MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Skill Based Mini Project Report

on

INVENTORY MANAGEMENT SYSTEM

Submitted By:

Sanskar Gupta

0901CS201107

Faculty Mentor:

Ms. Jaimala Jha,

Assistant Professor,

Department of Computer Science and Engineering

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

JAN-JUNE 2022

CERTIFICATE

This is certified that **Sanskar Gupta** (0901CS201107) has submitted the project report titled **Inventory Management System** under the mentorship of **Ms. Jaimala Jha**, Assistant Professor, Department of Computer Science and Engineering, in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

Ms. Jaimala Jha
Faculty Mentor
Assistant Professor
Computer Science and Engineering

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Ms. Jaimala Jha, Assistant Professor**, Department of Computer Science and Engineering.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Sanskar
Gupta

Sgupta

0901CS201107
2nd Year,
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Ms. Jaimala Jha**, Assistant Professor, Department of Computer Science and Engineering, for her continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Sanskar Gupta
0901C201107
3rd Year,
Computer
Science and
Engineering

ABSTRACT

Project is related to Inventory Management System The project maintains three levels of users:- Billing Counter Level , Manager Level , Owner Level. Main facilities available in this project are:- We can forecast the sales by analyzing the previous sales statistics. We can get an idea of when we need to order new inventory. We can reduce the chances of any kind of frauds done by the staff members in the inventory. Customer details can be added. Invoice generation. We can keep a track of transactions received through different payment methods**.** We will be using SQL for the implementation of the Inventory Management System.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Abbreviation	**Description**
DBMS	Database Management System
SQL	Structured Query Language

Chapter 1: INTRODUCTION

Inventory management software streamlines all the inventory operations from procurement to sales with real-time tracking and helps gain control across outlets. An effective and reliable inventory management system eliminates error-prone manual tasks with automation for better tracking and increased visibility to run business with high customer satisfaction having optimum inventory available, always.

Inventory management system facilitates businesses to boost their growth, serve customers well, and increase sales by automating the key inventory operations. These inventory management tools give a complete view of what's happening in the inventory from fast moving stocks to dead stocks and make data-driven timely pricing decisions without affecting the margin to run business with increased profits.

In order to enjoy complete inventory control, businesses ranging from a single store to multiple-stores rely upon good retail inventory software. From using inventory software to keep up well with the changing demand and make more sales for small businesses, to managing the supply chain across warehouses and outlets from a centralized digital platform, stock inventory management software serves as an essential tool to run a business hassle-free.

Chapter 2: OVERVIEW

Project is related to Inventory Management System

- The project maintains three levels of users:-
    - Billing Counter Level
    - Manager Level
    - Owner Level

- Main facilities available in this project are:-
    - We can forecast the sales by analyzing the previous sales statistics.
    - We can get an idea of when we need to order new inventory.
    - We can reduce the chances of any kind of frauds done by the staff members in the inventory.
    - Customer details can be added.
    - Invoice generation.
    - We can keep a track of transactions received through different payment methods.

Chapter 3: Scope

- This will help us in maintaining the exact count of any product.
- Can help us to set a minimum quantity of any product below which we can order the product from the manufacturer.
- Can reduce duplicate entries

Benefits of Inventory Management System are

- Forecast customer demands and plan the supply of stocks
- Prevent loss of revenue, having the right stock in the store
- Optimize stocks and fulfil the customer orders without fail
- Empower sales force to increase sales with 100% stocks
- Reduce the time, efforts and cost involved in stock audits
- Eliminate manual errors amidst fluctuating stock level.

Chapter 4: SQL CODE IMPLEMENTATION

This application will have different front ends for different kinds of users. The person who is sitting on the billing counter will have access to only modify the quantity of any product i.e. he/she can either generate an invoice for any sold product or can generate a return note for any returns from any customer. The manager will have the access to modify the rates if there exists any dynamic price inventory. The owner of the firm will have the access to generate the final report which will consist of sales done on any particular day, the total sales on any particular counter or by any salesperson.

