = ".$id);
    if($delete)
        return 1;
}
```

3.4.5 INSERTING, DELETING AND UPDATING THE VALUES OF CATEGORY

```
function save_category(){
    extract($_POST);
    $data = " name = '$name' ";
    if(empty($id)){
        $save = $this->db->query("INSERT INTO category_list set ".$data);
    }else{
        $save = $this->db->query("UPDATE category_list set ".$data." where id=".$id);
    }
    if($save)
        return 1;
}

function delete_category(){
    extract($_POST);
    $delete = $this->db->query("DELETE FROM category_list where id = ".$id);
    if($delete)
        return 1;
}
```

3.4.6 FUNCTIONALITY TO CHECK THE PRODUCT AVAILABILITY

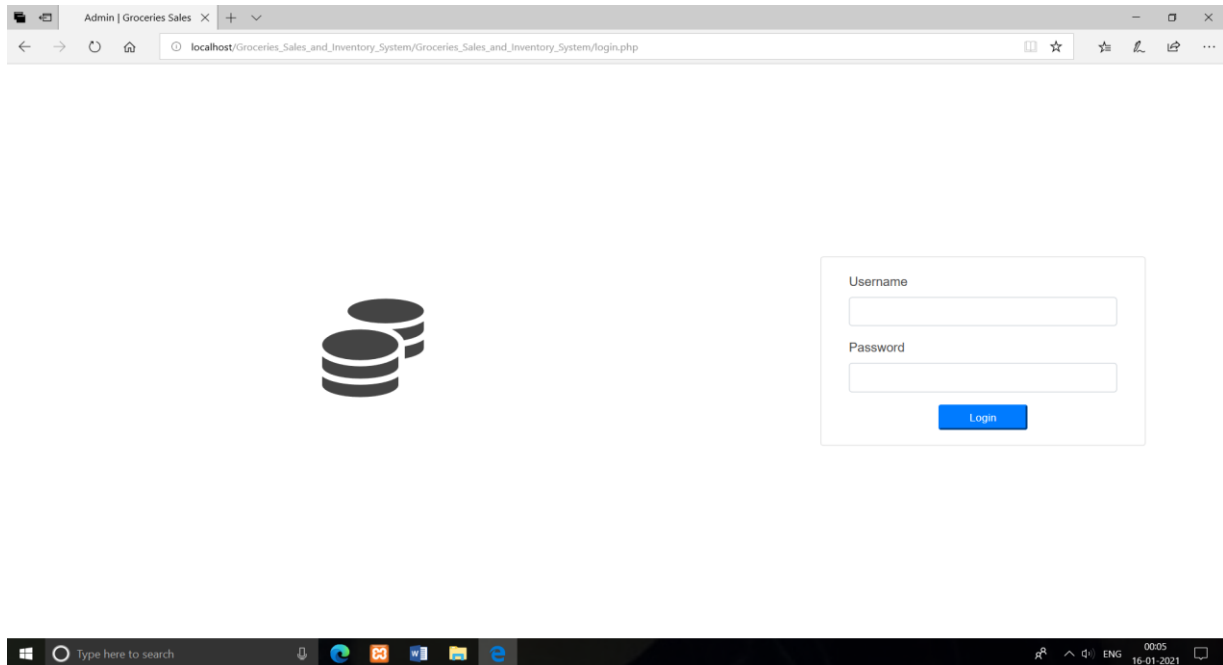
```
function chk_prod_availability(){
    extract($_POST);
    $price = $this->db->query("SELECT * FROM product_list where id = ".$id)->fetch_assoc()['price'];
    $inn = $this->db->query("SELECT sum(qty) as inn FROM inventory where type = 1 and product_id = ".$id);
    $inn = $inn && $inn->num_rows > 0 ? $inn->fetch_array()['inn'] : 0;
    $out = $this->db->query("SELECT sum(qty) as `out` FROM inventory where type = 2 and product_id = ".$id);
    $out = $out && $out->num_rows > 0 ? $out->fetch_array()['out'] : 0;
