

DBMS - Mini Project

ELECTRONIC SHOP AND REPAIR MANAGEMENT SYSTEM

Submitted By:

Name: Saanchita V
SRN: PES1UG20CS364
Sem: V Semester
Section: F

Short Description and Scope of the Project

This Electronic Shop Management System project is developed using Python, Streamlit and MySQL. The project has all the essential features required for the management of the electronic shop and to address repair of electronic items.

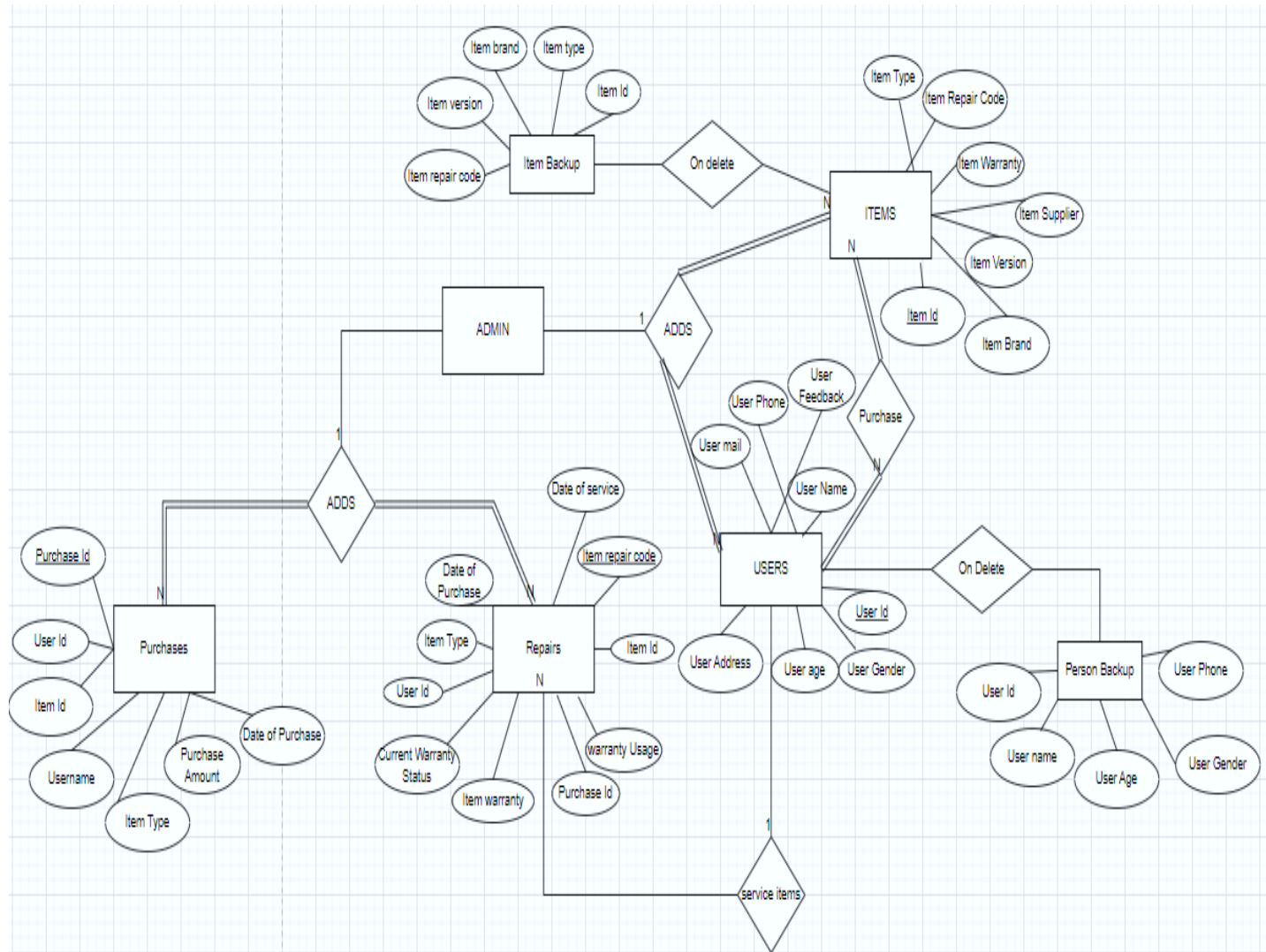
Today, there is not any household without an electronic device. Thousands of people purchase electronic items daily, thus requiring efficient management of data and provide high speed services.

