Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

MoE, UGC & AICTE Approved; NAAC Accredited A++

Karunya Nagar, Coimbatore - 641 114, Tamil Nadu, India.

DIVISION OF DATA SCIENCE AND CYBER SECURITY

SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

A SKILL BASED EVALUATION REPORT

SUBMITTED BY

AARYA B S(URK22AI1046)

COURSE CODE
20CS2016

COURSE NAME

DATABASE MANAGEMENT SYSTEMS

APRIL 2024

WHOLESALE MANAGEMENT SYSTEM

A REAL TIME APPLICATION REPORT

Submitted by

NIVEDITHA C S (URK22AI1028)
AARYA B S (URK22AI1046)

DIVISION OF DATA SCIENCE AND CYBER SECURITY

KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed-to-be-under Sec-3 of the UGC Act,
1956) Karunya Nagar, Coimbatore - 641 114. INDIA

APRIL 2024

CHAPTER -1

1.1 INTRODUCTION ABOUT DATABASE

A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques. A database management system refers to the technology for creating and managing databases. DBMS is a software tool to organize (create, retrieve, update and manage) data in a database. The main aim of a DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient. Computer scientists may classify database management systems according to the database models that they support. Relational databases became dominant in the 1980`s. These model data as rows and columns in a series of tables, and the vast majority use SQL for writing and querying data. In the 2000`s, non-relational databases became popular, referred to as no SQL because they use different query languages.

Stores any kind of data: A database management system should be able to store any kind of data. It should not be restricted to the employee name, salary, and address. Any kind of data that exists in the real world can be stored in DBMS because we need to work with all kinds of data that is present around us.

1.2 ABSTRACTION

A wholesale management system represents a sophisticated abstraction of business processes, designed specifically to streamline and optimize wholesale operations. At its core, this system encapsulates complex workflows into manageable modules, enabling wholesalers to oversee inventory, sales, purchasing, and customer relationships efficiently. By leveraging advanced database structures, such as relational databases or NoSQL solutions, the wholesale management system abstracts away low-level data storage details, allowing users to interact with intuitive interfaces for tasks like inventory tracking, order processing, and reporting.

This project is based on the sales transaction and billing of items in a Metro. The first activity is based on adding the items to the system along with the rate which are present in the Metro and the name of the items which the Metro will agree to sell. This authority is given only to admin (administrator). Any modifications to be done in the item name and the rate can be done only by admin. He also has the right to delete any item. As the customer buys the products and comes to the billing counter, the user is supposed to enter the item name he purchased and the quantity of the item he had purchased. This is not a huge a task.

The system will display all the items whose name starts with the letter selected by the user. He can select out of those displayed. Finally a separate bill will be generated for each customer. This will be saved in the database. Any periodic records can be viewed at any time. If the stock is not available, the Metro orders and buys from a prescribed vendor. The amount will be paid by deducting the total amount acquired in the sales activity. Admin provides a unique username and password for each through which he can login.

1.3 PROBLEM STATEMENT

The development of a Wholesale Management System (WMS) is essential to address the operational challenges faced by wholesalers in managing inventory, sales, and customer relationships. Key challenges include inaccurate inventory tracking leading to stockouts or overstocking, manual order processing resulting in delays and errors, and a lack of centralized customer relationship management (CRM) affecting customer satisfaction and retention. The primary objective of this project is to create a comprehensive solution that streamlines inventory management with real-time visibility, automates order processing from placement to fulfillment, enhances CRM capabilities to track customer interactions and preferences, and provides robust reporting and analytics for data-driven decision-making. By implementing a user-friendly interface, integrating with advanced technologies like barcode scanning and RFID, and ensuring scalability and security, the Wholesale Management System aims to optimize operations, improve efficiency, and drive growth in the wholesale sector. This project will deliver tangible benefits such as improved inventory accuracy, faster order processing enhanced customer satisfaction, and strategic insights for business expansion and competitiveness.

1.4 OBJECTIVE

The objective of this project is to design, develop, and implement a comprehensive Wholesale Management System (WMS) that addresses the specific operational needs and challenges faced by wholesalers. The primary goal is to streamline and optimize key business processes including inventory management, order processing, and customer relationship management (CRM) through automation and centralized data management.

By creating a user-friendly interface with intuitive functionalities, the WMS aims to provide real-time visibility into inventory levels, movement, and trends, enabling efficient stock tracking and replenishment. Automating order processing from order capture to fulfillment will reduce manual errors, minimize lead times, and enhance overall operational efficiency.

Additionally, incorporating robust CRM capabilities will empower wholesalers to manage customer interactions, preferences, and sales histories effectively, leading to improved customer satisfaction and retention. Furthermore, the Wholesale Management System will deliver comprehensive reporting and analytics tools to derive actionable insights from sales data, inventory turnover rates, and customer behaviors, enabling data-driven decision-making for strategic business growth. Ultimately, the objective of this project is to equip wholesalers with a scalable, adaptable, and technology-driven solution that enhances productivity, optimizes resources, and fosters competitiveness in the wholesale industry.

Furthermore, the project seeks to enhance supply chain visibility and efficiency by enabling real-time tracking of inventory movements, order statuses, and supplier interactions. This visibility will not only optimize inventory levels and reduce stockouts but also strengthen relationships with suppliers through timely communication and data-sharing.

CHAPTER – 2

2.1 ENTITY RELATIONSHIP DIAGRAM

An entity–relationship model (ER model) describes inter-related things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

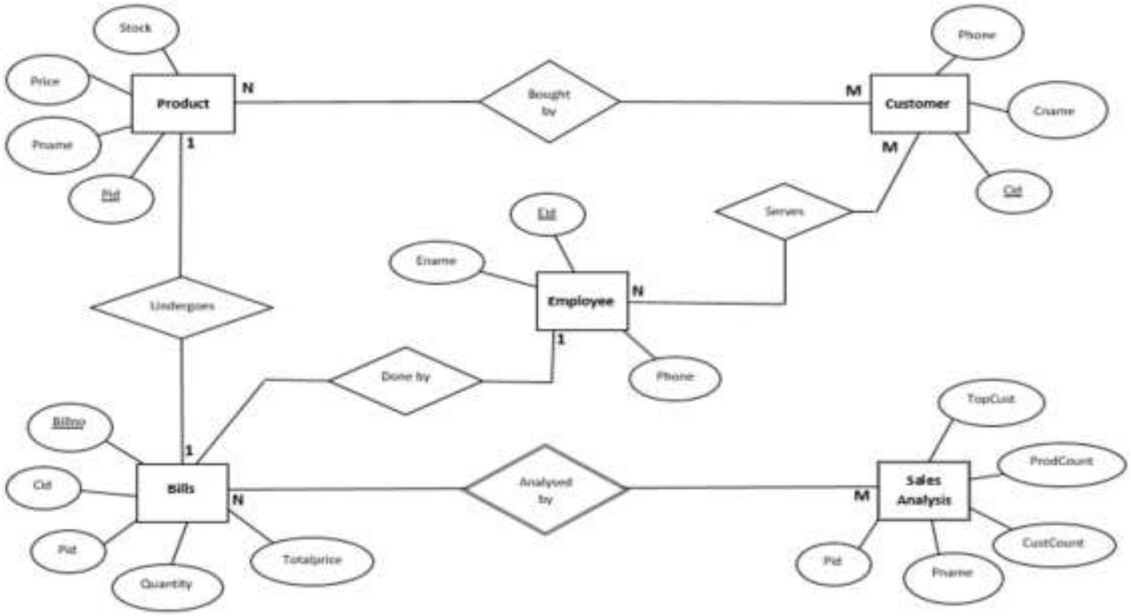


fig : 2.2 entity relational diagram

2.3 ENTITIES:

1) Products
2) Departments
3) Employees
4) Customers
5) Billing
6) Admin

2.4 Relations between the various Entities are:

1) ADMIN **adds** PRODUCTS. The relation between them is 1:M, one admin can add any

number of products.

2) ADMIN **adds** EMPLOYEES. The relation between them is 1:M, one admin can add any number of employee.
3) ADMIN **adds** CUSTOMERS. The relation between them is 1:M, one admin can add any number of customer.
4) CUSTOMERS **buys** PRODUCTS, Here the relation between them is N:M.
5) EMPLOYEE **servres** CUSTOMERS to shop. Here the relation between them is N:M.
6) PRODUCTS **undergoes** BILLING for the customers. Here the relation between them is 1:1, a bill can be paid by only one counter.

7)BILLS are **analysed by** SALES ANALYSIS. Here the relation is M:N

2.5 ER To Relational Mapping:

ER-to-Relational Mapping Algorithm:

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation Types

Step 4: Mapping of Binary 1:N Relationship Types.

Step 5: Mapping of Binary M:N Relationship Types.

Step 6: Mapping of Multivalued attributes.

Step 7: Mapping of N-ary Relationship Types.

2.6 RELATIONAL MODEL

A relational database schema is the tables, columns and relationships that make up a relational database. A relational database schema helps you to organize and understand the structure of a database. This is particularly useful when designing a new database, modifying an existing database to support more functionality, or building integration between databases. The Relational model contains all ent0ities and their relations with other entities. It also contains all relations having (m: n) cardinality.

