

**A Project Report
On**

IT INVENTORY MANAGEMENT SYSTEM

Submitted by:

Akshat Manhar

Under the guidance of

Mr. Prabhat Das
(PGT-IP)



**ARMY PUBLIC SCHOOL JORHAT
MILITARY STATION SARAIBARI JORHAT
Assam-785004**

CERTIFICATE BY PRINCIPAL

This is to certify that this project report entitled “**IT INVENTORY MANAGEMENT SYSTEM**” submitted by **Akshat Manhar** to Army Public School Jorhat has been examined and evaluated.

The report has been prepared as per the regulations of CBSE and qualifies to be accepted.

Date:

Place:

Mrs. Firdausi Sultana Hazarika
(Principal)
Army Public School Jorhat

CERTIFICATE BY EXAMINERS

This is to certify that this project report entitled “**IT INVENTORY MANAGEMENT SYSTEM**” is the bonafide work of who carried out the project work under my supervision and guidance.

To the best of my knowledge, the matter embodied in the report has not been submitted to any other institute for the award of any other degree.

Date:

Place:

(External Examiner)

Mr. Prabhat Das
(Internal Examiner)

ACKNOWLEDGEMENT

I take this opportunity to extend my heart full gratitude to Army Public School Jorhat for providing me the opportunity.

I am highly grateful to my guide Mr. Prabhat Das, PGT-IP, Army Public School Jorhat for giving us the opportunity to work under him and providing us an ample guidance and support through the project.

Lastly, I would also like to thank the authors whose publications guided us regarding our project.

DECLARATION

I admit that this report is of my own work and all the sources of the information used in this report have fully acknowledged.

I hereby declare that the dissertation work entitled “**IT INVENTORY MANAGEMENT SYSTEM**” submitted to the Army Public School Jorhat, is prepared by me and was not submitted to any other institution for award of any other degree.

Date:

Place:

Signature

Abstract

Inventory refers to all the goods, items and materials purchased or manufactured by business for sale to the customer to make profit.

Inventory management is all about tracking and controlling of business inventory right from manufacturing, buying to storing and using. It controls the entire flow of goods from purchasing to sale and ensures that you always have the right quantities of the right item in the right location at the right time.

Inventory Management System is an application which refers to Inventory Management developed for small business. It can be used by business to manage Inventory using a computerized system where they can manage details of purchase, sale, products and customers. They can also analyze data by visualization.

IT Inventory Management is a system that integrates technology and practices to monitor and maintain inventory or assets.

Contents

• Tools and Library	1
• Introduction	3
• Project overview	4
• Source code	18
• Commands in SQL	28
• Conclusion and Future work	29
• References	30

List Of Figures

Figure	Page No.
• Fig 1 & Fig 2	04
• Fig 3 & Fig 4	05
• Fig 5, 6, 7 & 8	06
• Fig 9 & Fig 10	07
• Fig 11.1 & Fig 11.2	08
• Fig 12.1 & Fig 12.2	09
• Fig 13.1 & Fig 13.2	10
• Fig 14.1 & Fig 14.2	11
• Fig 15.1 & Fig 15.2	12
• Fig 16.1 & Fig 16.2	13
• Fig 17.1 & Fig 17.2	14
• Fig 18.1 & Fig 18.2	15
• Fig 18.3 & Fig 19.1	16
• Fig 19.2 & Fig 19.3	17

Tools and Libraries Used

MySQL

MySQL is the world's most used relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is named after developer Michael Widenius' daughter, My. The phrase "SQL" stands for "structured query language."

Open source projects that require a full-featured database management system often use MySQL. Applications which use MySQL databases include:

TYPO3, Joomla, WordPress, phpBB, Drupal, and other software built on the LAMP software stack. MySQL is also used in many high-profile, large-scale Web products, including Wikipedia, Google (though not for searches), Facebook, and Twitter.

The free open-source version of MySQL is commonly referred to as the MySQL Community Edition. For commercial use, several paid editions are also available, offering additional functionality.[1]

PyCharm

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.[5] It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license. [2]

Pandas

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself. Wes McKinney started building what would become pandas at AQR Capital while he was a researcher there from 2007 to 2010. [3]

NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing NumPy array into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. [4]

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter. Since then it has an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell.[5]

Introduction

The theme of our project is '**IT INVENTORY MANAGEMENT SYSTEM**'. This project is fine thought to make complex procedure of managing IT inventory in an easy manner which is systematic, modular designed, selective menu based user display. The modular design and constructed is very much user oriented in which user can easily understand the tools and can do edit of his own choice. The system is not any though more and does not possesses many application but it is made by focusing on maintaining IT record in a computerized manner rather than time taking and cumbersome manual system.

The project software application that can easily handle by minimum educated and simple computer knowledge person without any option of error.

Project Overview

```
mysql> desc registration_details;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	char(30)	YES		NULL	
email	varchar(50)	YES		NULL	

```
3 rows in set (0.24 sec)
```



```
mysql> select * from registration_details;
```

id	name	email
1	Prabhat Das	prabhat@apsjorhat.org
2	Akshat	manharakshat@gmail.com
3	Joseph	josephnath73@gmail.com
4	Alex	hirenmanu2@gmail.com
5	Amit	onlineusage377@gmail.com
6	santosh	santoshphukon5@gmail.com
7	apsjorhat	4151@apsjorhat.org
12	Bisi	bishalsaikia2000@gmail.com
13	Amit	bishalsaikia2004@gmail.com
14	Kunal	kunal666kaushik@gmail.com

```
10 rows in set (0.00 sec)
```

Figure 1: All the information regarding the users are stored in this table.