4.1: CREATE TABLE

```
SQL> create table brands(
  2 bid number(5),
  3 bname varchar(20)
  4 );

Table created.

SQL> alter table brands
  2 add primary

key(bid); Table altered.

SQL> create table inv_user(
  2   user_id varchar(20),
  3   name varchar(20),
  4   password varchar(20),
  5   last_login timestamp,
  6   user_type varchar(10)
  7     );

Table created.

SQL> create table categories(
  2 cid number(5),
  3                 category_name
  varchar(20) 4 );
```

```
Table created.

SQL> alter table categories
  2 add primary

key(cid); Table altered.

SQL> alter table inv_user
  2 add primary key(user_id);

  Table altered.

SQL> create table product(
  2 pid number(5) primary key,
  3 cid number(5) references categories(cid),
  4 bid number(5) references brands(bid),
  5 sid number(5),
  6 pname varchar(20),
  7 p_stock number(5),
  8 price number(5),
  9 added_date date);

Table created.

SQL> create table stores(
  2 sid number(5),
  3 sname varchar(20),
  4 address varchar(20),
 5              mobno
 number(10) 6 );

Table created.

SQL> alter table stores
  2 add primary key(sid);

Table altered.

SQL> alter table product
  2 add foreign key(sid)references stores(sid);

Table altered.

SQL> create table provides(
  2 bid number(5)references brands(bid),
  3 sid number(5)references stores(sid),
  4 discount number(5));

Table created.

SQL> create table customer_cart(
  2 cust_id number(5) primary key,
  3 name varchar(20),
  4              mobno
  number(10) 5 );
Table created.

SQL> create table select_product(
  2 cust_id number(5) references customer_cart(cust_id),
  3 pid number(5)references product(pid),
  4 quantity number(4)
  5 );

Table created.

SQL> create table transaction(
  2 id number(5) primary key,
  3 total_amount number(5),
  4 paid number(5),
  5 due number(5),
  6 gst number(3),
  7 discount number(5),
  8 payment_method varchar(10),
  9        cart_id        number(5)
 references
 customer_cart(cust_id) 10 );

Table created.


SQL> create table invoice(
  2 item_no number(5),
  3 product_name varchar(20),
  4 quantity number(5),
  5 net_price number(5),
  6                  transaction_id
  number(5)references
  transaction(id) 7 );
```

4.2: INSERTION

4.2.1: INSERT INTO BRANDS

```
SQL> insert into brands values(
  2 '&bid'
  3 ,
  4 '&bname');
Enter value for bid: 1
old   2: '&bid'
new  2: '1'
Ent
er
valu
e
for
bna
me:
App
le
old
4:
'&b
nam
e')


new  4: 'Apple')

1 row created.
1 row created.

SQL> insert into brands values(2,'Samsung');

1 row created.

SQL> insert into brands values(3,'Nike');

1 row created.

SQL> insert into brands values(4,'Fortune');

1 row created.
```

4.2.2: INSERT INTO INV_USER

```
SQL> insert into inv_user values(
  2 '&user_id',
  3 '&name',
  4 '&password',
  5 '&last_login',
  6 '&user_type');
Enter value for
user_id:
vidit@gmail.co
m old 2:
'&user_id',
new 2:
'vidit@gmail.com',
Enter value for
name: vidit
old  3: '&name',
new  3: 'vidit',
Enter
value
for
pass
word:
1234
old 4:
'&pas
swor
d',
new  4: '1234',
Enter value for
last_login: 31-
oct-18 12:40 old
5: '&last_login',
new  5: '31-oct-18 12:40',
Enter
value
for
user_ty
pe:
admin
old 6:
'&user
_type')
```

```
new  6: 'admin')

1 row created.

SQL> insert into inv_user values('harsh@gmail.com','Harsh Khanelwal','1111','30-oct-
18 10:20','Manager');

1 row created.


SQL> insert into inv_user
values('prashant@gmail.com','Prashant','0011','29-oct-18
10:20','Accountant');

1 row created.
```

4.2.3: INSERT INTO CATEGORIES

```
SQL> insert into categories values(
  2 '&cid',
  3
'&category_name');
Enter value for cid:
1 old 2: '&cid',
new  2: '1',
Enter value for
category_name:
Electroincs old  3:
'&category_name'
)
new  3: 'Electroincs')

1 row created.

SQL> insert into categories values(2,'Clothing');

1 row created.

SQL> insert into categories values(3,'Grocey');

1 row created.
```