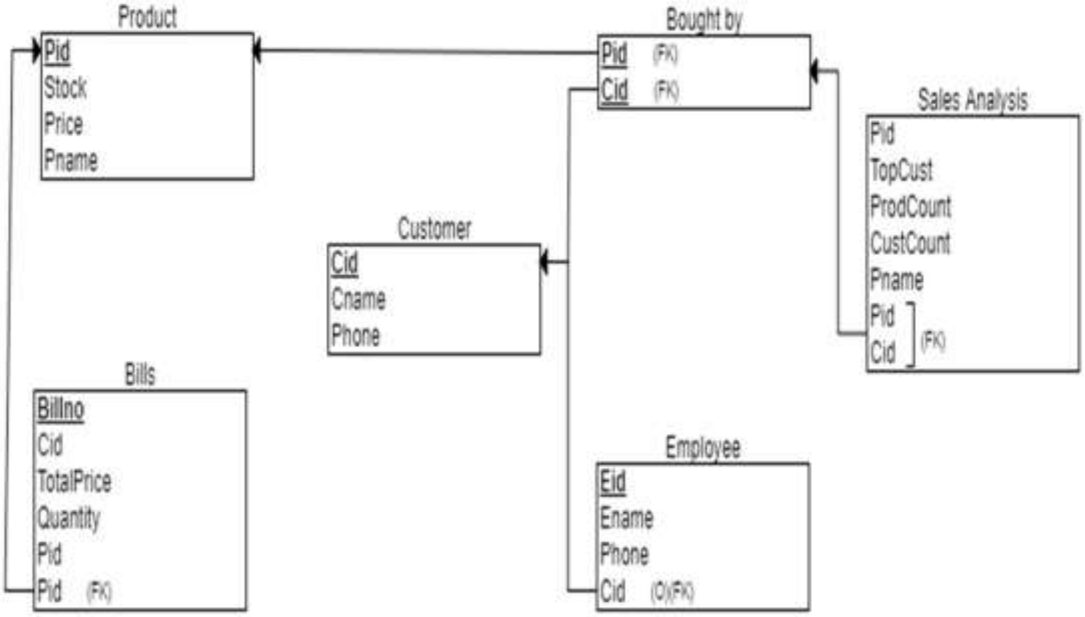


fig : 2.7 Relational Model of Wholesale management system

2.8 RELATINAL DATABASE SCHEMA

The database schema of a database system is its structure described in a formal language supported by the database management system (DBMS). The term "schema" refers to the organization of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a database schema is a set of formulas (sentences) called integrity constraints imposed on a database.[citation needed] These integrity constraints ensure compatibility between parts of the schema.

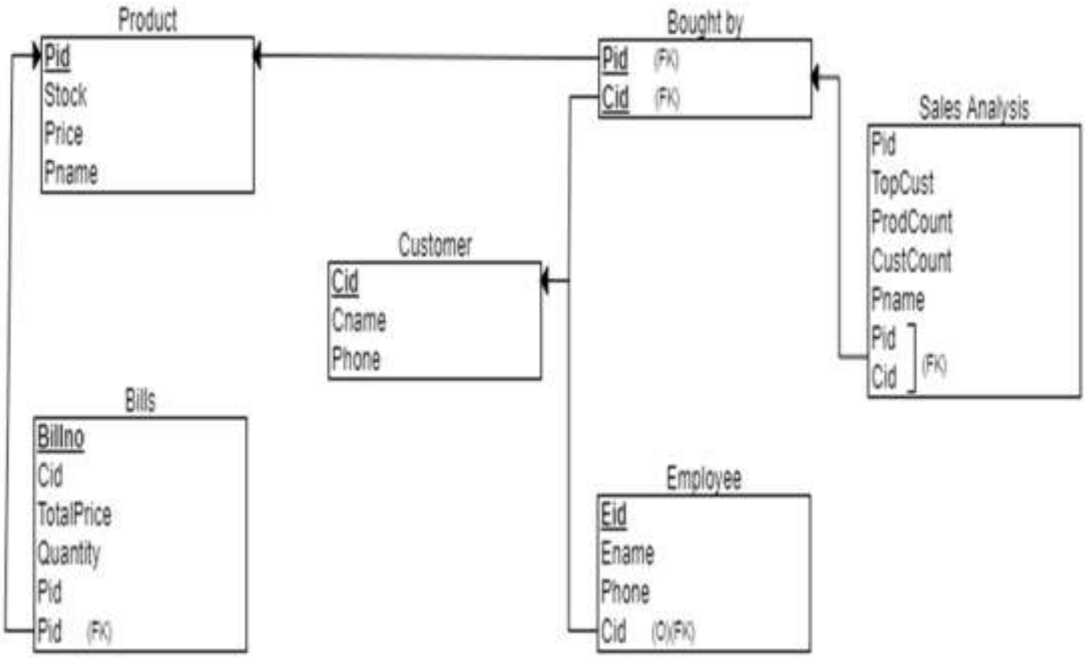


Fig : 2.9 Schema Diagram of Wholesale Management System

2.10 NORMALIZATION METHOD

TO PROVE THAT THEY ARE IN 1NF:

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.

First normal form is an essential property of a relation in a relational database. Database normalization is the process of representing a database in terms of relations in standard normal forms, where first normal is a minimal requirement.

First normal form enforces these criteria:

- Eliminate repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with a primary key

Products: pid, pcost, pdept, pname, pqty, poffer. Thus the relation is in 1NF.

Customers: cid, cname, cmob, cdept. Thus the relation is in 1NF.

Department: drptname, deptid. Thus the relation is in 1NF.

Employees: eid, ename, emob. Thus the relation is in 1NF.

Sales Analysis: pid. Pname, cost. Thus the relation is in 1NF.

Billing: cid, pid, soldqty, cost, total, billno, status.

KEY ATTRIBUTES:

1) Products: pid
2) Customers: cid
3) Employees: eid
4) Billing: bill no
5) Department: dname
6) Sales analysis: sid

TO PROVE THAT THEY ARE IN 2NF:

1. The customer table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non-prime attribute. Hence the table is in 2NF.
2. The employee table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non-prime attribute. Hence the table is in 2NF.
3. The product table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non-prime attribute. Hence the table is in 2NF.
4. The sales analysis table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non prime attribute. Hence the table is in 2NF.
5. The billing table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non prime attribute. Hence the table is in 2NF.
6. The department table is in 1NF and based on the functional dependencies defined above, there is no prime attribute that determines a non prime attribute. Hence the table is in 2NF.

Therefore all the tables are in 2NF.

TO PROVE THAT THEY ARE IN 3NF:

The third normal form (3NF) is a normal form used in database normalization. Codd's definition states that a table is in 3NF if and only if both of the following conditions hold:

- The relation R (table) is in second normal form (2NF)
- Every non-prime attribute of R is non-transitively dependent on every key of R.

1. The customer table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.
2. The product table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.
3. The employee table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.
4. The sales analysis table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.
5. The billing table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.
6. The department table is in both 1NF and 2NF. In this table there are no transitive dependencies i.e a non-prime attribute does not functionally determine another non-prime attribute. Hence according to the definition, the table is in 3NF.

Therefore all the tables are in 3NF.

CHAPTER – 3

3.1 APPENDIX 1

3.2 SQL COMMANDS FOR TABLE STRUCTURE

Database: product_sales

Table structure for table admin

```
CREATE TABLE admin (
  id varchar(100) NOT NULL,
  username varchar(200) NOT NULL,
  password varchar(200) NOT NULL,
  mno varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Dumping data for table admin

```
INSERT INTO admin (id, username, password, mno) VALUES
('11', 'admin', '1234', '1234567890');
```

Table structure for table billing

```
CREATE TABLE billing (
  cid varchar(500) DEFAULT NULL,
  pid varchar(500) DEFAULT NULL,
  sold_qty varchar(500) DEFAULT NULL,
  cost varchar(500) DEFAULT NULL,
  total varchar(500) DEFAULT NULL,
  billno varchar(500) NOT NULL,
  status varchar(500) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Dumping data for table billing

```
INSERT INTO billing (cid, pid, sold_qty, cost, total, billno, status) VALUES
('1211', '1290', '2', '100', '200', '1', 'closed'),
('1411', '1001', '3', '600', '1800', '2', 'closed'),
('1211', '1290', '10', '500', '5000', '3', 'closed'),
('1211', '1094', '5', '2500', '12500', '4', 'closed'),
('1211', '1094', '28', '14000', '392000', '5', 'closed'),
('1211', '1001', '2', '400', '800', '6', 'closed');
```

Triggers billing

```
DELIMITER $$
CREATE TRIGGER t1 BEFORE INSERT ON billing FOR EACH ROW BEGIN
SET new.total=new.sold_qty*new.cost;
END
$$
DELIMITER ;
```

Table structure for table customers

```
CREATE TABLE customers (
  cid varchar(500) NOT NULL,
  cname varchar(500) NOT NULL,
  mno varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Dumping data for table customers

```
INSERT INTO customers (cid, cname, mno) VALUES
('1', 'Ram', '1029384650'),
('1211', 'sandeep', '6362367037'),
('1411', 'Deeps', '1234567890');
```

Table structure for table dept

```
CREATE TABLE dept (
  dept varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Dumping data for table dept

```
INSERT INTO dept (dept) VALUES
('groceries');
```

Table structure for table employees

```
CREATE TABLE employees (
  eid varchar(500) NOT NULL,
  ename varchar(500) NOT NULL,
  emno varchar(200) NOT NULL,
  edept varchar(200) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Dumping data for table employees

```
INSERT INTO employees (eid, ename, emno, edept) VALUES

('1111', 'Sandy', '1234123412', 'billing'),

('1112', 'Rakshi', '0987654321', 'Women'),

('123456', 'sam', '1472583690', 'Health');
```

Table structure for table products

```
CREATE TABLE products (

  pid varchar(500) NOT NULL,

  pname varchar(500) NOT NULL,

  pcost varchar(500) NOT NULL,

  pqty varchar(500) NOT NULL,

  poffer varchar(200) NOT NULL,

  pdept varchar(500) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Dumping data for table products

```
INSERT INTO products (pid, pname, pcost, pqty, poffer, pdept) VALUES

('1', 'Wheat Flour', '100', '100', 'Buy 1 get 1 Free', 'groceries'),

('1001', 'Almonds', '200', '500', '20% off', 'groceries'),

('1094', 'Shirt', '500', '30', '20% off ', 'menfashion'),

('12', 'turmeric powder', '100', '50', '10% off', 'groceries'),

('1212', 'lipstick', '350', '5', '10% off', 'womenfashion'),

('1290', 'Chilli powder', '50', '10', '10% off', 'groceries');
```

Table structure for table sales

```
CREATE TABLE sales (

  cid varchar(200) NOT NULL,

  pcost varchar(200) NOT NULL,

  pqty varchar(200) NOT NULL,

  sid varchar(500) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Indexes for dumped tables

Indexes for table admin

```
ALTER TABLE admin
  ADD PRIMARY KEY (id);
```

Indexes for table billing

```
ALTER TABLE billing
  ADD PRIMARY KEY (billno);
```

Indexes for table customers

```
ALTER TABLE customers
  ADD PRIMARY KEY (cid);
```

Indexes for table dept

```
ALTER TABLE dept
  ADD PRIMARY KEY (dept);
```

Indexes for table employees

```
ALTER TABLE employees
  ADD PRIMARY KEY (eid);
```

Indexes for table products