```
mysql> desc product_table;
```

Field	Type	Null	Key	Default	Extra
s_no	int	YES		NULL	
product_id	varchar(10)	YES		NULL	
product_name	varchar(40)	YES		NULL	
cat_id	varchar(10)	YES		NULL	
brand	char(10)	YES		NULL	
price	int	YES		NULL	
purchase_date	date	YES		NULL	

```
7 rows in set (0.01 sec)
```



```
mysql> select * from product_table;
```

s_no	product_id	product_name	cat_id	brand	price	purchase_date
1	P1001	HP 144 Hz Monitor	D101	HP	10999	2017-11-03
2	P1011	Zebronics Zeb Wired Mouse	D102	Zebronics	199	2019-01-02
3	P1111	Zebronics Zeb Viking CPU	D103	Zebronics	3999	2018-09-14
4	P2011	boAt Rockerz 510	D104	Boat	1199	2020-12-19
5	P2001	Canon PIXMA Inkjet	D105	Canon	2499	2021-05-20
6	P2000	Vertiv Liebert UPS	D106	Vertiv	2199	2017-05-20
7	P2111	JBL bar 91 wireless speaker	D107	JBL	89999	2021-05-13
8	P3000	Dell KB216 Wired Keyboard	D108	Dell	299	2017-07-18
9	P3001	Samsung S pen pro	D109	samsungg	16999	2022-09-11
10	P3011	Secureye S_xpon 4G Router	D110	Secureye	2999	2021-10-22
11	P3111	Urbntec Type_B Cable	D111	Urbntec	89	2022-11-20
12	P4000	boAt Deuce 300 Type_C USB	D112	boAt	329	2022-06-10
13	P4001	Dell 1TB 7200 RPM Hard_Drive	D113	Dell	6999	2019-04-30
14	P4011	Hp X760W 64GB USB 3 1 PenDrive	D114	HP	759	2020-12-07
15	P4111	Oculus Quest2 Advanced Headset	D115	Oculus	40999	1700-07-05
16	P5001	Dell 18.5 inch Monitor	D101	Dell	8499	2019-10-01
17	P5011	Lenovo 130 Wireless Mouse	D102	Lenovo	599	2020-11-10
18	P6001	HP Intel Core i5 CPU	D103	HP	14999	2020-09-20
19	P5111	HP x1000 Mouse	D102	HP	399	2020-12-13

```
19 rows in set (0.00 sec)
```

Figure 2: All the information regarding the products are stored in this table.

```
mysql> desc category_table;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| cat_id | varchar(30) | NO   | PRI | NULL    |       |
| items  | varchar(30) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from category_table;
+-----+-----+
| cat_id | items      |
+-----+-----+
| D101   | Monitor    |
| D102   | Mouse      |
| D103   | CPU        |
| D104   | Printers   |
| D105   | Headphones |
| D106   | UPS        |
| D107   | Speakers   |
| D108   | Keyboards  |
| D109   | Stylus     |
| D110   | WiFi Router |
| D111   | Cables     |
| D112   | USB        |
| D113   | Hard-Drive |
| D114   | Pen-Drive  |
| D115   | AR & VR Headset |
+-----+-----+
15 rows in set (0.00 sec)
```

Figure 3: All the information regarding the category of products are stored in this table.

```
mysql> desc deleted_records;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| s_no  | int       | YES  |     | NULL    |       |
| product_id | varchar(20) | YES  |     | NULL    |       |
| product_name | varchar(40) | YES  |     | NULL    |       |
| cat_id | varchar(10) | YES  |     | NULL    |       |
| brand | char(20)   | YES  |     | NULL    |       |
| price | int       | YES  |     | NULL    |       |
| purchase_date | date      | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.05 sec)

mysql> select * from deleted_records;
+-----+-----+-----+-----+-----+-----+-----+
| s_no | product_id | product_name      | cat_id | brand | price | purchase_date |
+-----+-----+-----+-----+-----+-----+-----+
| 16   | P5000      | Acer EK220Q Monitor | D101   | Acer  | 7999  | 2019-12-01    |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Figure 4: All the information regarding the deleted users are stored in this table.

```

+-----+
| WELCOME TO THE IT INVENTORY MANAGEMENT SYSTEM |
+-----+
----- SELECT THE OPTION TO BEGIN ! -----
+-----+
| > Press 1 to Login |
| > Press 2 to Register |
+-----+
Enter your Choice:

```

Figure 5: When the home page is opened two login options are presented to the user.

```

Enter your Choice: 1
ENTER THE USER ID TO LOGIN: akshat@gmail.com
Please wait we are sending the OTP to the given ID.....
Enter the OTP: 123456
Verifying the OTP....
+-----+
| HEY ! AKSHAT LOGIN SUCCESSFUL SELECT AN OPTION TO BEGIN |
+-----+

```

Figure 6: When an existing user chooses option 1 the program asks for his/her email address, on entering the email-id the user receives an OTP from the program. Entering the OTP completes the login procedure.

```

Enter your Choice: 2
+-----+
| ENTER YOUR NAME: Akshat |
| ENTER EMAIL ID: akshat@gmail.com |
+-----+
***** REGISTRATION COMPLETED ! *****
ENTER THE USER ID TO LOGIN: akshat@gmail.com
Please wait we are sending the OTP to the given ID.....
Enter the OTP: 456789
Verifying the OTP....

```

Figure 7: Login procedure for new users.

```

+-----+
| HEY ! AKSHAT LOGIN SUCCESSFUL SELECT AN OPTION TO BEGIN |
+-----+
+-----+
| > Press a to view all products |
| > Press b to See Category Wise Quantity |
| > Press c to Delete Record |
| > Press d to Update S_no |
| > Press e to Update product_id |
| > Press f to Update product_name |
| > Press g to Update cat_id |
| > Press h to Update brand |
| > Press i to Update price |
| > Press j to Update purchase_date |
| > Press k to Insert New Product |
| > Type logout to end |
+-----+
Enter your Choice: |

```

Figure 8: After the login is completed the user is presented with the following list of options.

The data can be retrived by entering the options as given in the list. The output is received as follows:

```
Enter your Choice: a
```

S.No.	Product_ID	Product_Name	Cat_ID	Brand	Price	Purchase_Date
1	P1001	HP 144 Hz Monitor	D101	HP	10999	2017-11-03
2	P1011	Zebtronics Zeb Wired Mouse	D102	Zebtronics	199	2019-01-02
3	P1111	Zebtronics Zeb Viking CPU	D103	Zebtronics	3999	2018-09-14
4	P2011	boAt Rockerz 510	D104	Boat	1199	2020-12-19
5	P2001	Canon PIXMA Inkjet	D105	Canon	2499	2021-05-20
6	P2000	Vertiv Liebert UPS	D106	Vertiv	2199	2017-05-20
7	P2111	JBL bar 91 wireless speaker	D107	JBL	89999	2021-05-13
8	P3000	Dell KB216 Wired Keyboard	D108	Dell	299	2017-07-18
9	P3001	Samsung S pen pro	D109	samsungg	16999	2022-09-11
10	P3011	Secureye S_xpon 4G Router	D110	Secureye	2999	2021-10-22
11	P3111	Urbntec Type_B Cable	D111	Urbntec	89	2022-11-20
12	P4000	boAt Deuce 300 Type_C USB	D112	boAt	329	2022-06-10
13	P4001	Dell 1TB 7200 RPM Hard_Drive	D113	Dell	6999	2019-04-30
14	P4011	Hp X760W 64GB USB 3 1 PenDrive	D114	HP	759	2020-12-07
15	P4111	Oculus Quest2 Advanced Headset	D115	Oculus	40999	1700-07-05
16	P5001	Dell 18.5 inch Monitor	D101	Dell	8499	2019-10-01
17	P5011	Lenovo 130 Wireless Mouse	D102	Lenovo	599	2020-11-10
18	P6001	HP Intel Core i5 CPU	D103	HP	14999	2020-09-20
19	P5111	HP x1000 Mouse	D102	HP	399	2020-12-13

```

+-----+
|                                     Do you Want to go to Menu Again?                                     |
| Type y to go Back                                                         Type n to exit |
+-----+