    $available = $inn - $out;
    return json_encode(array('available'=>$available,'price'=>$price));
}
```

CHAPTER 4

RESULT

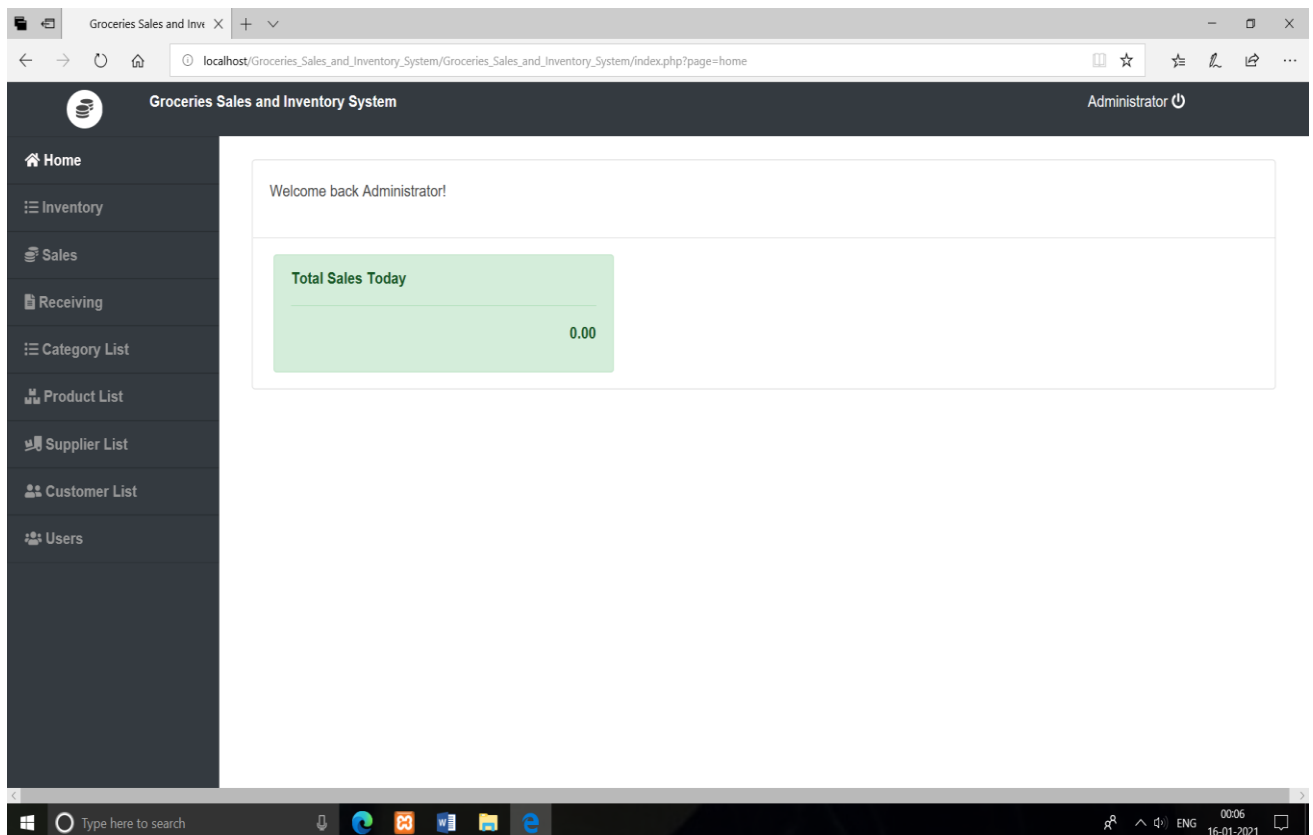
SNAPSHOTS:

4.1 LOGIN PAGE:



This is a login page where users can login (admin, cashier)

4.2 HOME PAGE:



This is the home page of groceries sales and inventory system

4.3 INVENTORY PAGE:

The screenshot shows the 'Inventory' page of the 'Groceries Sales and Inventory System'. The page has a dark sidebar on the left with navigation links: Home, Inventory (selected), Sales, Receiving, Category List, Product List, Supplier List, Customer List, and Users. The main content area is titled 'Inventory' and shows a table with 6 entries. The table has columns: #, Product Name, Stock In, Stock Out, and Stock Available. The data is as follows:

#	Product Name	Stock In	Stock Out	Stock Available
1	Alcohol Name	73	16	57
2	BULLET RICE	10000	2	9998
3	Chips (big)	50	44	6
4	cool drink	500		500
5	Lemon Iced Tea	100	10	90
6	Tuna	100	5	95

Below the table, it says 'Showing 1 to 6 of 6 entries'. There are 'Previous', '1' (current page), and 'Next' buttons. A search bar is located at the top right of the table area.