```
ALTER TABLE products
  ADD PRIMARY KEY (pid);
```

Indexes for table sales

```
ALTER TABLE sales
  ADD PRIMARY KEY (sid);
COMMIT;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

3.3 BACK END IMPLEMENTATION IN MYSQL

CREATE TABLE QUERIES

```
1) CREATE TABLE `billing` (
  `cid` varchar(500),
  `pid` varchar(500),
  `sold_qty` varchar(500),
  `cost` varchar(500),
  `total` varchar(500),
  `billno` varchar(500),
  `status` varchar(500));
2) CREATE TABLE `customers` (
  `cid` varchar(500),
  `cname` varchar(500),
  `mno` varchar(200));
3) CREATE TABLE `employees` (
  `eid` varchar(500),
  `ename` varchar(500),
  `emno` varchar(200),
  `edept` varchar(200));
4) CREATE TABLE `dept` (
  `dept` varchar(200));
5) CREATE TABLE `products` (
  `pid` varchar(500),
  `pname` varchar(500),
  `pcost` varchar(500),
  `pqty` varchar(500) ,
  `poffer` varchar(200),
  `pdept` varchar(500));
6) CREATE TABLE `sales` (
  `cid` varchar(200),
  `pcost` varchar(200),
  `pqty` varchar(200));
```

INSERT TABLE QUERIES

1) insert into products (pid,pname,pcost,pqty) values(:a, :b, :c, :d) ;

```
                    $stmt->bindParam(':a',$pid);
                    $stmt->bindParam(':b',$pn);
                    $stmt->bindParam(':c',$pc);
                    $stmt->bindParam(':d',$pq);
```

2) Insert into billing(cid,pid,sold_qty,cost,total,billno,status) VALUES(:a, :b, :c, :d, :e, :f, :g);

```
                    $stmt->bindParam(':a',$cid);
                    $stmt->bindParam(':b',$pid);
                    $stmt->bindParam(':c',$pq);
                    $stmt->bindParam(':d',$pc);
                    $stmt->bindParam(':e',$main_total);
                    $stmt->bindParam(':f',$bill_no);
  $stmt->bindParam(':g',$sta);
```

3) insert into employees(eid,ename,emno) VALUES(:a, :b, :c);

```
                    $stmt->bindParam(':a',$eid);
                    $stmt->bindParam(':b',$en);
                    $stmt->bindParam(':c',$emno);
```

4) insert into customers(cid,cname,mno) VALUES(:a, :b, :c)'

```
                    $stmt->bindParam(':a',$cid);
                    $stmt->bindParam(':b',$cn);
                    $stmt->bindParam(':c',$mno);
```

5) insert into department(deptid,deptname) VALUES(:a, :b)'

```
                    $stmt->bindParam(':a',$deptid);
                    $stmt->bindParam(':b',$deptname);
```

3.4 FRONT END IMPLEMENTATION

On opening the website registered admin can login to his account with unique username and password so that he can see some of the information like products, employee, customers, what the current month’s bill management and manager details etc…

Same as admin, manager also having an account where he can view every details like products, employees, customers, billing, and so on.

The pseudo code for various parts of the html page is given below:

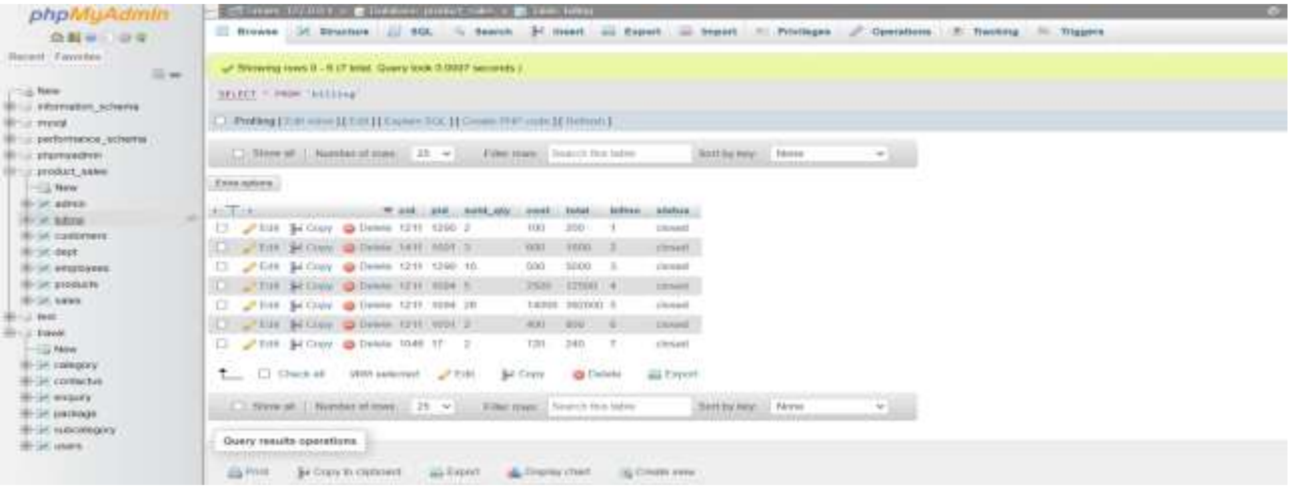
1) The **buttons** are implemented using the form tag with the input type as submit.

```
 <form action=”first_page.php”>
<input type=”submit” name=”submit” value=”admin” id=”btn”>
</form>
```

The ‘action’ specifies the page which should be opened when the submit button is clicked. ‘value’ specifies the text to be displayed on the button. ‘id’ specifies a unique value which can be used in the css. Similarly text, radio, checkbox etc. can be added in a form in the ‘type’ field.

3.5 APPENDIX 2

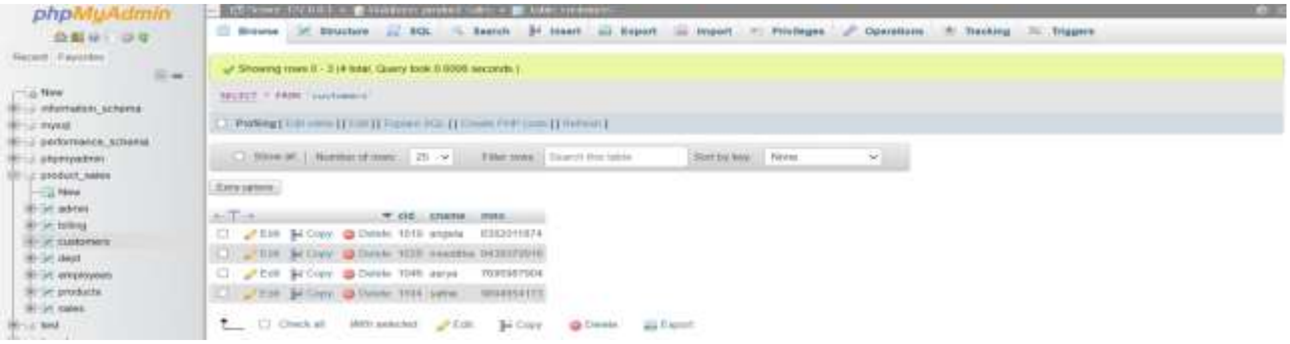
3.6 SQL OUTPUT



Showing rows 0 - 6 (7 total, Query took 0.0007 seconds.)

```
SELECT * FROM `billing`
```

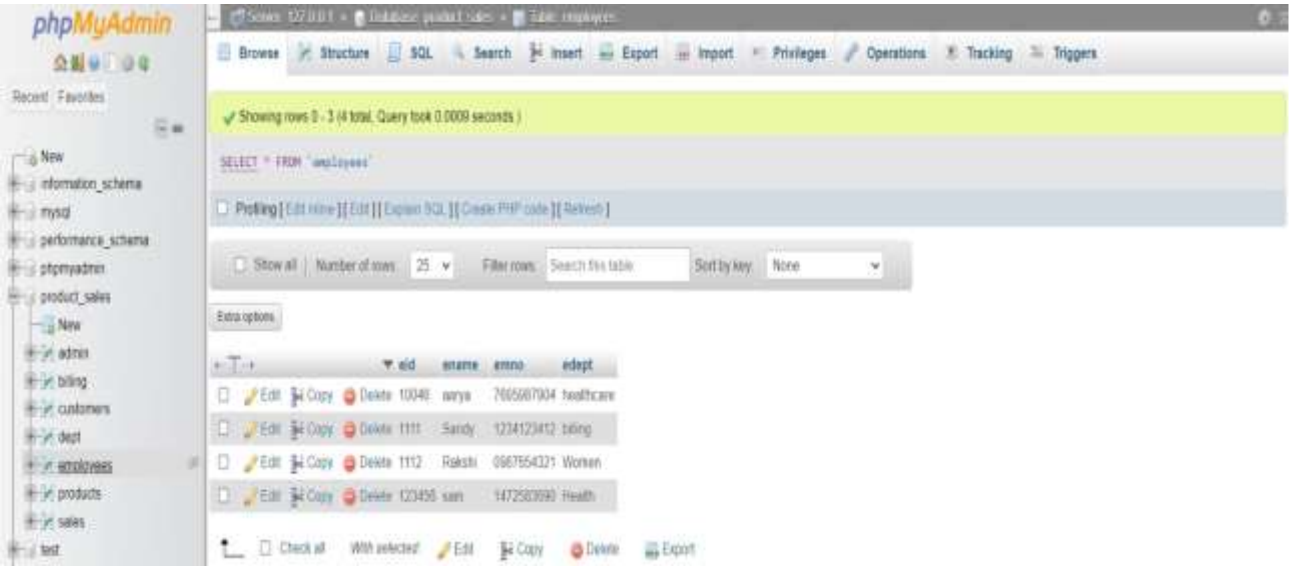
cid	pid	sold_qty	cost	total	billno	status
1211	1290	2	100	200	1	closed
1411	5501	3	600	1800	2	closed
1211	1290	10	500	5000	3	closed
1211	5504	5	2500	12500	4	closed
1211	5504	28	14000	392000	5	closed
1211	5501	2	400	800	6	closed
1048	17	2	120	240	7	closed



Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.)

```
SELECT * FROM `customers`
```

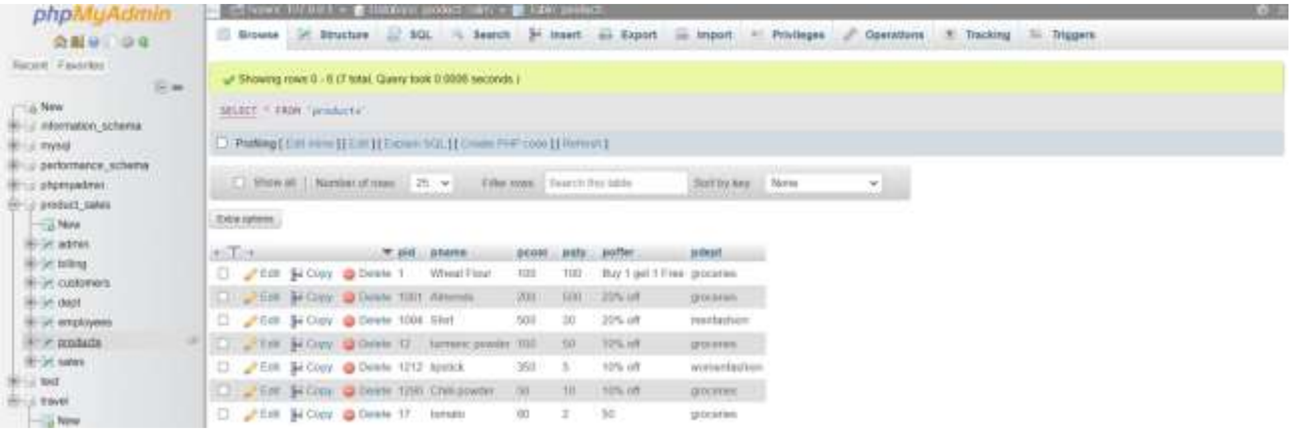
cid	cname	mno
1018	angela	8382011874
1020	[illegible]	[illegible]
1048	aarya	7095987904
1114	[illegible]	[illegible]



Showing rows 0 - 3 (4 total, Query took 0.0009 seconds.)

```
SELECT * FROM `employees`
```

eid	ename	emno	edept
10048	aarya	7695987904	healthcare
1111	Sandy	1234123412	billing
1112	Rakshi	0987654321	Women
123456	sam	1472583690	Health



✓ Showing rows 0 - 6 (7 total, Query took 0.0006 seconds.)