```

Figure 9: Output for option ‘a’:(To view all products)

```
Enter your Choice: b
```

Cat_Id	Items	Quantity
D101	Monitor	2
D102	Mouse	3
D103	CPU	2
D104	Printers	1
D105	Headphones	1
D106	UPS	1
D107	Speakers	1
D108	Keyboards	1
D109	Stylus	1
D110	WiFi Router	1
D111	Cables	1
D112	USB	1
D113	Hard-Drive	1
D114	Pen-Drive	1
D115	AR & VR Headset	1

```

+-----+
|                                     Do you Want to go to Menu Again?                                     |
| Type y to go Back                                                         Type n to exit |
+-----+
ENTER YOUR CHOICE:

```

Figure 10: Output for option ‘b’:(To See Category Wise Quantity)

```

Enter your Choice: c
| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|-----|-----|-----|-----|-----|-----|
| 1 | P1001 | HP 144 Hz Monitor | D101 | HP | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 18 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
| 19 | P5111 | HP x1000 Mouse | D102 | HP | 399 | 2020-12-13 |
Enter the S.No of Product you want to delete
19

```

Figure 11.1: Output for option ‘c’:(To Delete Record).

```

| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|-----|-----|-----|-----|-----|-----|
| 1 | P1001 | HP 144 Hz Monitor | D101 | HP | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 18 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
***** DELETED SUCCESSFULLY ! *****

```

Figure 11.2: Output for option ‘c’:(Updated Table).


```

Enter your Choice: d
| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|:-----|:-----|:-----|:-----|-----|:-----|
| 1 | P1001 | HP 144 Hz Monitor | D101 | HP | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 18 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
Enter the Old S_no:
38
Enter the New S_no:
19

```

Figure 12.1: Output for option ‘d’:(To Update the S No of product).

```

| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|:-----|:-----|:-----|:-----|-----|:-----|
| 1 | P1001 | HP 144 Hz Monitor | D101 | HP | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 19 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
***** UPDATED SUCCESSFULLY ! *****

```

Figure 12.2: Output for option ‘d’:(Updated Table).

```

Enter your Choice: e
| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|-----|-----|-----|-----|-----|-----|
| 1 | P1001 | HP 144 Hz Monitor | D101 | HP | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 19 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
Enter the S_no of Product_ID whose value you want to Update:
19
Enter the new Product_ID
P6000

```

Figure 13.1: Output for option ‘e’:(To Update the Product ID of product).

```

| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|-----|-----|-----|-----|-----|-----|
| 1 | P1001 | HP 144 Hz Monitor | D101 | HP | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 19 | P6000 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
***** UPDATED SUCCESSFULLY ! *****

```

Figure 13.2: Output for option ‘e’(Updated Table).


```

Enter your Choice: f
| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|-----|-----|-----|-----|-----|-----|
| 1 | P1001 | HP 144 Hz Monitor | D101 | HP | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 19 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
Enter the S_no of Product_Name whose value you want to Update:
1
Enter the new Product_Name
AOC C32 G2E LED Monitor

```

Figure 14.1: Output for option ‘f’:(To Update Product Name).

```

| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|-----|-----|-----|-----|-----|-----|
| 1 | P1001 | AOC C32 G2E LED Monitor | D101 | HP | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 19 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
***** UPDATED SUCCESSFULLY ! *****

```

Figure 14.2: Output for option ‘f’:(Updated Table).

S.No.	Product_ID	Product_Name	Cat_ID	Brand	Price	Purchase_Date
1	P1001	AOC C32 G2E LED Monitor	D101	HP	10999	2017-11-03
2	P1011	Zebronics Zeb Wired Mouse	D102	Zebronics	199	2019-01-02
3	P1111	Zebronics Zeb Viking CPU	116	Zebronics	3999	2018-09-14
4	P2011	boAt Rockerz 510	D104	Boat	1199	2020-12-19
5	P2001	Canon PIXMA Inkjet	D105	Canon	2499	2021-05-20
6	P2000	Vertiv Liebert UPS	D106	Vertiv	2199	2017-05-20
7	P2111	JBL bar 91 wireless speaker	D107	JBL	89999	2021-05-13
8	P3000	Dell KB216 Wired Keyboard	D108	Dell	299	2017-07-18
9	P3001	Samsung S pen pro	D109	samsungg	16999	2022-09-11
10	P3011	Secureye S_xpon 4G Router	D110	Secureye	2999	2021-10-22
11	P3111	Urbntec Type_B Cable	D111	Urbntec	89	2022-11-20
12	P4000	boAt Deuce 300 Type_C USB	D112	boAt	329	2022-06-10
13	P4001	Dell 1TB 7200 RPM Hard_Drive	D113	Dell	6999	2019-04-30
14	P4011	Hp X760W 64GB USB 3 1 PenDrive	D114	HP	759	2020-12-07
15	P4111	Oculus Quest2 Advanced Headset	D115	Oculus	40999	1700-07-05
16	P5001	Dell 18.5 inch Monitor	D101	Dell	8499	2019-10-01
17	P5011	Lenovo 130 Wireless Mouse	D102	Lenovo	599	2020-11-10
19	P6001	HP Intel Core i5 CPU	D103	HP	14999	2020-09-20

Enter the S_no of cat_ID whose value you want to Update:
3
Enter the new cat_ID
0116

Figure 15.1: Output for option 'g':(To Update Cat ID).