This project is a smaller version of how data is stored, retrieved, updated, and viewed in an electronic store.

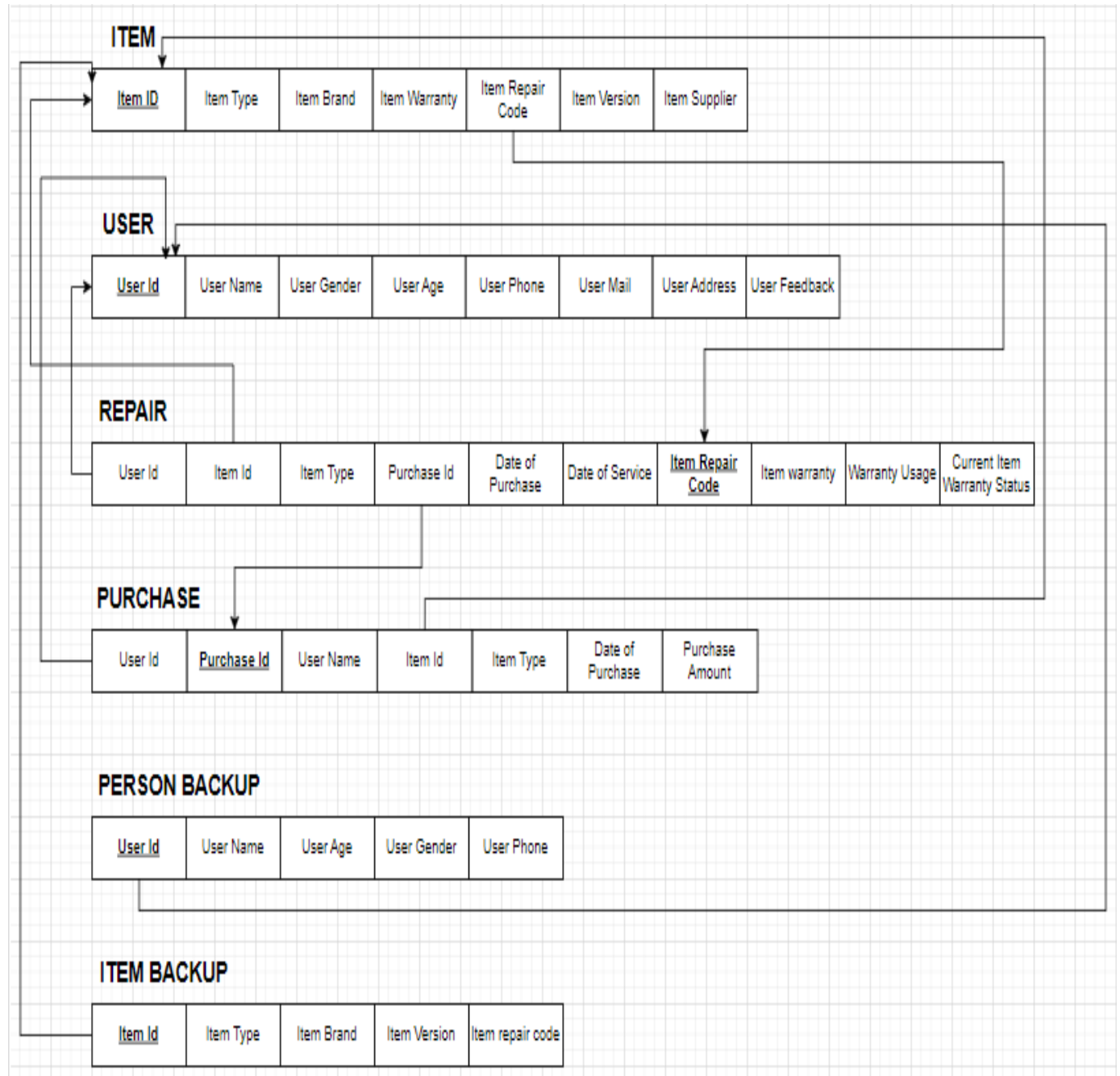
There are a total of six tables – User table to keep track of users visiting the store, Item table to add, update and remove items from inventory, Purchase table to keep track of all the purchases, Repair table to keep track of all the items that have been given for service.