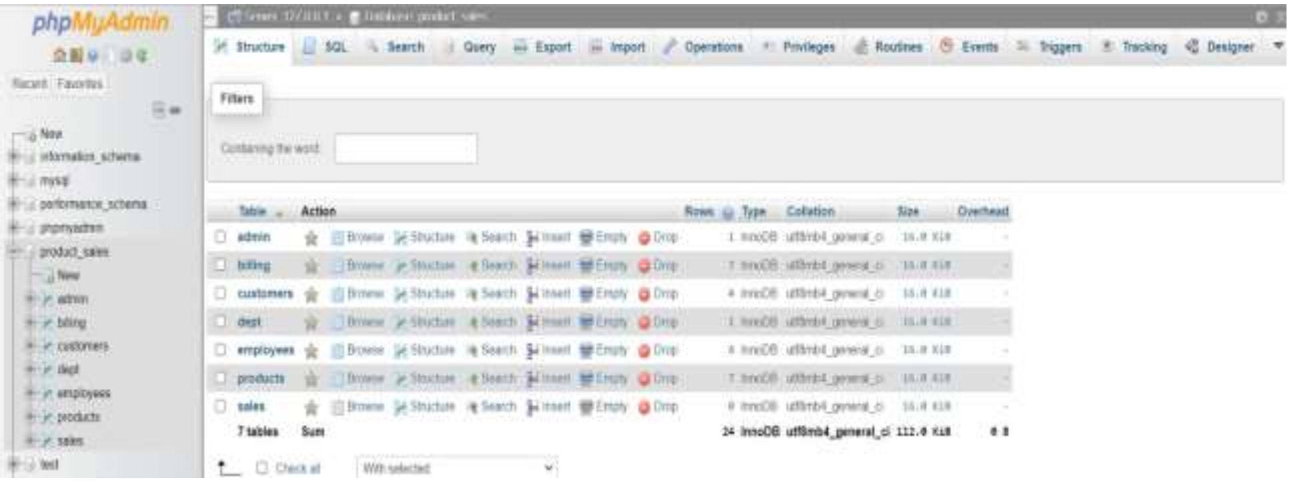
```
SELECT * FROM `products`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

				pid	pname	pcost	pqty	poffer	pdept
□	Edit	Copy	Delete	1	Wheat Flour	100	100	Buy 1 get 1 Free	groceries
□	Edit	Copy	Delete	1001	Almonds	200	100	20% off	groceries
□	Edit	Copy	Delete	1004	Shirt	500	30	20% off	menfashion
□	Edit	Copy	Delete	12	turmeric powder	100	50	10% off	groceries
□	Edit	Copy	Delete	1212	lipstick	350	5	10% off	womenfashion
□	Edit	Copy	Delete	1250	Chili powder	50	10	10% off	groceries
□	Edit	Copy	Delete	17	tomato	60	2	50	groceries

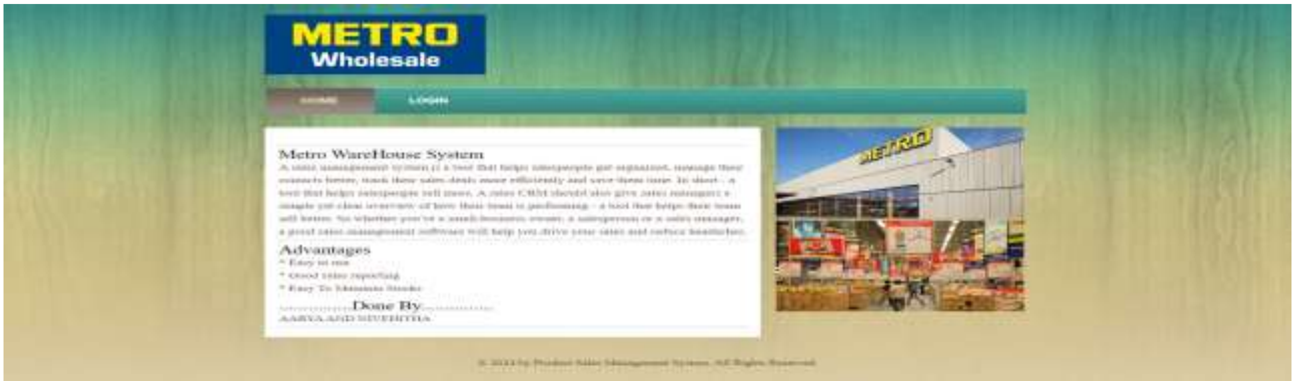


Containing the word:

	Table	Action							Rows	Type	Collation	Size	Overhead
□	admin	☆	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	billing	☆	Browse	Structure	Search	Insert	Empty	Drop	7	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	customers	☆	Browse	Structure	Search	Insert	Empty	Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	dept	☆	Browse	Structure	Search	Insert	Empty	Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	employees	☆	Browse	Structure	Search	Insert	Empty	Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	products	☆	Browse	Structure	Search	Insert	Empty	Drop	7	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	sales	☆	Browse	Structure	Search	Insert	Empty	Drop	0	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	7 tables	Sum							24	InnoDB	utf8mb4_general_ci	112.0 KiB	0 B

□ Check all | With selected:

3.7 OUTPUT



Metro WareHouse System

A sales management system is a tool that helps salespeople get organized, manage their contacts better, track their sales deals more efficiently and save them time. In short - a tool that helps salespeople sell more. A sales CRM should also give sales managers a simple yet clear overview of how their team is performing - a tool that helps their team sell better. So whether you're a small-business owner, a salesperson or a sales manager, a good sales management software will help you drive your sales and reduce headaches.