This is the inventory page where we can see the availability of stock.

4.4 ADD AND EDIT PAGE OF CUSTOMER:

The screenshot shows the 'Customer' page of the 'Groceries Sales and Inventory System'. The page has a dark sidebar on the left with navigation links: Home, Inventory, Sales, Receiving, Category List, Product List, Supplier List, Customer List (selected), and Users. The main content area is divided into two parts. On the left is the 'Customer Form' with fields for 'Customer Name', 'Contact', and 'Address', and 'Save' and 'Cancel' buttons. On the right is a table titled 'Customer' with columns: #, Customer, and Action. The data is as follows:

#	Customer	Action
1	Name : John Smith Contact : 8747808787 Address : Sample Only	Edit Delete
2	Name : George Wilson Contact : +14526-5455-44 Address : Sample	Edit Delete
3	Name : nagaraju Contact : 9880354610 Address : mandya	Edit Delete

This is customer form where customer can be added and edited by the admin.

4.5 ADD AND EDIT PAGE OF SUPPLIER:

The screenshot shows the 'Supplier' page of the 'Groceries Sales and Inventory System'. The page has a dark sidebar on the left with navigation links: Home, Inventory, Sales, Receiving, Category List, Product List, Supplier List (selected), Customer List, and Users. The main content area is divided into two parts. On the left is the 'Supplier Form' with fields for 'Supplier', 'Contact', and 'Address', and 'Save' and 'Cancel' buttons. On the right is a table titled 'Supplier Info' with columns: #, Supplier Info, and Action. The data is as follows:

#	Supplier Info	Action
1	Name : Supplier 1 Contact : 65524596 Address : Sample Address	Edit Delete
2	Name : Supplier 2 Contact : 6546531 Address : Supplier2 Address	Edit Delete
3	Name : Supplier 3 Contact : 9163455598 Address : Bangalore	Edit Delete
4	Name : vinutha Contact : 9163456790 Address : kerala	Edit Delete

4.6 ADD USERS PAGE:

New User

Name

Username

Password

User Type

Admin

Save

Cancel

4.7 MANAGE RECEIVING PAGE:

Groceries Sales and Inv

localhost/Groceries_Sales_and_Inventory_System/Groceries_Sales_and_Inventory_System/index.php?page=manage_receiving

Groceries Sales and Inventory System

Administrator

Home

Inventory

Sales

Receiving

Category List

Product List

Supplier List

Customer List

Users

Manage Receiving

Supplier

Please select here

Product

Please select here

Qty

Price

+ Add to List

Product	Qty	Price	Amount	
Total				

Save

4.8 ADD AND EDIT PAGE OF CATEGORY:

Groceries Sales and Inv

localhost/Groceries_Sales_and_Inventory_System/Groceries_Sales_and_Inventory_System/index.php?page=categories

Groceries Sales and Inventory System

Administrator

Home

Inventory

Sales

Receiving

Category List

Product List

Supplier List

Customer List

Users

Category Form

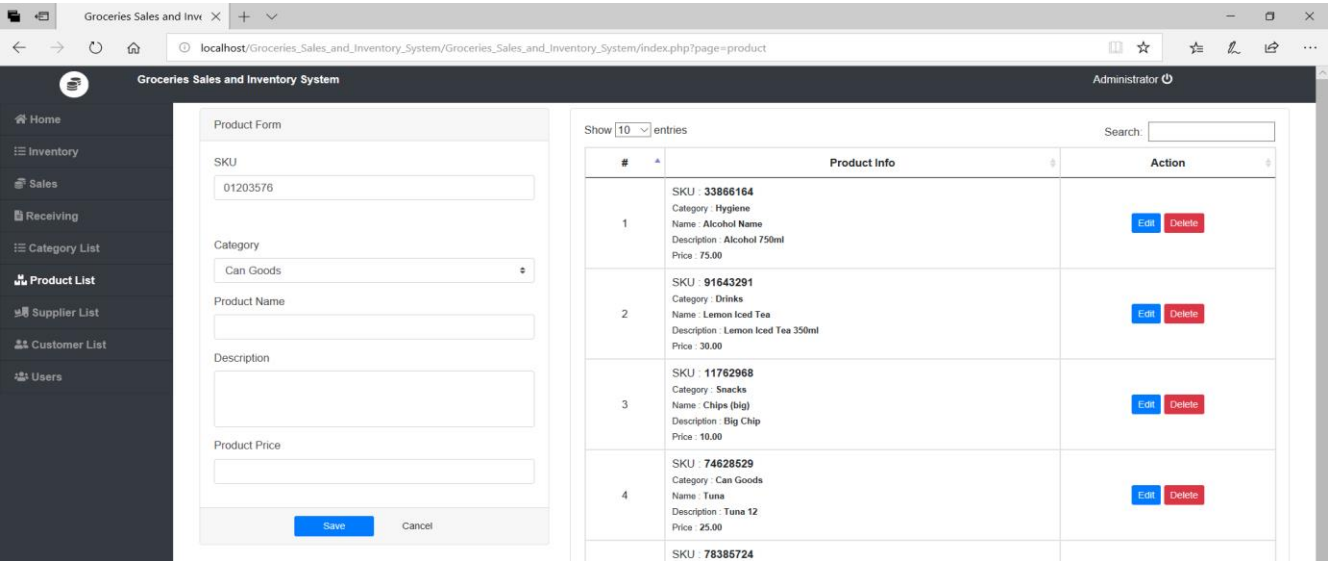
Category

Save

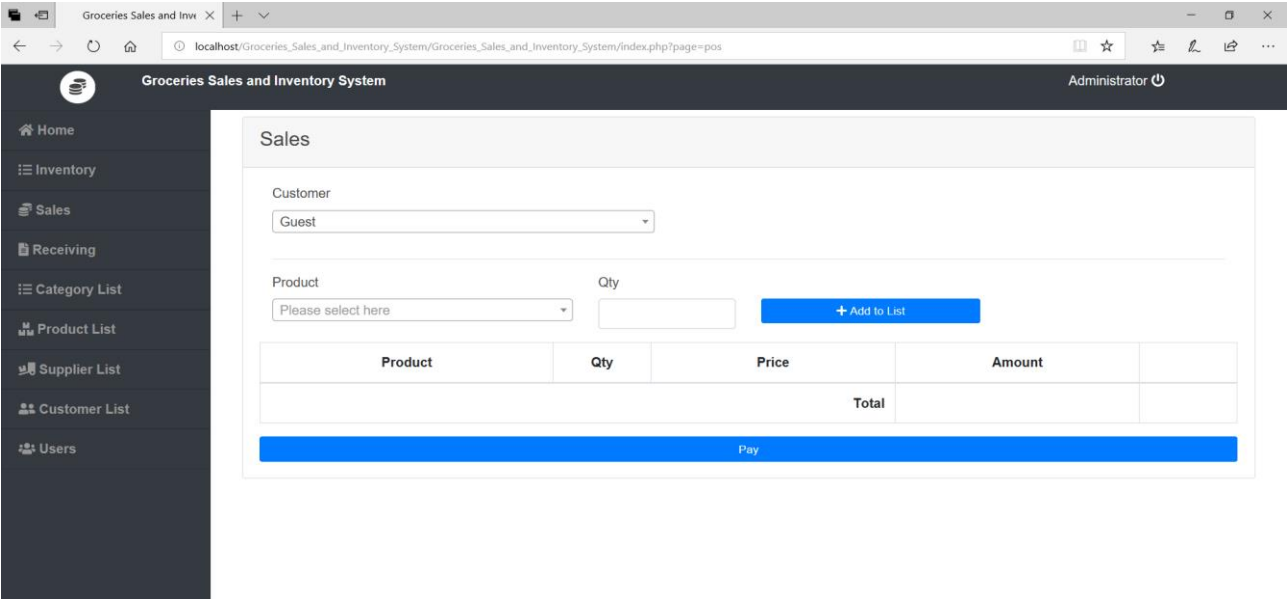
Cancel

#	Name	Action
1	Can Goods	<div>EditDelete</div>
2	Shampoo	<div>EditDelete</div>
3	Hygiene	<div>EditDelete</div>