The two other tables are backup tables – user backup and item back where previously existing users and items are stored. If a user or an item is deleted from the User or Item table, that user/item's information will be stored in the backup tables. Insertion into these tables is trigger induced and are not done manually.

ER Diagram



Relational Schema



DDL statements - Building the database

CREATING TABLES

1. User Table

```
mysql> CREATE TABLE user(  
-> user_id INT,  
-> user_gender TEXT,  
-> user_age INT,  
-> user_name TEXT,  
-> user_phone TEXT,  
-> user_mail TEXT,  
-> user_address TEXT,  
-> user_feedback TEXT,  
-> PRIMARY KEY(user_id)  
-> );  
Query OK, 0 rows affected (0.05 sec)
```

2. Item Table

```
mysql> CREATE TABLE item(  
-> item_id INT,  
-> item_type TEXT,  
-> item_brand TEXT,  
-> item_warranty INT,  
-> item_repair_code INT,  
-> item_version TEXT,  
-> item_supplier TEXT,  
-> PRIMARY KEY(item_id)  
-> );  
Query OK, 0 rows affected (0.03 sec)
```

3. Repair Table

```
mysql> CREATE TABLE repair(  
-> user_id INT,  
-> item_id INT,  
-> item_type TEXT,  
-> purchase_id INT,  
-> date_of_purchase DATE,  
-> date_of_service DATE,  
-> item_repair_code INT,  
-> item_warranty INT,  
-> item_warranty_status TEXT,  
-> PRIMARY KEY(item_repair_code)  
-> );  
Query OK, 0 rows affected (0.13 sec)
```

4. Purchase Table

```
mysql> CREATE TABLE purchase(  
-> user_id INT,  
-> user_name TEXT,  
-> item_id INT,  
-> item_type TEXT,  
-> purchase_id INT,  
-> date_of_purchase DATE,  
-> purchase_amt INT,  
-> PRIMARY KEY(purchase_id)  
-> );  
Query OK, 0 rows affected (0.02 sec)
```

5. Item Backup Table

```
mysql> CREATE TABLE item_backup(  
-> item_id INT,  
-> item_type TEXT,  
-> item_brand TEXT,  
-> item_version TEXT,  
-> item_repair_code INT,  
-> PRIMARY KEY(item_id)  
-> );  
Query OK, 0 rows affected (0.02 sec)
```

6. Person Backup Table

```
mysql> CREATE TABLE person_backup(  
-> user_name TEXT,  
-> user_age INT,  
-> user_id INT,  
-> user_gender TEXT,  
-> user_phone TEXT,  
-> PRIMARY KEY(user_id)  
-> );  
Query OK, 0 rows affected (0.02 sec)
```

Adding Constraints

```
mysql> ALTER TABLE purchase ADD CONSTRAINT iid FOREIGN KEY (item_id) REFERENCES item(item_id);  
ALTER TABLE repair ADD CONSTRAINT rid FOREIGN KEY (purchase_id) REFERENCES purchase(purchase_id);  
Query OK, 0 rows affected (0.18 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
Query OK, 0 rows affected (0.14 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

Populating the Database

INSERTING VALUES INTO EACH OF THE TABLES

1. User Table

```
mysql> INSERT INTO user(`user_id`,`user_gender`,`user_age`,`user_name`,`user_phone`,`user_mail`,`user_address`,`user_feedback`)VALUES(1,"Female",20,"Samanvitha","9023345711","sam@gmail.com","25,Banashankari","Good");
Query OK, 1 row affected (0.00 sec)
```