S.No.	Product_ID	Product_Name	Cat_ID	Brand	Price	Purchase_Date
1	P1001	AOC C32 G2E LED Monitor	D101	HP	10999	2017-11-03
2	P1011	Zebronics Zeb Wired Mouse	D102	Zebronics	199	2019-01-02
3	P1111	Zebronics Zeb Viking CPU	D116	Zebronics	3999	2018-09-14
4	P2011	boAt Rockerz 510	D104	Boat	1199	2020-12-19
5	P2001	Canon PIXMA Inkjet	D105	Canon	2499	2021-05-20
6	P2000	Vertiv Liebert UPS	D106	Vertiv	2199	2017-05-20
7	P2111	JBL bar 91 wireless speaker	D107	JBL	89999	2021-05-13
8	P3000	Dell KB216 Wired Keyboard	D108	Dell	299	2017-07-18
9	P3001	Samsung S pen pro	D109	samsungg	16999	2022-09-11
10	P3011	Secureye S_xpon 4G Router	D110	Secureye	2999	2021-10-22
11	P3111	Urbntec Type_B Cable	D111	Urbntec	89	2022-11-20
12	P4000	boAt Deuce 300 Type_C USB	D112	boAt	329	2022-06-10
13	P4001	Dell 1TB 7200 RPM Hard_Drive	D113	Dell	6999	2019-04-30
14	P4011	Hp X760W 64GB USB 3 1 PenDrive	D114	HP	759	2020-12-07
15	P4111	Oculus Quest2 Advanced Headset	D115	Oculus	40999	1700-07-05
16	P5001	Dell 18.5 inch Monitor	D101	Dell	8499	2019-10-01
17	P5011	Lenovo 130 Wireless Mouse	D102	Lenovo	599	2020-11-10
19	P6001	HP Intel Core i5 CPU	D103	HP	14999	2020-09-20

***** UPDATED SUCCESSFULLY ! *****

Figure 15.2: Output for option 'g':(Updated Table).


```

Enter your Choice: h
| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|:-----|:-----|:-----|:-----|:-----|:-----|
| 1 | P1001 | AOC C32 G2E LED Monitor | D101 | HP | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 19 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
Enter the S_no of Brand whose value you want to Update:
1
Enter the new Brand Name
AOC

```

Figure 16.1: Output for option ‘h’:(To Update Brand).

```

| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|:-----|:-----|:-----|:-----|:-----|:-----|
| 1 | P1001 | AOC C32 G2E LED Monitor | D101 | AOC | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 19 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
***** UPDATED SUCCESSFULLY ! *****

```

Figure 16.2: Output for option ‘h’:(Updated Table).

```

Enter your Choice: i
| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|:-----|:-----|:-----|:-----|:-----|:-----|
| 1 | P1001 | AOC C32 G2E LED Monitor | D101 | AOC | 10999 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 18 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
Enter the S_No of product:
1
Enter the New Price:
12000

```

Figure 17.1: Output for option ‘i’:(To Update Price).

```

| S.No. | Product_ID | Product_Name | Cat_ID | Brand | Price | Purchase_Date |
|-----|:-----|:-----|:-----|:-----|:-----|:-----|
| 1 | P1001 | AOC C32 G2E LED Monitor | D101 | AOC | 12000 | 2017-11-03 |
| 2 | P1011 | Zebronics Zeb Wired Mouse | D102 | Zebronics | 199 | 2019-01-02 |
| 3 | P1111 | Zebronics Zeb Viking CPU | D103 | Zebronics | 3999 | 2018-09-14 |
| 4 | P2011 | boAt Rockerz 510 | D104 | Boat | 1199 | 2020-12-19 |
| 5 | P2001 | Canon PIXMA Inkjet | D105 | Canon | 2499 | 2021-05-20 |
| 6 | P2000 | Vertiv Liebert UPS | D106 | Vertiv | 2199 | 2017-05-20 |
| 7 | P2111 | JBL bar 91 wireless speaker | D107 | JBL | 89999 | 2021-05-13 |
| 8 | P3000 | Dell KB216 Wired Keyboard | D108 | Dell | 299 | 2017-07-18 |
| 9 | P3001 | Samsung S pen pro | D109 | samsungg | 16999 | 2022-09-11 |
| 10 | P3011 | Secureye S_xpon 4G Router | D110 | Secureye | 2999 | 2021-10-22 |
| 11 | P3111 | Urbntec Type_B Cable | D111 | Urbntec | 89 | 2022-11-20 |
| 12 | P4000 | boAt Deuce 300 Type_C USB | D112 | boAt | 329 | 2022-06-10 |
| 13 | P4001 | Dell 1TB 7200 RPM Hard_Drive | D113 | Dell | 6999 | 2019-04-30 |
| 14 | P4011 | Hp X760W 64GB USB 3 1 PenDrive | D114 | HP | 759 | 2020-12-07 |
| 15 | P4111 | Oculus Quest2 Advanced Headset | D115 | Oculus | 40999 | 1700-07-05 |
| 16 | P5001 | Dell 18.5 inch Monitor | D101 | Dell | 8499 | 2019-10-01 |
| 17 | P5011 | Lenovo 130 Wireless Mouse | D102 | Lenovo | 599 | 2020-11-10 |
| 18 | P6001 | HP Intel Core i5 CPU | D103 | HP | 14999 | 2020-09-20 |
***** UPDATED SUCCESSFULLY ! *****

```

Figure 17.2: Output for option ‘i’:(Updated Table).

Enter your Choice: j

S.No.	Product_ID	Product_Name	Cat_ID	Brand	Price	Purchase_Date
1	P1001	AOC C32 G2E LED Monitor	D101	AOC	12000	2017-11-03
2	P1011	Zebronics Zeb Wired Mouse	D102	Zebronics	199	2019-01-02
3	P1111	Zebronics Zeb Viking CPU	D103	Zebronics	3999	2018-09-14
4	P2011	boAt Rockerz 510	D104	Boat	1199	2020-12-19
5	P2001	Canon PIXMA Inkjet	D105	Canon	2499	2021-05-20
6	P2000	Vertiv Liebert UPS	D106	Vertiv	2199	2017-05-20
7	P2111	JBL bar 91 wireless speaker	D107	JBL	89999	2021-05-13
8	P3000	Dell KB216 Wired Keyboard	D108	Dell	299	2017-07-18
9	P3001	Samsung S pen pro	D109	samsungg	16999	2022-09-11
10	P3011	Secureye S_xpon 4G Router	D110	Secureye	2999	2021-10-22
11	P3111	Urbntec Type_B Cable	D111	Urbntec	89	2022-11-20
12	P4000	boAt Deuce 300 Type_C USB	D112	boAt	329	2022-06-10
13	P4001	Dell 1TB 7200 RPM Hard_Drive	D113	Dell	6999	2019-04-30
14	P4011	Hp X760W 64GB USB 3 1 PenDrive	D114	HP	759	2020-12-07
15	P4111	Oculus Quest2 Advanced Headset	D115	Oculus	40999	1700-07-05
16	P5001	Dell 18.5 inch Monitor	D101	Dell	8499	2019-10-01
17	P5011	Lenovo 130 Wireless Mouse	D102	Lenovo	599	2020-11-10
18	P6001	HP Intel Core i5 CPU	D103	HP	14999	2020-09-20

Enter the S_no of Purchase_Date whose value you want to Update:
j

Figure 18.1: Output for option 'j':(To Update Purchase Date).