4.2.4: INSERT INTO STORE

```
SQL> insert into stores values(
  2 '&sid',
  3 '&sname',
  4 '&address',
  5 '&mobno');
Enter value for sid:
1 old 2: '&sid', new  2: '1',
Enter
value  for
sname:
Ram
kumar
old      3:
'&sname'
,
new  3: 'Ram kumar',
Enter     value
for     address:
Katpadi
vellore old   4:
'&address',
new  4: 'Katpadi vellore',
Enter
value    for
mobno:
99999999
99  old    5:
'&mobno')

new  5: '9999999999')

1 row created.


SQL> insert into stores values(2,'Rakesh kumar','chennai',8888555541);

1 row created.

SQL> insert into stores values(3,'Suraj','Haryana',7777555541);

1 row created.
```

4.2.5: INSERT INTO PRODUCT

```
SQL> insert into product values(
  2 '&pid',
  3 '&cid',
  4 '&bid',
  5 '&sid',
  6 '&pname',
  7 '&p_stock',
  8 '&price',
  9     '&added_date');
Enter value for pid: 1
old 2: '&pid',
new  2: '1',
Enter  value  for  cid:  1
old  3: '&cid',
new  3: '1',
Enter  value  for  bid:  1  old   4:
'&bid',
new  4: '1',
Enter value for sid: 1 old
5: '&sid',
new  5: '1',
Enter
value
for
pname:
IPHO
NE old
6:
'&pna
me',
new  6: 'IPHONE',
Enter
value    for
p_stock:  4
old         7:
'&p_stock',
new  7: '4',
Enter  value  for
price:      45000
old  8: '&price',
new  8: '45000',
Enter value
for
added_date:
31-oct-18
old 9:
'&added_dat
e')
new 9: '31-
oct-18')

1 row created.
SQL> insert into product

values(2,1,1,1,'Airpods',3,19000,'27-oct-

18'); 1 row created.


SQL> insert into product values(3,1,1,1,'Smart Watch',3,19000,'27-oct-18');

1 row created.

SQL> insert into product values(4,2,3,2,'Air Max',6,7000,'27-oct-18');

1 row created.

SQL> insert into product values(5,3,4,3,'REFINED OIL',6,750,'25-oct-18');

1 row created.
```

4.2.6: INSERT INTO PROVIDES

```
SQL> insert into provides values(1,1,12);

1 row created.

SQL> insert into provides values(2,2,7);

1 row created.

SQL> insert into provides values(3,3,15);

1 row created.
```

```
SQL> insert into provides values(1,2,7);

1 row created.
SQL> insert into provides values(4,2,19);

1 row created.

SQL> insert into provides values(4,3,20);

1 row created.
```

4.2.7: INSERT INTO CUSTOMER_CART

```
SQL> insert into customer_cart values(
  2 '&cust_id',
  3 '&name',
  4 '&mobno');
Enter  value
for  cust_id:
1    old    2:
'&cust_id',
new  2: '1',
Enter value for name: Ram
old   3: '&name',
new  3: 'Ram',
Enter
value    for
mobno:
98765432
10   old   4:
'&mobno')
new  4: '9876543210')

1 row created.

SQL> insert into customer_cart values(2,'Shyam',7777777777);
```

```
1 row created.

SQL> insert into customer_cart values(3,'Mohan',7777777775);

1 row created.
```

4.2.8: INSERT INTO SELECT_PRODUCT

```
SQL> insert into select_product values(
  2 '&cust_id',
  3 '&pid',
  4 '&quantity');
Enter     value
for  cust_id:  1
old             2:
'&cust_id',
new  2: '1',
Enter value for pid: 2
old   3: '&pid',
new  3: '2',
Enter  value  for  quantity:
2 old 4: '&quantity')
new  4: '2')

1 row created.

SQL> insert into select_product values(1,3,1);

1 row created.


SQL> insert into select_product values(2,3,3);

1 row created.

SQL> insert into select_product values(3,2,1);

1 row created.
```