```
mysql> INSERT INTO user(`user_id`,`user_gender`,`user_age`,`user_name`,`user_phone`,`user_mail`,`user_address`,`user_feedback`)VALUES(2,"Female",20,"Isiri","9012245699","Isiri@gmail.com","29,Kumarswamy Layout","Good"),(3,"Male",19,"Sumukh","8725543979","sumukh@gmail.com","12,Jayanagar","Okay"),(4,"Male",19,"Sumedh","9035576122","sumedh@gmail.com","43,Jayanagar","Bad");
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
mysql> LOAD DATA INFILE 'D:/DBMS/PROJECT_FINAL/usertable.csv'
-> INTO TABLE user
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n'
-> ;
Query OK, 14 rows affected (0.01 sec)
Records: 14 Deleted: 0 Skipped: 0 Warnings: 0

mysql> _
```

```
mysql> SELECT * FROM user;
```

user_id	user_gender	user_age	user_name	user_phone	user_mail	user_address	user_feedback
1	Female	20	Samanvitha	9012245677	sam@gmail.com	100,4th cross,Banashankari	Good
2	Female	20	Saanchita V	9019978254	saanchita@gmail.com	100,2nd phase,JP Nagar	Good
3	Male	19	Sumukh	6542289711	sumukh@gmail.com	20,Jayanagar	Good
4	Male	30	Anirudh	765543221	anirudh@gmail.com	Kengeri,Bengaluru	Okay
5	Male	26	Sumedh	4533216778	sumedh@gmail.com	Jayanagr,Bengaluru	Bad
6	Female	25	Megha	9023345632	megha@gmail.com	Girinagar,Bengaluru	Good
7	Female	19	Adithi	7862234977	adithi@gmail.com	12,Jayanagar,Bengaluru	Okay
8	Female	19	Samhitha	8762213988	samhitha@gmail.com	29,BTM Layout,Bengaluru	Good
	Female	20	Pramithi	9017756322	pram@gmail.com	24-Jayanagar	Good
	Female	20	Druthi	9034456711	drut@gmail.com	35-Jayanagar	Good
	Male	26	Samarth	8743321655	samarth@gmail.com	20-Jayanagar	Okay
	Female	22	Anagha	7320098711	anagha@gmail.com	31-Jayanagar	Good
	Female	20	Spoorthi	9909923499	spoo@gmail.com	19-Girinagar	Bad
	Female	20	Sanmathi	9019978652	san@gmail.com	25-Girinagar	Okay
	Female	21	Supriya	9238856122	sup@gmail.com	120-RR Nagar	Okay
	Male	28	Sudith	9900345611	sudith@gmail.com	23-Jayanagar	Okay
	Male	29	Chinmay	8781234904	chin@gmail.com	18-Jayanagar	Good
	Male	24	Yashas	7022456113	yashas@gmail.com	56-Vijaynagar	Good
	Female	25	Keerthi	8214790235	kee@gmail.com	5-Banashankari	Bad
	Female	20	Kruthika	9920104566	kruthika@gmail.com	44-Whitefield	Good
	Female	19	Vedika	9927786433	dved@gmail.com	44-Whitefield	Good
22	Female	23	Prajna	8827885462	prajna@gmail.com	24-Jayanagar	Good

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

	user_id	user_gender	user_age	user_name	user_phone	user_mail	user_address	user_feedback
▶	1	Female	20	Samanvitha	9012245677	sam@gmail.com	100,4th cross,Banashankari	Good
	2	Female	20	Saanchita V	9019978254	saanchita@gmail.com	100,2nd phase,JP Nagar	Good
	3	Male	19	Sumukh	6542289711	sumukh@gmail.com	20,Jayanagar	Good
	4	Male	30	Anirudh	765543221	anirudh@gmail.com	Kengeri,Bengaluru	Okay
	5	Male	26	Sumedh	4533216778	sumedh@gmail.com	Jayanagr,Bengaluru	Bad
	6	Female	25	Megha	9023345632	megha@gmail.com	Girinagar,Bengaluru	Good
	7	Female	19	Adithi	7862234977	adithi@gmail.com	12,Jayanagar,Bengaluru	Okay
	8	Female	19	Samhitha	8762213988	samhitha@gmail.com	29,BTM Layout,Bengaluru	Good
	9	Female	20	Pramithi	9017756322	pram@gmail.com	24-Jayanagar	Good
	10	Female	20	Druthi	9034456711	drut@gmail.com	35-Jayanagar	Good
	11	Male	26	Samarth	8743321655	samarth@gmail.com	20-Jayanagar	Okay
	12	Female	22	Anagha	7320098711	anagha@gmail.com	31-Jayanagar	Good
	13	Female	20	Spoorthi	9909923499	spoo@gmail.com	19-Girinagar	Bad
	14	Female	20	Sanmathi	9019978652	san@gmail.com	25-Girinagar	Okay
	15	Female	21	Supriya	9238856122	sup@gmail.com	120-RR Nagar	Okay
	16	Male	28	Sudith	9900345611	sudith@gmail.com	23-Jayanagar	Okay
	17	Male	29	Chinmay	8781234904	chin@gmail.com	18-Jayanagar	Good
	18	Male	24	Yashas	7022456113	yashas@gmail.com	56-Vijaynagar	Good
	19	Female	25	Keerthi	8214790235	kee@gmail.com	5-Banashankari	Bad

2. Item Table