Enter New Day
13
Enter New Month
05
Enter New Year
2020

Figure 18.2: Entering the new Purchase Date.

S.No.	Product_ID	Product_Name	Cat_ID	Brand	Price	Purchase_Date
1	P1001	AOC C32 G2E LED Monitor	D101	AOC	12000	2020-05-13
2	P1011	Zebronics Zeb Wired Mouse	D102	Zebronics	199	2019-01-02
3	P1111	Zebronics Zeb Viking CPU	D103	Zebronics	3999	2018-09-14
4	P2011	boAt Rockerz 510	D104	Boat	1199	2020-12-19
5	P2001	Canon PIXMA Inkjet	D105	Canon	2499	2021-05-20
6	P2000	Vertiv Liebert UPS	D106	Vertiv	2199	2017-05-20
7	P2111	JBL bar 91 wireless speaker	D107	JBL	89999	2021-05-13
8	P3000	Dell KB216 Wired Keyboard	D108	Dell	299	2017-07-18
9	P3001	Samsung S pen pro	D109	samsungg	16999	2022-09-11
10	P3011	Secureye S_xpon 4G Router	D110	Secureye	2999	2021-10-22
11	P3111	Urbntec Type_B Cable	D111	Urbntec	89	2022-11-20
12	P4000	boAt Deuce 300 Type_C USB	D112	boAt	329	2022-06-10
13	P4001	Dell 1TB 7200 RPM Hard_Drive	D113	Dell	6999	2019-04-30
14	P4011	Hp X760W 64GB USB 3 1 PenDrive	D114	HP	759	2020-12-07
15	P4111	Oculus Quest2 Advanced Headset	D115	Oculus	40999	1700-07-05
16	P5001	Dell 18.5 inch Monitor	D101	Dell	8499	2019-10-01
17	P5011	Lenovo 130 Wireless Mouse	D102	Lenovo	599	2020-11-10
18	P6001	HP Intel Core i5 CPU	D103	HP	14999	2020-09-20

***** UPDATED SUCCESSFULLY ! *****

Figure 18.1: Output for option 'j'(Updated Table).

Enter your Choice: **k**

S.No.	Product_ID	Product_Name	Cat_ID	Brand	Price	Purchase_Date
1	P1001	AOC C32 G2E LED Monitor	D101	AOC	12000	2020-05-13
2	P1011	Zebronics Zeb Wired Mouse	D102	Zebronics	199	2019-01-02
3	P1111	Zebronics Zeb Viking CPU	D103	Zebronics	3999	2018-09-14
4	P2011	boAt Rockerz 510	D104	Boat	1199	2020-12-19
5	P2001	Canon PIXMA Inkjet	D105	Canon	2499	2021-05-20
6	P2000	Vertiv Liebert UPS	D106	Vertiv	2199	2017-05-20
7	P2111	JBL bar 91 wireless speaker	D107	JBL	89999	2021-05-13
8	P3000	Dell KB216 Wired Keyboard	D108	Dell	299	2017-07-18
9	P3001	Samsung S pen pro	D109	samsungg	16999	2022-09-11
10	P3011	Secureye S_xpon 4G Router	D110	Secureye	2999	2021-10-22
11	P3111	Urbntec Type_B Cable	D111	Urbntec	89	2022-11-20
12	P4000	boAt Deuce 300 Type_C USB	D112	boAt	329	2022-06-10
13	P4001	Dell 1TB 7200 RPM Hard_Drive	D113	Dell	6999	2019-04-30
14	P4011	Hp X760W 64GB USB 3 1 PenDrive	D114	HP	759	2020-12-07
15	P4111	Oculus Quest2 Advanced Headset	D115	Oculus	40999	1700-07-05
16	P5001	Dell 18.5 inch Monitor	D101	Dell	8499	2019-10-01
17	P5011	Lenovo 130 Wireless Mouse	D102	Lenovo	599	2020-11-10
18	P6001	HP Intel Core i5 CPU	D103	HP	14999	2020-09-20

Figure 19.1: Output for option 'k':(To insert new product).


```

Enter the S_no of New Product
19
Enter the Product_id of New Product
P5111
Enter the Product_Name of New Product
HP x1000 Mouse
Enter the Cat_ID of New_Product
D102
Enter the Brand of New_Product
HP
Enter the Price of New_Product
399
Enter the Purchase_date of New_Product in format (yyyy-mm-dd)
2020-12-13

```

Figure 19.2: Entering the details of new product.

S.No.	Product_ID	Product_Name	Cat_ID	Brand	Price	Purchase_Date
1	P1001	AOC C32 G2E LED Monitor	D101	AOC	12000	2020-05-13
2	P1011	Zebronics Zeb Wired Mouse	D102	Zebronics	199	2019-01-02
3	P1111	Zebronics Zeb Viking CPU	D103	Zebronics	3999	2018-09-14
4	P2011	boAt Rockerz 510	D104	Boat	1199	2020-12-19
5	P2001	Canon PIXMA Inkjet	D105	Canon	2499	2021-05-20
6	P2000	Vertiv Liebert UPS	D106	Vertiv	2199	2017-05-20
7	P2111	JBL bar 91 wireless speaker	D107	JBL	89999	2021-05-13
8	P3000	Dell KB216 Wired Keyboard	D108	Dell	299	2017-07-18
9	P3001	Samsung S pen pro	D109	samsungg	16999	2022-09-11
10	P3011	Secureye S_xpon 4G Router	D110	Secureye	2999	2021-10-22
11	P3111	Urbntec Type_B Cable	D111	Urbntec	89	2022-11-20
12	P4000	boAt Deuce 300 Type_C USB	D112	boAt	329	2022-06-10
13	P4001	Dell 1TB 7200 RPM Hard_Drive	D113	Dell	6999	2019-04-30
14	P4011	Hp X760W 64GB USB 3 1 PenDrive	D114	HP	759	2020-12-07
15	P4111	Oculus Quest2 Advanced Headset	D115	Oculus	40999	1700-07-05
16	P5001	Dell 18.5 inch Monitor	D101	Dell	8499	2019-10-01
17	P5011	Lenovo 130 Wireless Mouse	D102	Lenovo	599	2020-11-10
18	P6001	HP Intel Core i5 CPU	D103	HP	14999	2020-09-20
19	P5111	HP x1000 Mouse	D102	HP	399	2020-12-13

***** ITEM INSERTED SUCCESSFULLY ! *****

Figure 19.3: Output for option 'k':(Updated Table).