Advantages

* Easy to use

* Good sales reporting

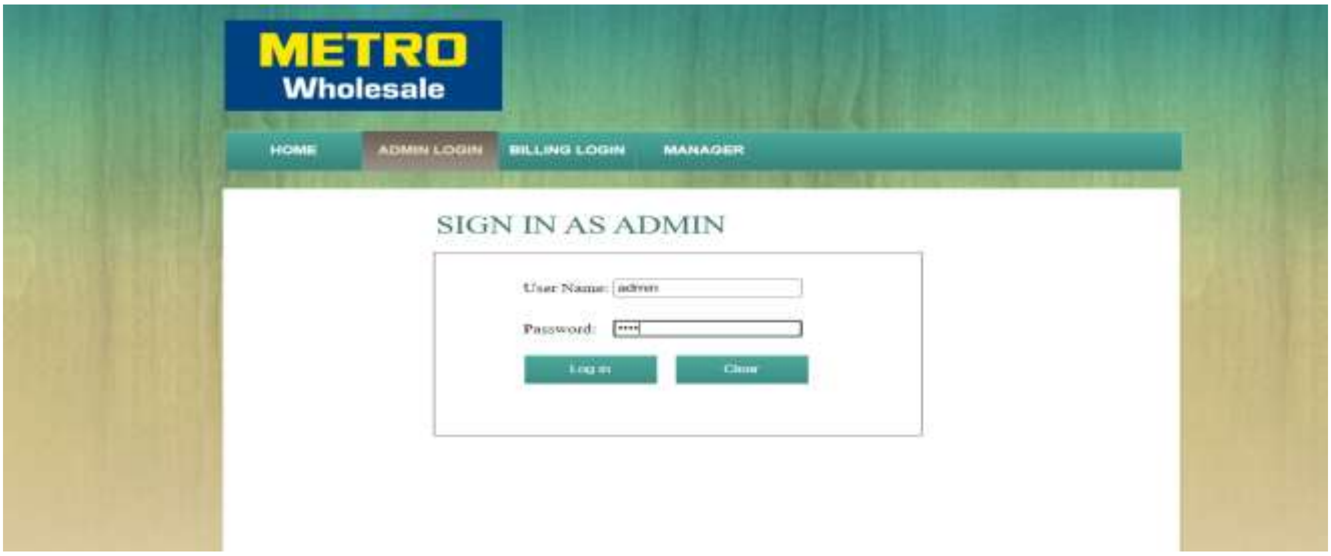
* Easy To Maintain Stocks

...............Done By...............

AARYA AND NIVEDHTHA

© 2023 by Product Sales Management System. All Rights Reserved

Fig : 3.8 home page

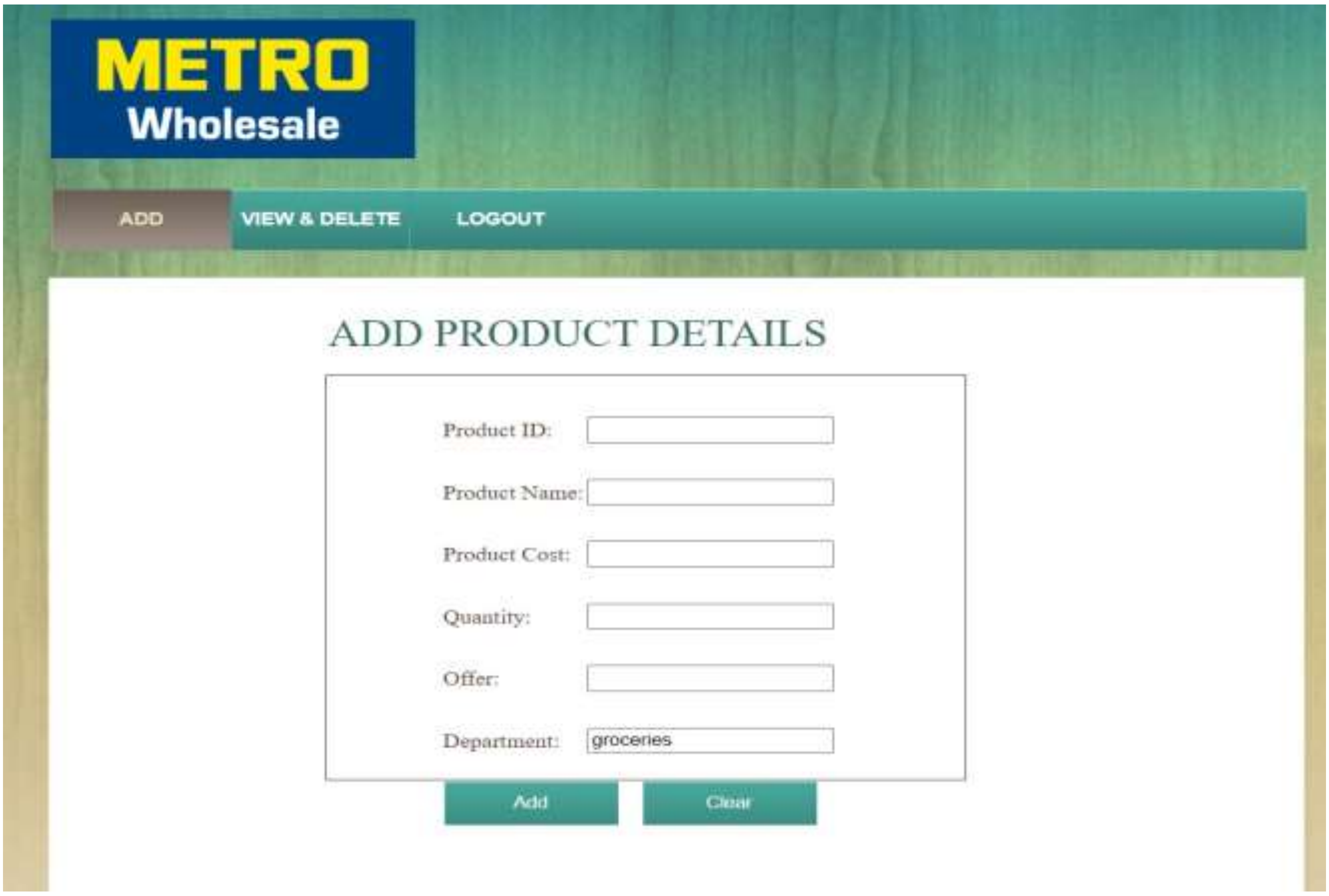


SIGN IN AS ADMIN

User Name: admin

Password: ****

Fig 3.9 Admin login



ADD PRODUCT DETAILS

Product ID:

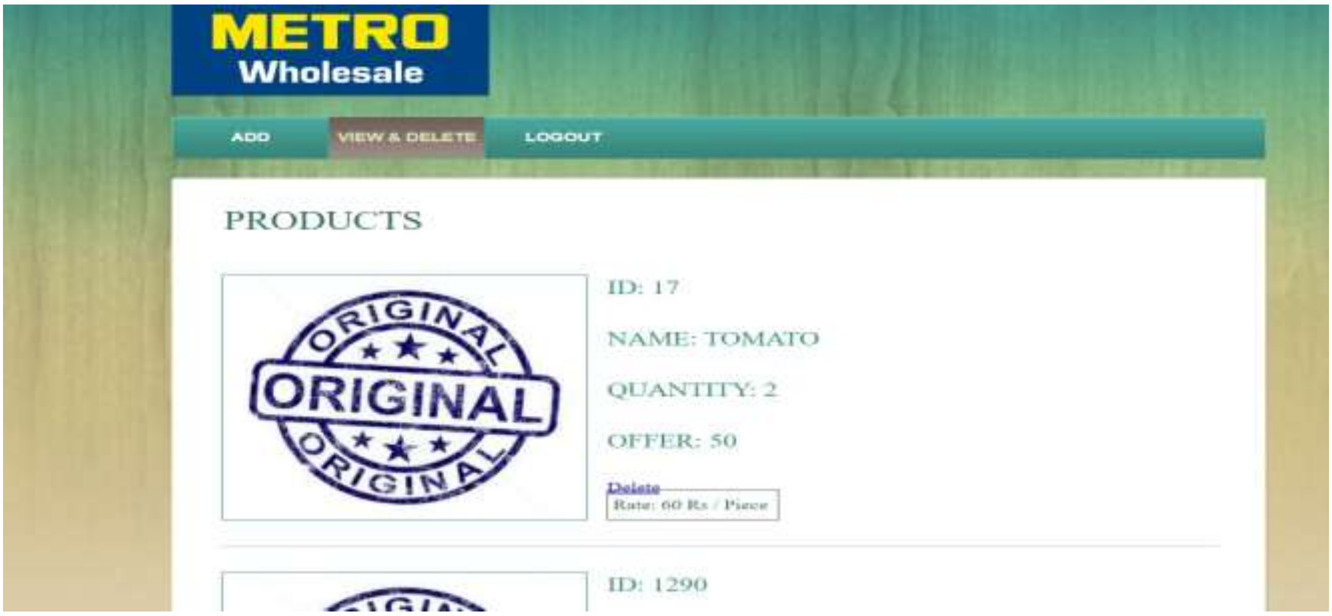
Product Name:

Product Cost:

Quantity:

Offer:

Department: groceries



PRODUCTS

ID: 17

NAME: TOMATO

QUANTITY: 2

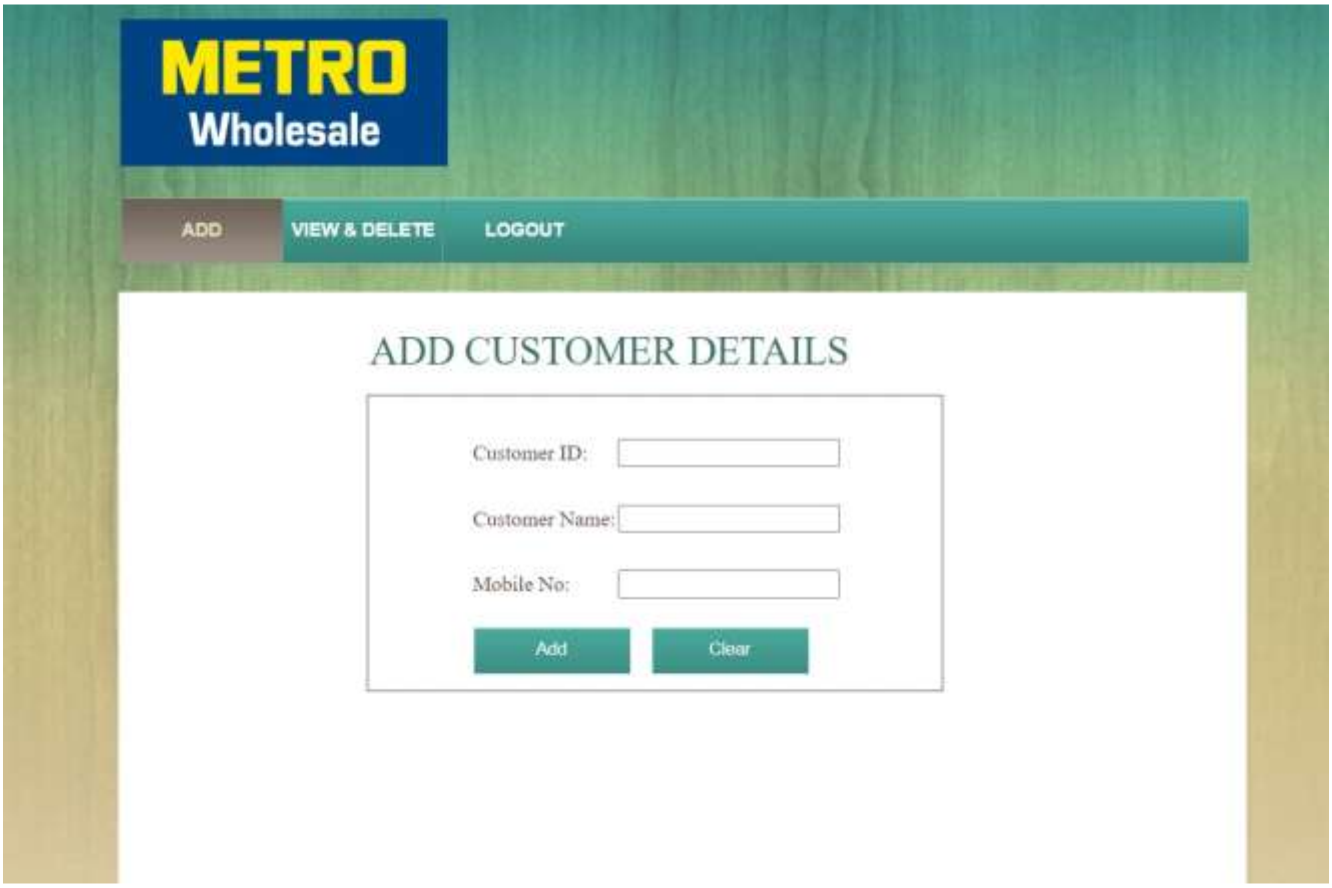
OFFER: 50

Delete

Rate: 60 Rs / Piece

ID: 1290

Fig 3.10 product details

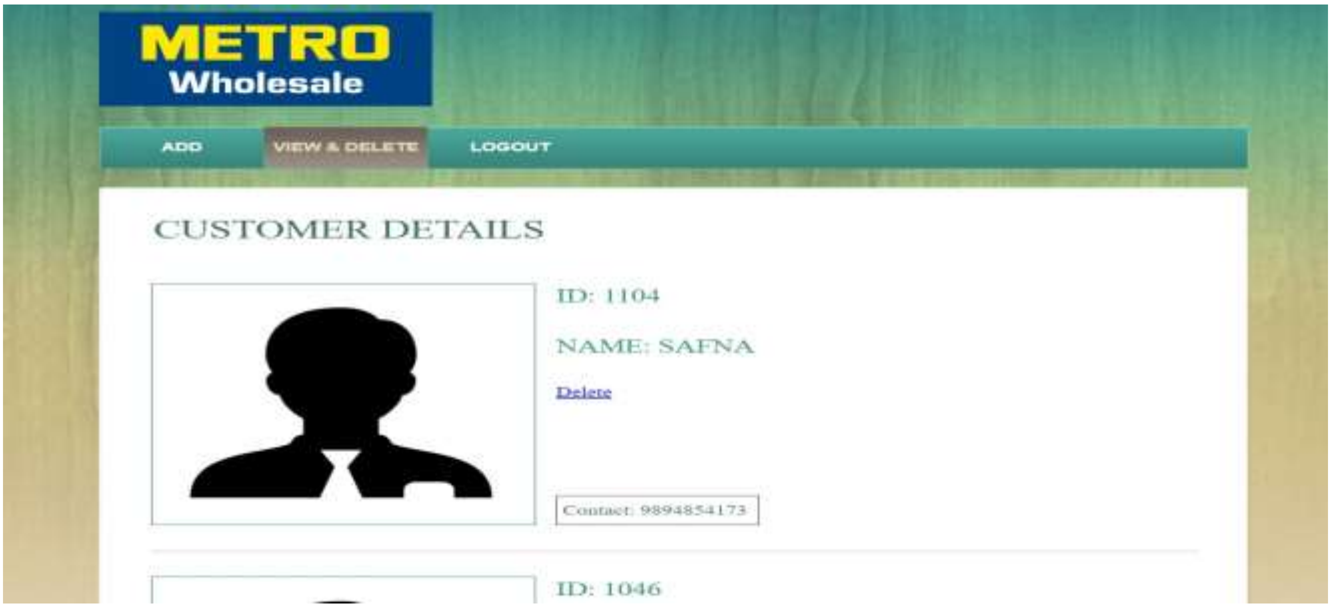


ADD CUSTOMER DETAILS

Customer ID:

Customer Name:

Mobile No:



CUSTOMER DETAILS

ID: 1104

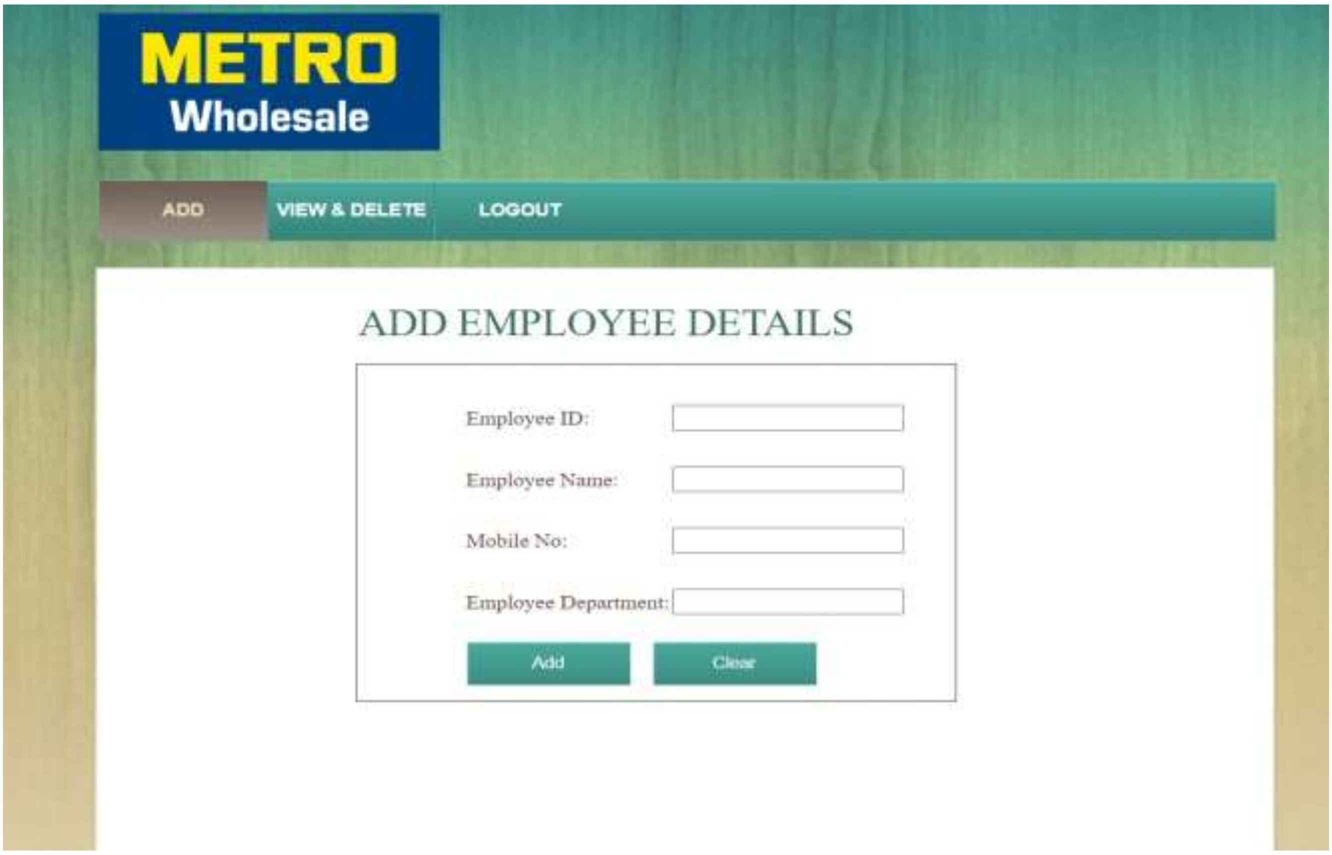
NAME: SAFNA

Delete

Contact: 9894854173

ID: 1046

Fig 3.11 customer details



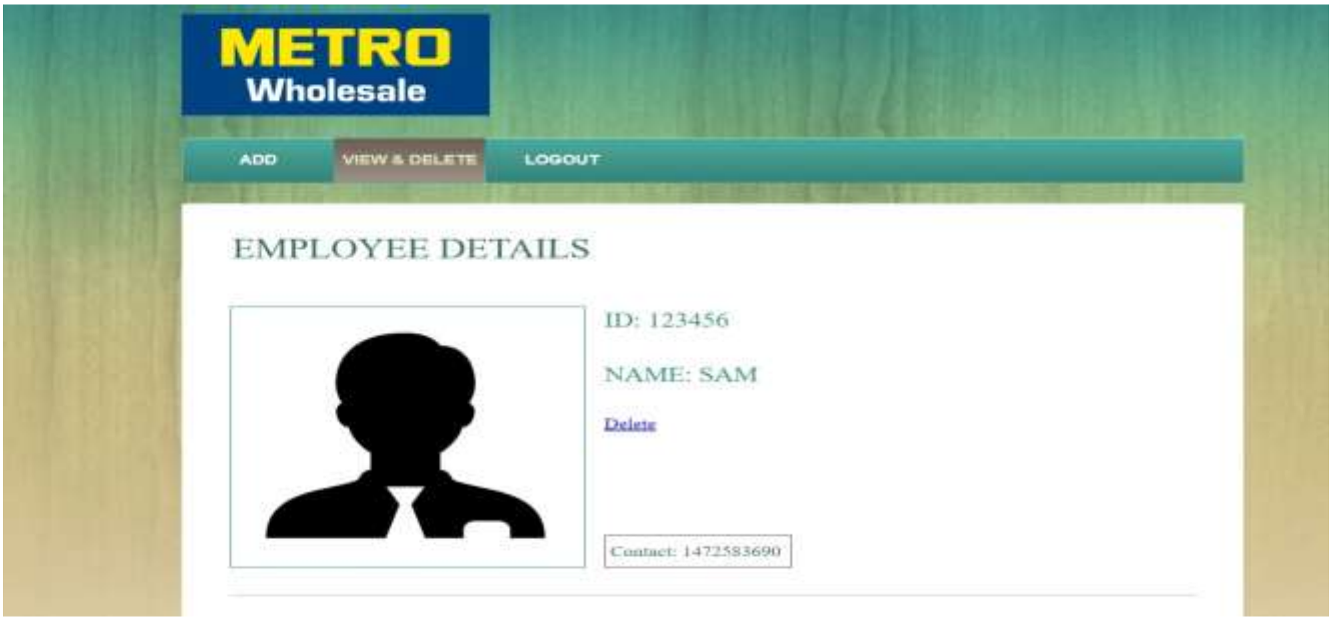
ADD EMPLOYEE DETAILS

Employee ID:

Employee Name:

Mobile No:

Employee Department:



EMPLOYEE DETAILS

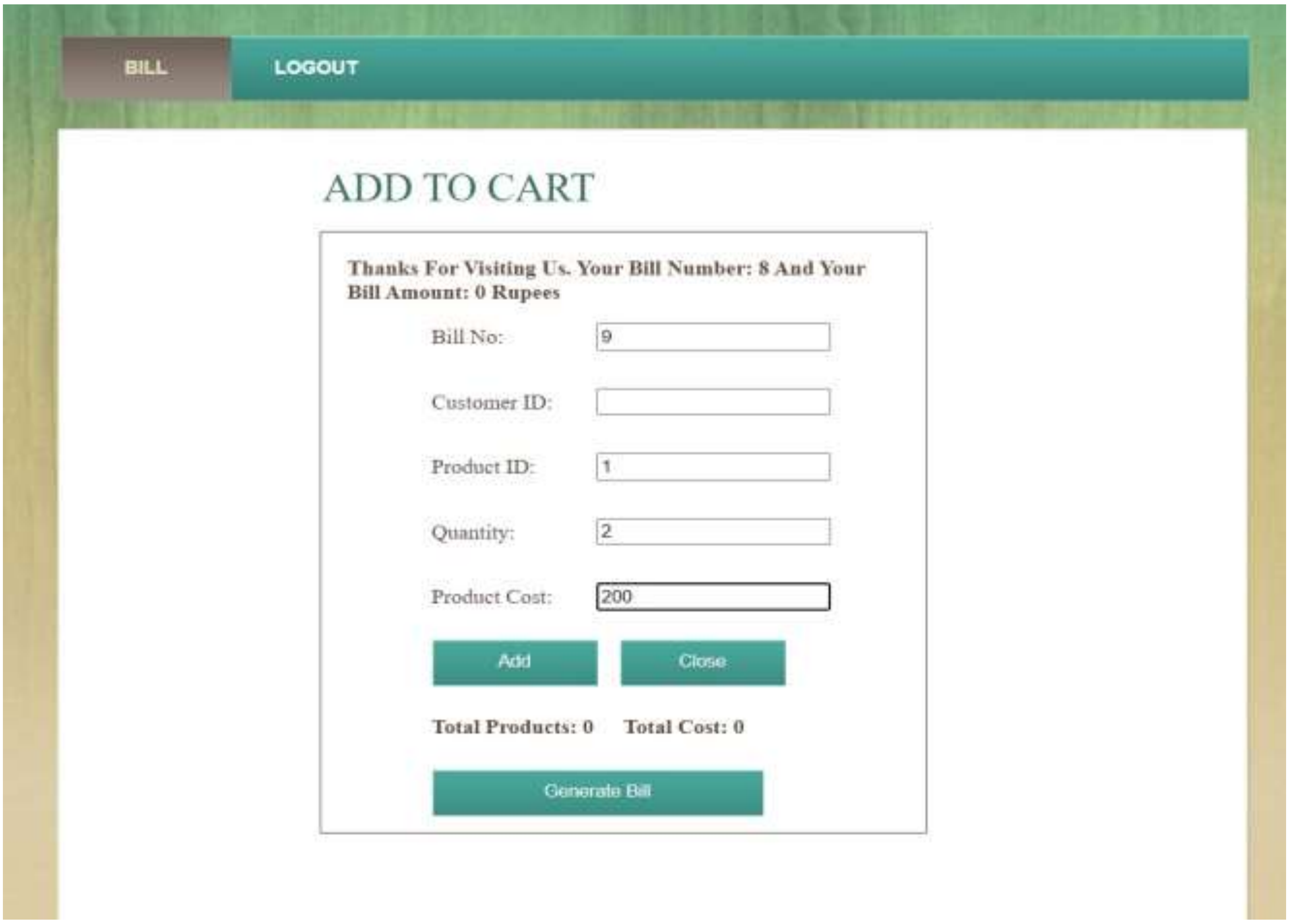
ID: 123456

NAME: SAM

Delete

Contact: 1472583690

Fig 3.12 employee details



ADD TO CART

Thanks For Visiting Us. Your Bill Number: 8 And Your Bill Amount: 0 Rupees

Bill No:	9
Customer ID:	
Product ID:	1
Quantity:	2
Product Cost:	200

Total Products: 0 Total Cost: 0

Fig 3.13 adding to cart

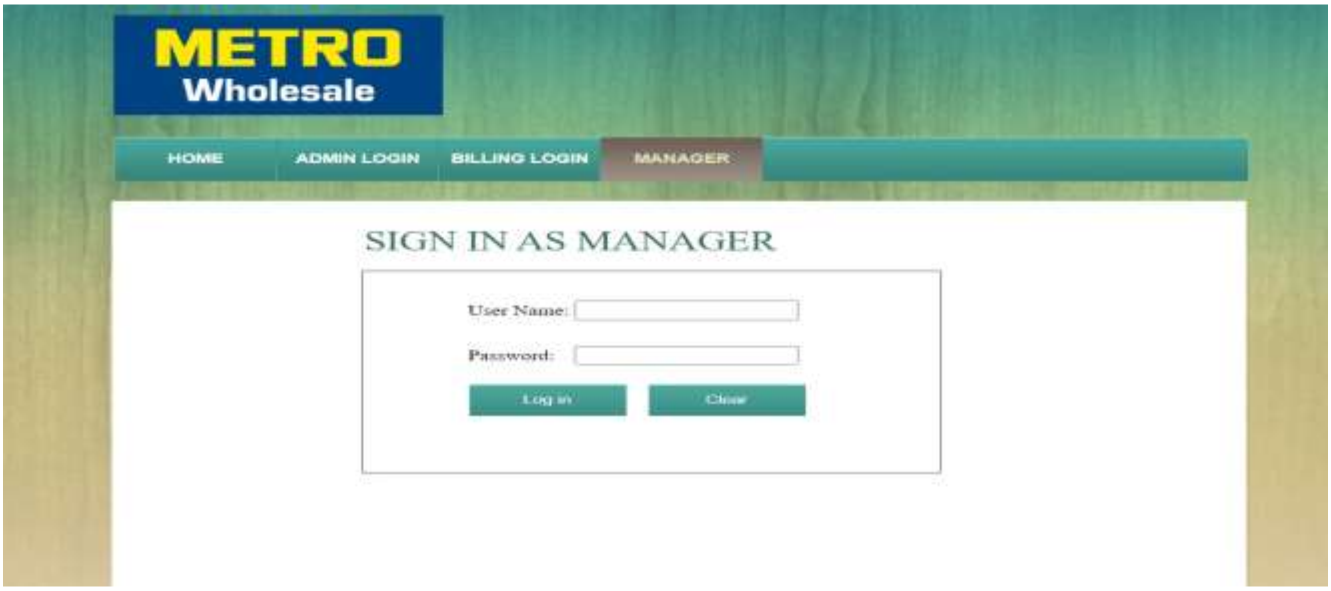
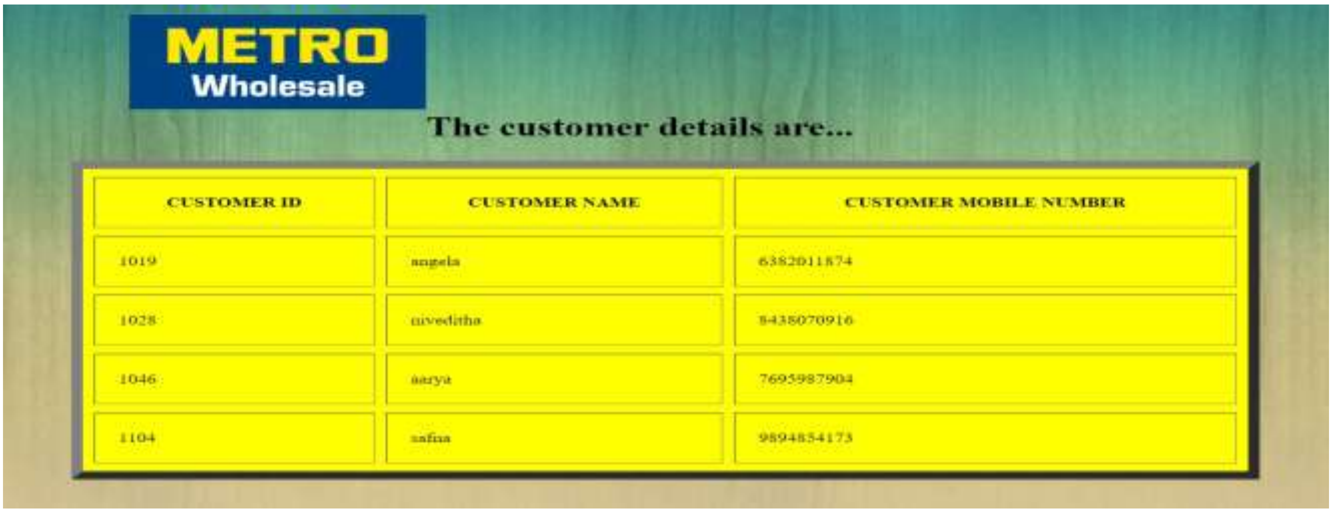


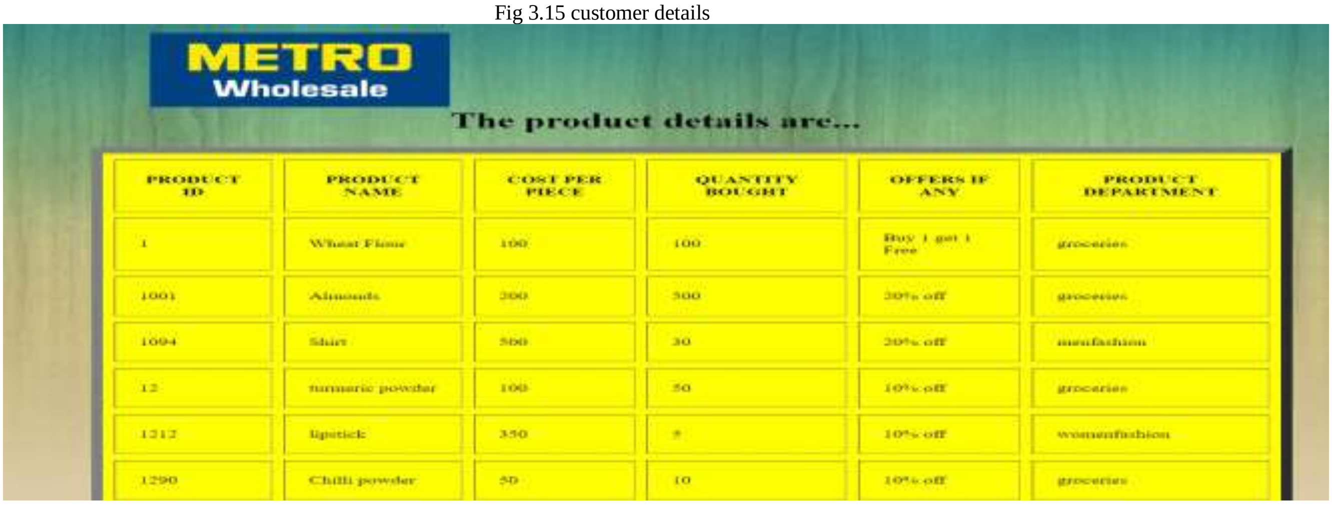
Fig 3.14 sign in as manager



The customer details are...

CUSTOMER ID	CUSTOMER NAME	CUSTOMER MOBILE NUMBER
1019	angela	6382011874