4	Snacks	<div>EditDelete</div>
5	Drinks	<div>EditDelete</div>
6	RICE	<div>EditDelete</div>
7	powder	<div>EditDelete</div>

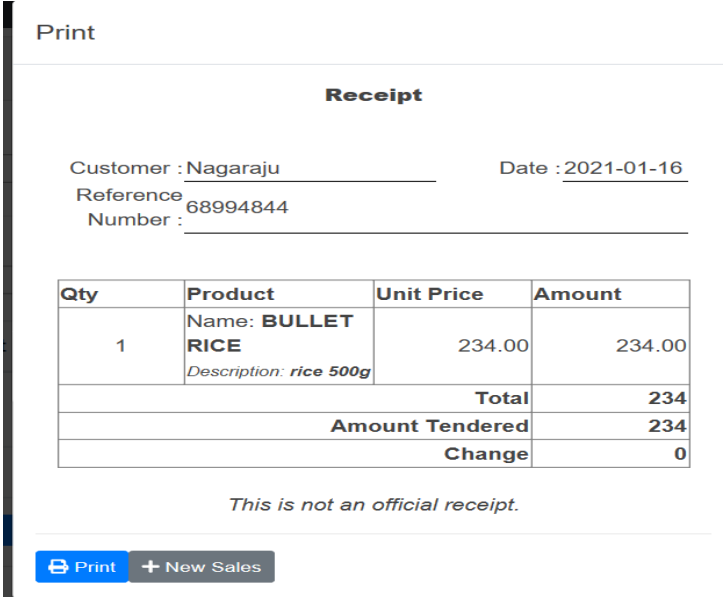
4.9 ADD AND EDIT PAGE OF PRODUCT:



4.10 PAYMENT PAGE:



4.11 PRINT OF RECEIPT PAGE:



CHAPTER 5:

CONCLUSION

5.1 Project limitation:

Since this is my first project it has some limitation. Due to less knowledge in particular fields and limited time I am not able to fulfil all my expectation that I expected I could do while the project got started. i hope this limitations are considerable. Some of the project limitation are:

- This application is not suitable for those organization where there is large quantity of product and different level of warehouses
- This software application is able to generate only simple reports.
- It is not suitable for large organization.

5.2 Conclusion:

To conclude, Groceries sales and Inventory Management System is a simple desktop based application basically suitable for small organization. It has every groceries items which are used for the small organization. I am successful in making the application where I can update, insert and delete the item as per the requirement. This application also provides a simple report on daily basis to know the daily sales and purchase details. This application matches for small organization where there small limited if go downs. Through it has some limitations, our team strongly believes that the implementation of this system will surely benefit the organization.