Source Code

Index:

```
import time
import auth
import pandas as pd
import connector as con
def login():
    a = ("ENTER THE USER ID TO LOGIN: ")
    user_id = input(a)
    fetch_query = "select * from registration_details;"
    con.cursor.execute(fetch_query)
    count = 0
    for i in con.cursor:
        if user_id == i[2]:
            count = count + 1
            print("                                     Please wait we are sending the OTP to the given
ID.....")
            login_stat = auth.auth(user_id)
    if count == 0:
        print("                                     USER NOT REGISTERED !")
        print("      +-----+")
        print("      |                                     Do you Want to Register?
|")
        print("      | Type YES to Register                                     Type NO
to exit |")
        print("      +-----+")
        reg = input()
        reg2 = reg.upper()
        if reg2 == "NO":
            print("Thank You")
            exit()
        elif reg2 == "YES":
            registration()
    return user_id

def registration():
    print("      +-----+")
```

```

name=(" ENTER YOUR NAME: ")
un=input(name)
email=(" ENTER EMAIL ID: ")
ue=input(email)
print("+-----+")
query="insert into registration_details (name,email) values" +"('"+un+"','"+ue+"');"
con.cursor.execute(query)
con.dbc.commit()
print("*****
REGISTRATION COMPLETED ! *****")
login()
menu()

def view_items():
    fetch_query = "select * from product_table;"
    con.cursor.execute(fetch_query)
    data=[]
    for i in con.cursor:
        data.append(i)

df=pd.DataFrame(data,columns=['S.No.','Product_ID','Product_Name','Cat_ID','Brand','Price',
'Purchase_Date'])
print(df.to_markdown(index=False))
print("          +-----+
-----+")
print("          |          Do you Want to go to Menu Again?
|")
print("          | Type y to go Back          Type n
to exit |")
print("          +-----+
-----+")
rm2 = ("ENTER YOUR CHOICE:")
rm = (input(rm2))
if rm == "y":
    menu()
elif rm == "n":
    print("Thank You")
    exit()

def cat_quantity():
    fetch_query = "select category_table.cat_id,category_table.items,count(*) as quantity from
category_table,product_table where product_table.cat_id=category_table.cat_id group by
cat_id;"
    con.cursor.execute(fetch_query)

```

```

data=[]
for i in con.cursor:
    data.append(i)
df=pd.DataFrame(data,columns=['Cat_Id','Items','Quantity'])
print(df.to_markdown(index=False))
print("
+-----+
-----+")
print("
|
| Do you Want to go to Menu Again?
|")
print("
| Type y to go Back
| Type n
to exit |")
print("
+-----+
-----+")
rm2 = ("ENTER YOUR CHOICE:")
rm = (input(rm2))
if rm == "y":
    menu()
elif rm == "n":
    print("Thank You")
    exit()

def delete_record():
    fetch_query = "select * from product_table;"
    con.cursor.execute(fetch_query)
    data=[]
    for i in con.cursor:
        data.append(i)

df=pd.DataFrame(data,columns=['S.No.','Product_ID','Product_Name','Cat_ID','Brand','Price',
'Purchase_Date'])
print(df.to_markdown(index=False))
print("Enter the S.No of Product you want to delete")
s_no=int(input())
insert_query="insert into deleted_records select * from product_table where
s_no="+str(s_no)+";"
con.cursor.execute(insert_query)
con.dbc.commit()
delete_query="delete from product_table where s_no="+str(s_no)+";"
con.cursor.execute(delete_query)
con.dbc.commit()
fetch_query = "select * from product_table;"
con.cursor.execute(fetch_query)
data = []
for i in con.cursor:

```

```

        data.append(i)
    df = pd.DataFrame(data,
        columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price',
'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("***** DELETED SUCCESSFULLY
! *****")
    print("
+-----+
+-----+")
    print("
|
|")
    print("
| Type y to go Back
to exit |")
    print("
+-----+
+-----+")
    rm2 = ("ENTER YOUR CHOICE:")
    rm = (input(rm2))
    if rm == "y":
        menu()
    elif rm == "n":
        print("Thank You")
        exit()

def update_sno():
    fetch_query = "select * from product_table;"
    con.cursor.execute(fetch_query)
    data = []
    for i in con.cursor:
        data.append(i)
    df = pd.DataFrame(data, columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID',
'Brand', 'Price', 'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("Enter the Old S_no: ")
    old = input()
    print("Enter the New S_no: ")
    vlu = input()
    update_query = "update product_table set s_no = "+vlu+" where s_no="+old+";"
    con.cursor.execute(update_query)
    con.dbc.commit()
    data = []
    for i in con.cursor:
        data.append(i)
    fetch_query = "select * from product_table;"
    con.cursor.execute(fetch_query)

```

```

data = []
for i in con.cursor:
    data.append(i)
df = pd.DataFrame(data,
                    columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price',
'Purchase_Date'])
print(df.to_markdown(index=False))
print("***** UPDATED
SUCCESSFULLY ! *****")
print("
+-----+
-----+")
print("
|
|")
print("
| Type y to go Back
to exit |")
print("
+-----+
-----+")
rm2 = ("ENTER YOUR CHOICE:")
rm = (input(rm2))
if rm == "y":
    menu()
elif rm == "n":
    print("Thank You")
    exit()

def price():
    fetch_query = "select * from product_table;"
    con.cursor.execute(fetch_query)
    data = []
    for i in con.cursor:
        data.append(i)
    df = pd.DataFrame(data,
                        columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price',
'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("Enter the S_No of product: ")
    sno = input()
    print("Enter the New Price: ")
    vlu = input()
    update_query = "update product_table set price = "+vlu+" where s_no="+sno+";"
    con.cursor.execute(update_query)
    con.dbc.commit()
    data = []
    for i in con.cursor:

```

```

        data.append(i)
    fetch_query = "select * from product_table;"
    con.cursor.execute(fetch_query)
    data = []
    for i in con.cursor:
        data.append(i)
    df = pd.DataFrame(data, columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID',
'Brand', 'Price', 'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("***** UPDATED
SUCCESSFULLY ! *****")
    print("          +-----")
    print("          |          Do you Want to go to Menu Again?
|")
    print("          | Type y to go Back                      Type n
to exit |")
    print("          +-----")
    print("          +")
    rm2 = ("ENTER YOUR CHOICE:")
    rm = (input(rm2))
    if rm == "y":
        menu()
    elif rm == "n":
        print("Thank You")
        exit()

def product_id():
    s= "select * from product_table;"
    con.cursor.execute(s)
    data = []
    for i in con.cursor:
        data.append(i)
    df = pd.DataFrame(data,
        columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price',
'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("Enter the S_no of Product_ID whose value you want to Update: ")
    sno= input()
    print("Enter the new Product_ID")
    new= input()
    product_query="update product_table set product_id = " + "" +new+ "" where s_no="
+sno+";"
    con.cursor.execute(product_query)