```
mysql> LOAD DATA INFILE 'D:/DBMS/PROJECT_FINAL/item.csv'
-> INTO TABLE item
-> FIELDS TERMINATED BY ','
-> LINES TERMINATED BY '\n'
-> ;
Query OK, 16 rows affected (0.03 sec)
Records: 16 Deleted: 0 Skipped: 0 Warnings: 0
```

```
mysql> SELECT * FROM item;
```

item_id	item_type	item_brand	item_warranty	item_repair_code	item_version	item_supplier
1	Laptop	Lenovo	2	123	Yoga	ABC Wholesale
2	Laptop	Asus	2	899	14	ABC Wholesale
3	Phone	Apple	2	783	12	XYZ Wholesale
4	Phone	Samsung	1	654	s6	XYZ Wholesale
5	Phone	One Plus	1	222	7	ABC Wholesale
6	Laptop	Lenovo	2	143	Thinkpad	ABC Wholesale
7	Phone	Motorola	1	688	5a	XYZ Wholesale
8	Phone	Apple	2	351	14	ABC Wholesale
9	Headphones	Boat	1	732	3.2	ABC Wholesale
10	Laptop	Dell	2	521	7	ABC Wholesale
11	Phone	Apple	2	322	12	XYZ Wholesale
12	Headphones	Bose	1	765	2	ABC Wholesale
13	Headphones	Boat	1	778	2.3	ABC Wholesale
14	Headphones	Apple	1	98	2	XYZ Wholesale
15	Laptop	Apple	2	65	2	XYZ Wholesale
16	Phone	One Plus	2	334	6	XYZ Wholesale
17	Phone	Nokia	0	113	6a	XYZ Wholesale
18	Phone	Motorola	1	145	5c2	XYZ Wholesale
19	Headphones	Boat	1	169	2	ABC Wholesale
20	Headphones	Apple	1	333	Pro	ABC Wholesale
21	Headphones	Apple	1	221	Pro	ABC Wholesale
22	Laptop	Asus	2	677	4	XYZ Wholesale
23	Laptop	HP	1	455	7	XYZ Wholesale
24	Phone	Samsung	1	956	a7	XYZ Wholesale
25	Headphones	Bose	1	237	2b	ABC Wholesale

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


item_id	item_type	item_brand	item_warranty	item_repair_code	item_version	item_supplier
8	Phone	Apple	2	351	14	ABC Wholesale
9	Headphones	Boat	1	732	3.2	ABC Wholesale
10	Laptop	Dell	2	521	7	ABC Wholesale
11	Phone	Apple	2	322	12	XYZ Wholesale
12	Headphones	Bose	1	765	2	ABC Wholesale
13	Headphones	Boat	1	778	2.3	ABC Wholesale
14	Headphones	Apple	1	98	2	XYZ Wholesale
15	Laptop	Apple	2	65	2	XYZ Wholesale
16	Phone	One Plus	2	334	6	XYZ Wholesale
17	Phone	Nokia	0	113	6a	XYZ Wholesale
18	Phone	Motorola	1	145	5c2	XYZ Wholesale
19	Headphones	Boat	1	169	2	ABC Wholesale
20	Headphones	Apple	1	333	Pro	ABC Wholesale
21	Headphones	Apple	1	221	Pro	ABC Wholesale
22	Laptop	Asus	2	677	4	XYZ Wholesale
23	Laptop	HP	1	455	7	XYZ Wholesale
24	Phone	Samsung	1	956	a7	XYZ Wholesale
25	Headphones	Bose	1	237	2b	ABC Wholesale
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Similarly, data is inserted into the repair and purchase tables.

Data is inserted into the repair table every time a user wants to get an existing item serviced.

For the repair it is not compulsory that the item must be from the item list.

However, for purchase, a user can purchase only purchase those items that are in the item table.

Join Queries

Highlight at least four join queries

Write the query in English Language, Show the equivalent SQL statement and ascreenshot of the query and the results

1. JOIN OF USER TABLE WITH REPAIR TABLE

- This is to see how many users from the users list have given items they have purchased for repair