1028	niveditha	8438070916
1046	aarya	7695987904
1104	safna	9894854173

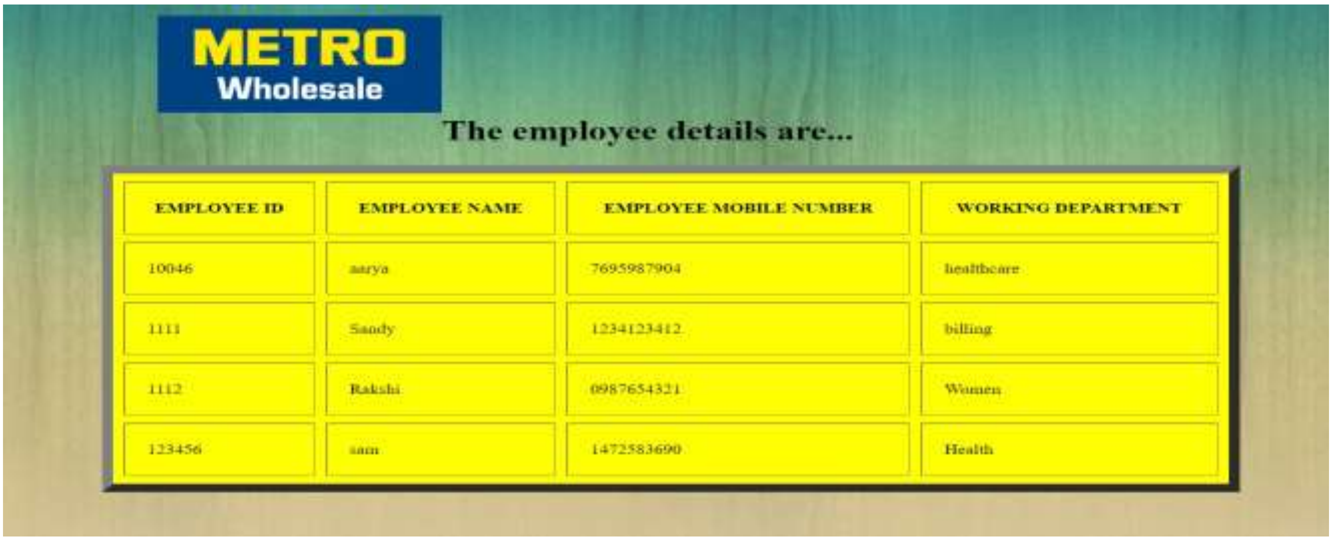
Fig 3.15 customer details



The product details are...

PRODUCT ID	PRODUCT NAME	COST PER PIECE	QUANTITY BOUGHT	OFFERS IF ANY	PRODUCT DEPARTMENT
1	Wheat Flour	100	100	Buy 1 get 1 Free	groceries
1001	Almonds	300	500	30% off	groceries
1094	Shirt	500	30	20% off	menfashion
12	turmeric powder	100	50	10% off	groceries
1212	lipstick	350	8	10% off	womenfashion
1290	Chilli powder	50	10	10% off	groceries

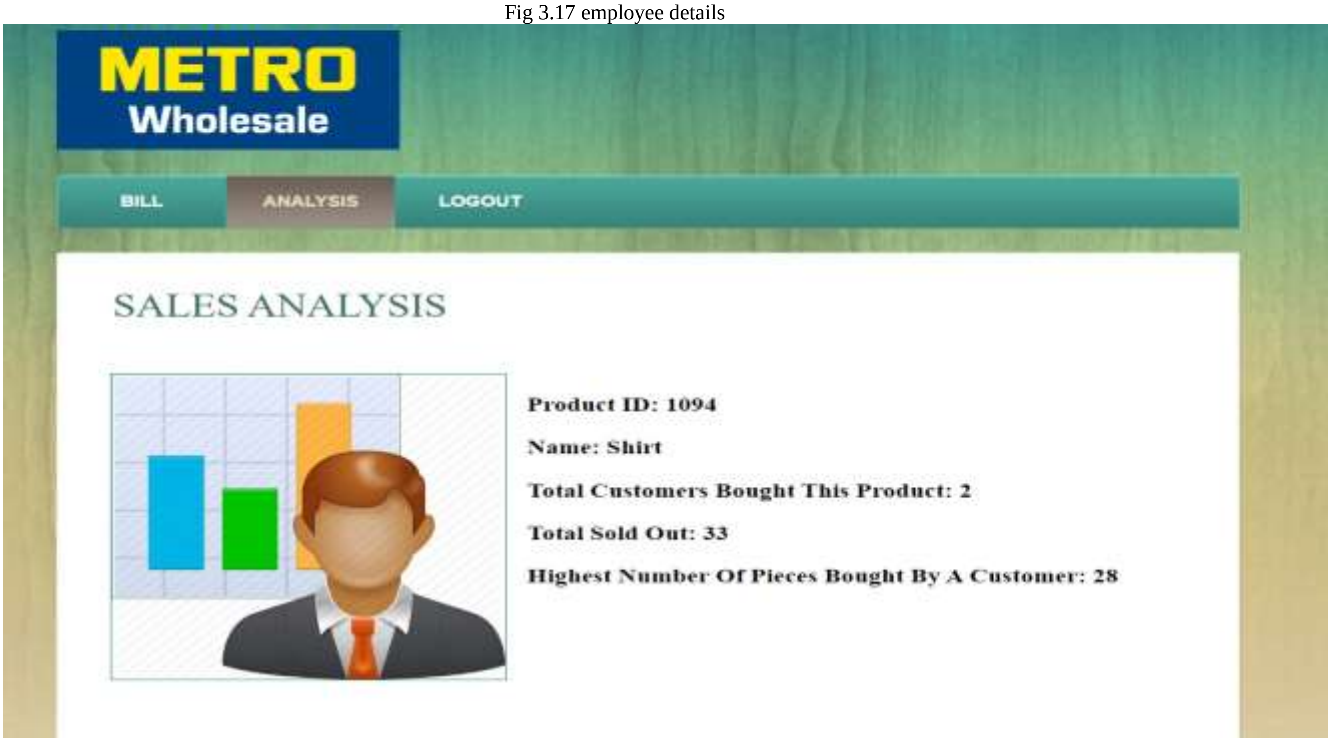
Fig 3.16 product details



The employee details are...

EMPLOYEE ID	EMPLOYEE NAME	EMPLOYEE MOBILE NUMBER	WORKING DEPARTMENT
10046	aarya	7695987904	healthcare
1111	Sandy	1234123412	billing
1112	Rakshi	0987654321	Women
123456	sam	1472583690	Health

Fig 3.17 employee details



SALES ANALYSIS

Product ID: 1094

Name: Shirt

Total Customers Bought This Product: 2

Total Sold Out: 33

Highest Number Of Pieces Bought By A Customer: 28

Fig 3.18 sales analysis

CHAPTER – 4

4.1 TOOLS USED

MySQL – MySQL is an open-source relational database management system. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a central component of the LAMP open-source web application software stack. LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python".

XAMPP - XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages.

VISUAL STUDIO CODE – Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It provides basic support for HTML, CSS, PHP, Java Script ,etc programming out of the box. It is opensource software.

Web Browser - A web browser is a software application for retrieving, presenting and traversing information resources on the World Wide Web. An information resource is identified by a Uniform Resource Identifier (URI/URL) that may be a web page, image, video or other pieceof content.