```

```

con.dbc.commit()
con.cursor.execute("select * from product_table;")
data = []
for i in con.cursor:
    data.append(i)
df = pd.DataFrame(data, columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID',
'Brand', 'Price', 'Purchase_Date'])
print(df.to_markdown(index=False))
print("***** UPDATED
SUCCESSFULLY ! *****")
print("
+-----+
-----+")
print("
|
| Do you Want to go to Menu Again?
|")
print("
| Type y to go Back
| Type n
to exit |")
print("
+-----+
-----+")
rm2 = ("ENTER YOUR CHOICE:")
rm = (input(rm2))
if rm == "y":
    menu()
elif rm == "n":
    print("Thank You")
    exit()

def product_name():
    name_query = "select * from product_table;"
    con.cursor.execute(name_query)
    data = []
    for i in con.cursor:
        data.append(i)
    df = pd.DataFrame(data,
        columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price',
'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("Enter the S_no of Product_Name whose value you want to Update: ")
    sno = input()
    print("Enter the new Product_Name")
    new = input()
    name_query = "update product_table set product_name = " + "'" + new + "' where s_no=" +
sno + ";"
    con.cursor.execute(name_query)
    con.dbc.commit()

```



```

con.cursor.execute("select * from product_table;")
data = []
for i in con.cursor:
    data.append(i)
df = pd.DataFrame(data,
                    columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price',
'Purchase_Date'])
print(df.to_markdown(index=False))
print("***** UPDATED
SUCCESSFULLY ! *****")
print("
+-----+
-----+")
print("
|
| Do you Want to go to Menu Again?
|")
print("
| Type y to go Back
| Type n
to exit |")
print("
+-----+
-----+")
rm2 = ("ENTER YOUR CHOICE:")
rm = (input(rm2))
if rm == "y":
    menu()
elif rm == "n":
    print("Thank You")
    exit()

def cat_ID():
    cat_query = "select * from product_table;"
    con.cursor.execute(cat_query)
    data = []
    for i in con.cursor:
        data.append(i)
    df = pd.DataFrame(data,
                        columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price',
'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("Enter the S_no of cat_ID whose value you want to Update: ")
    sno = input()
    print("Enter the new cat_ID")
    new = input()
    cat_query = "update product_table set cat_ID = " + "" + new + " where s_no=" + sno + ";"
    con.cursor.execute(cat_query)
    con.dbc.commit()
    con.cursor.execute("select * from product_table;")

```

```

data = []
for i in con.cursor:
    data.append(i)
df = pd.DataFrame(data,
                   columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price',
'Purchase_Date'])
print(df.to_markdown(index=False))
print("***** UPDATED
SUCCESSFULLY ! *****")
print("          +-----+
-----+")
print("          |          Do you Want to go to Menu Again?
|")
print("          | Type y to go Back          Type n
to exit |")
print("          +-----+
-----+")
rm2 = ("ENTER YOUR CHOICE:")
rm = (input(rm2))
if rm == "y":
    menu()
elif rm == "n":
    print("Thank You")
    exit()
def insert():
    brand_query = "select * from product_table;"
    con.cursor.execute(brand_query)
    data = []
    for i in con.cursor:
        data.append(i)
    df = pd.DataFrame(data, columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID',
'Brand', 'Price', 'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("Enter the S_no of New Product")
    s=input()
    print("Enter the Product_id of New Product")
    id=input()
    print("Enter the Product_Name of New Product")
    pn=input()
    print("Enter the Cat_ID of New_Product")
    cid=input()
    print("Enter the Brand of New_Product")
    bnd=input()
    print("Enter the Price of New_Product")

```

```

    prc=input()
    print("Enter the Purchase_date of New_Product in format (yyyy-mm-dd)")
    npd=input()
    insert_query = "insert into product_table values(" +s+ ", " +id+ ", " +pn+ ", " +cid+ ", " +bnd+ ", " +prc+ ", " +npd+ ");"
    con.cursor.execute(insert_query)
    con.dbc.commit()
    con.cursor.execute("select * from product_table;")
    data=[]
    for i in con.cursor:
        data.append(i)
    df = pd.DataFrame(data, columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price', 'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("***** ITEM INSERTED SUCCESSFULLY ! *****")
    print("
+-----+
-----+")
    print("
|
|")
    print("
| Type y to go Back
to exit |")
    print("
+-----+
-----+")
    rm2 = ("ENTER YOUR CHOICE:")
    rm = (input(rm2))
    if rm == "y":
        menu()
    elif rm == "n":
        print("Thank You")
        exit()

def brand():
    brand_query = "select * from product_table;"
    con.cursor.execute(brand_query)
    data = []
    for i in con.cursor:
        data.append(i)
    df = pd.DataFrame(data, columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price', 'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("Enter the S_no of Brand whose value you want to Update: ")
    sno = input()
    print("Enter the new Brand Name")

```

```

new = input()
bra_query = "update product_table set Brand = " + "'" + new + "'" + " where s_no=" + sno + ";"
con.cursor.execute(bra_query)
con.dbc.commit()
con.cursor.execute("select * from product_table;")
data = []
for i in con.cursor:
    data.append(i)
df = pd.DataFrame(data,
                   columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand', 'Price',
                             'Purchase_Date'])
print(df.to_markdown(index=False))
print("***** UPDATED
SUCCESSFULLY ! *****")
print("
+-----")
print("
| Do you Want to go to Menu Again?
|")
print("
| Type y to go Back
| Type n
to exit |")
print("
+-----")
print("
+-----")
rm2 = ("ENTER YOUR CHOICE:")
rm = (input(rm2))
if rm == "y":
    menu()
elif rm == "n":
    print("Thank You")
    exit()

def date():
    date_query = "select * from product_table;"
    con.cursor.execute(date_query)
    data = []
    for i in con.cursor:
        data.append(i)
    df = pd.DataFrame(data, columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID', 'Brand',
                                     'Price', 'Purchase_Date'])
    print(df.to_markdown(index=False))
    print("Enter the S_no of Purchase_Date whose value you want to Update: ")
    sno = input()
    print("Enter New Day")
    dd=input()
    print("Enter New Month")