4.2.9: INSERT INTO TRANSACTIONS

```
SQL> insert into transaction values(
  2 '&id',
  3 '&total_amount',
  4 '&paid',
  5 '&due',
  6 '&gst',
  7 '&discount',
  8 '&payment_method',
  9 '&cart_id');
Enter value for id: 1
old   2: '&id',
new  2: '1',
Enter
value    for
total_amo
unt:  57000
old         3:
'&total_a
mount',
new  3: '25000',
Enter value for paid: 2000
old   4: '&paid',
new  4: '20000',
Enter value for due: 5000
old   5: '&due',
new  5: '5000',
Enter value for gst: 350
old   6: '&gst',
new  6: '350',
Enter value for discount: 350 old 7: '&discount',
new  7: '350',
Enter value for payment_method: card old 8: '&payment_method',
new  8: 'card',
Enter value for cart_id: 1 old 9: '&cart_id')
new  9: '1')

1 row created.

insert into transaction values(2,57000,57000,0,570,570,'cash',2);

SQL> insert into transaction values(3,19000,17000,2000,190,190,'cash',3);

1 row created. SQL> insert into transaction

values(3,19000,17000,2000,190,190,'cash',3);

1 row created.
```

4.3: PL/SQL

4.3.1: Functions

```
SQL> declare
2 due1 number(7);
3 cart_id1 number(7);
4 function get_cart(c_id number)return number is
5 begin
6 return (c_id);
7 end;
8 begin
9 cart_id1:=get_cart('&c_id');
10 select due into due1 from transaction where cart_id=cart_id1;
11 dbms_output.put_line(due1);
12 end;
13   /
Enter value for c_id: 1
old   9: cart_id1:=get_cart('&c_id'); new   9: cart_id1:=get_cart('1'); 5000
PL/SQL procedure successfully completed.
```

4.3.2: Cursors

```
SQL> DECLARE
2 p_id product.pid%type;
3 p_name product.pname%type;
4 p_stock product.p_stock%type;
5 cursor p_product is
6 select pid,pname ,p_stock from product;
7 begin
8 open p_product;
9 loop
10 fetch p_product into p_id,p_name,p_stock;
11 exit when p_product%notfound;
12 dbms_output.put_line(p_id||' '||p_name||' '||p_stock);
13 end loop;
14 close p_product;
15 end;
16 /
1 IPHONE 4
2 Airpods 3
```

```
3 Smart Watch 3
4 Air Max 6
5 REFINED OIL 6
PL/SQL procedure successfully completed.
```

4.3.3: Procedure

```
SQL> DECLARE
2   a number;
3   b number;
4 PROCEDURE check_stock(x IN number) IS
5 BEGIN
6   IF x < 2 THEN
7     dbms_output.put_line('Stock is Less');
8   ELSE
9     dbms_output.put_line('Enough
Stock'); 10 END IF;
11 END;
12 BEGIN
13   b:='&b';
14 select p_stock into a from product where pid=b;
15 check_stock(a);
16 END
7 /
Enter value for b: 2 old 13:    b:='&b'; new 13:   b:='2'; Enough Stock

PL/SQL procedure successfully completed.
```

Chapter 5: RESULTS AND CONCLUSION

In this project we developed a complete back end software in which we can update the stock, modify stock, we can forecast the stock, generate invoice.

From this application we can get an update that if a particular inventory or stock is less than some pre-fixed quantity then it'll be easy for the manager/owner to reorder the product from the supplier to overcome the "Out of Stock" stage. In addition to this it can also help us to manage the warehouses, add warehouses which can be proved as very useful features.

We can have complete customer details which can help us to retrieve the order details of regular customers.

From this program we can also keep a track of transactions performed by different customers/clients.

We can also get an idea of how much funds we received from different payment methodologies.

This application will keep a high inventory turnover ratio to ensure our products aren't spoiling, becoming obsolete for our working capital. It'll help us to calculate how many times inventory sells in a year and see where we can make better use of our resources.

References

- https://www.javatpoint.com/
- https://www.geeksforgeeks.org/
- https://www.youtube.com/
- https://www.softwaretestinghelp.com/database-management-software/