ERD PLUS and DRAW.IO –ERD Plus is a web-based database modelling tool that lets you quickly and easily create.

4.2 REFERENCE

Textbooks:

1) Database systems Models, languages, design and application programming, RamezElmasri and Shamkant B, 6th Edition, Pearson.
2) Database management systems, Ramkrishna and Gehrke, 3rd edition, 2014, McGraw Hill.

3) Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill 2007.

3) Jim Keogh: J2EE-TheCompleteReference, McGraw Hill, 2007.

Websites:

1) www.w3schools.com
2) Google Chrome.
3) YouTube.
4) www.draw.io
5) www.mysql.com
6) www.apachefriends.org

4.3 CONCLUSION

A sales management system is a tool that helps salespeople get organized, manage their contacts better, track their sales deals more efficiently and save them time. In short - a tool that helps salespeople sell more. A sales CRM should also give sales managers a simple yet clear overview of how their team is performing - a tool that helps their team sell better. So whether you're a small-business owner, a salesperson or a sales manager, a good sales management software will help you drive your sales and reduce headaches.

Advantages

1) Easy to use
2) Good sales reporting
3) Easy To Maintain Stocks

Limitations

1) Detailed information and gathering has to be done to obtain satisfactory results.
2) Implementing the software requires change in business practices.
3) Implementation and maintenance costs run very high.
4) The manager must add a new employee and later his login details can't be communicated through this project and must be communicated personally.

EVALUATION SHEET

Reg.No : URK22AI1046

Name: AARYA B S

Course code: 20CS2016

Course Name: Database Management Systems

S.No	**Rubrics**	**Maximum Marks**	**Marks Obtained**
1	Real-Time Applications Design	40	
Total		40	

Signature of the Faculty-in-charge

Signature of the Examiner1

Signature of the Examiner2