```

```

mm=input()
print("Enter New Year")
yyyy=input()
new_date=yyyy+"-"+mm+"-"+dd
date_query = "update product_table set purchase_date = " + ""+new_date+" where s_no="
+ sno + ";"
con.cursor.execute(date_query)
con.dbc.commit()
con.cursor.execute("select * from product_table;")
data = []
for i in con.cursor:
    data.append(i)
df = pd.DataFrame(data, columns=['S.No.', 'Product_ID', 'Product_Name', 'Cat_ID',
'Brand', 'Price', 'Purchase_Date'])
print(df.to_markdown(index=False))
print("***** UPDATED
SUCCESSFULLY ! *****")
print("          +-----+
-----+")
print("          |          Do you Want to go to Menu Again?
|")
print("          | Type y to go Back          Type n
to exit |")
print("          +-----+
-----+")
rm2 = ("ENTER YOUR CHOICE:")
rm = (input(rm2))
if rm=="y":
    menu()
elif rm=="n":
    print("Thank You")
    exit()

def menu():
    print("          +-----+")
    print("          | > Press a to View All Products      |")
    print("          | > Press b to See Category Wise Quantity |")
    print("          | > Press c to Delete Record          |")
    print("          | > Press d to Update S_no            |")
    print("          | > Press e to Update Product_id       |")
    print("          | > Press f to Update Product_name     |")
    print("          | > Press g to Update Cat_id          |")
    print("          | > Press h to Update Brand           |")

```

```

print("
print("
print("
print("
print("
b = ("Enter your Choice: ")
user_input = (input(b))
if user_input == "a":
    view_items()
elif user_input == "b":
    cat_quantity()
elif user_input == "c":
    delete_record()
elif user_input == "d":
    update_sno()
elif user_input == "e":
    product_id()
elif user_input == "f":
    product_name()
elif user_input == "g":
    cat_ID()
elif user_input == "h":
    brand()
elif user_input == "i":
    price()
elif user_input == "j":
    date()
elif user_input == "k":
    insert()
elif user_input == "logout":
    logout()

def logout():
    print("
    -----+")
    print("
    |
    |")
    print("
    |          yes                      no          |")
    print("
    -----+")
    -----+")
    lo2 = ("ENTER YOUR CHOICE:")
    lo = (input(lo2))
    if lo == "yes":
        print("
        |
        |")
        | > Press i to Update Price          |")
        | > Press j to Update Purchase_date    |")
        | > Press k to Insert New Product      |")
        | > Type logout to end                  |")
        +-----+")

        Please Wait, Logging Out..."

```

```

        time.sleep(2)
        print("***** LOGOUT
SUCCESSFULL ! *****")
        time.sleep(2)
        login()

class Format:
    end = '\033[0m'
    underline = '\033[4m'

print("
+-----+")
print("
| ",Format.underline+"WELCOME TO THE IT
INVENTORY MANAGEMENT SYSTEM"+ Format.end," |
")
print("
+-----+")
print("----- SELECT THE OPTION TO BEGIN
! -----")
print("
+-----+")
print("
| > Press 1 to Login      |")
print("
| > Press 2 to Register   |")
print("
+-----+")
a=("Enter your Choice: ")
user_input=int(input(a))

if user_input==1:
    val=login()
    fetch_query = "select name from registration_details where email='"+val+"';"
    con.cursor.execute(fetch_query)
    name=None
    for i in con.cursor:
        name=i[0]
    name=name.upper()
    if val!=0:
        class Format:
            end = '\033[0m'
            underline = '\033[4m'

print("
+-----")
-----+")
print("
",Format.underline + "HEY ! "+name+"
LOGIN SUCCESSFUL SELECT AN OPTION TO BEGIN" + Format.end,"
")
print("
+-----")

```

```

-----+")
print("
print("
print("
print("
print("
print("
print("
print("
print("
print("
print("
print("
print("
a = ("Enter your Choice: ")
user_input = (input(a))
if user_input == "a":
    view_items()
elif user_input == "b":
    cat_quantity()
elif user_input == "c":
    delete_record()
elif user_input == "d":
    update_sno()
elif user_input == "e":
    product_id()
elif user_input == "f":
    product_name()
elif user_input == "g":
    cat_ID()
elif user_input == "h":
    brand()
elif user_input == "i":
    price()
elif user_input == "j":
    date()
elif user_input == "k":
    insert()
elif user_input == "logout":
    logout()
elif user_input == 2:
    registration()
+-----+")
| > Press a to view all products      |")
| > Press b to See Category Wise Quantity |")
| > Press c to Delete Record          |")
| > Press d to Update S_no            |")
| > Press e to Update product_id      |")
| > Press f to Update product_name    |")
| > Press g to Update cat_id          |")
| > Press h to Update brand           |")
| > Press i to Update price           |")
| > Press j to Update purchase_date   |")
| > Press k to Insert New Product     |")
| > Type logout to end                |")
+-----+")

```


Commands Used in MySQL

Creating Database

Create database inventory;

Using Database

Use inventory;

Creating Table Product Table and Inserting Values

Create table product_table (s_no int(10), product_id varchar(20), product_name varchar(40), cat_id varchar(10), brand char(20), price int(10), purchase_date date);

Creating Table Category Table and Inserting Values

Create table category_table (cat_id varchar(30) primary key, items varchar(30));

Creating Table Deleted Records and Inserting Values

Create table product_table (s_no int(10), product_id varchar(10), product_name varchar(40), cat_id varchar(10), brand char(10), price int(10), purchase_date date);

Creating Table Registration Details and Inserting Values

Create table registration_details (id int(10) primary key auto_increment, name char(30), email varchar(50));

Conclusion and Future Work

By making this program project, we have successfully shown an example of a useful database, the kind that is used by the companies to extract information. Here we have successfully completed the task of making a database of an IT Company. And presenting it in front of the user.

The functions performed by this project program are in accordance to our assumptions for further upgradation, we can also add features such as a Graphical User Interface to this project , which can be developed in the form of a web , desktop or mobile application. This project can also be scaled to analyse global data in real time.

References

- [1] <https://www.limswiki.org/index.php/MySQL>
- [2] <https://en.wikipedia.org/wiki/PyCharm>
- [3] [https://en.wikipedia.org/wiki/Pandas_\(software\)](https://en.wikipedia.org/wiki/Pandas_(software))
- [4] <https://en.wikipedia.org/wiki/NumPy>
- [5] <https://en.wikipedia.org/wiki/Matplotlib>