```
mysql> SELECT user.user_id,user.user_name,user.user_phone,repair.item_id,repair.purchase_id,repair.date_of_service FROM electronic_shop_and_repair_management_364.user INNER JOIN electronic_shop_and_repair_management_364.repair ON user.user_id = repair.user_id;
```

user_id	user_name	user_phone	item_id	purchase_id	date_of_service
4	Anirudh	765543221	4	4	2021-10-20
5	Sumedh	4533216778	7	5	2022-08-02
2	Saanchita V	9019978254	2	2	2022-11-26

3 rows in set (0.00 sec)

2. JOIN OF ITEM AND PURCHASE TABLE

- This operation is done to see how many items from the list of items have been purchased by which user.
- It also gives information about the amount of the item purchased and who is the supplier of the item.

```
mysql> SELECT item.item_id,item.item_type,item.item_brand,item.item_version,item.item_supplier,purchase.purchase_id,purchase.user_name,purchase.date_of_purchase,purchase.purchase_amt FROM item INNER JOIN purchase ON item.item_id = purchase.item_id;
```

item_id	item_type	item_brand	item_version	item_supplier	purchase_id	user_name	date_of_purchase	purchase_amt
1	Laptop	Lenovo	Yoga	ABC Wholesale	1	Samanvitha	2022-11-27	30000
2	Laptop	Asus	14	ABC Wholesale	2	Saanchita	2022-10-14	60000
3	Phone	Apple	12	XYZ Wholesale	3	Sumukh	2021-09-12	50000
4	Phone	Samsung	s6	XYZ Wholesale	4	Anirudh	2020-05-04	30000
7	Phone	Motorola	5a	XYZ Wholesale	5	Sumedh	2022-06-02	20000

5 rows in set (0.01 sec)

3. LEFT JOIN ITEM AND REPAIR TABLE

- Displays the list of all items and in the 'date_of_service' column displays the dates only for those items that have been serviced

```
mysql> SELECT item.item_id,item.item_type,item.item_brand,repair.date_of_service FROM item LEFT JOIN repair ON item.item_id = repair.item_id;
```

item_id	item_type	item_brand	date_of_service
1	Laptop	Lenovo	NULL
2	Laptop	Asus	2022-11-26
3	Phone	Apple	NULL
4	Phone	Samsung	2021-10-20
5	Phone	One Plus	NULL
6	Laptop	Lenovo	NULL
7	Phone	Motorola	2022-08-02
8	Phone	Apple	NULL
9	Headphones	Boat	NULL
10	Laptop	Dell	NULL
11	Phone	Apple	NULL
12	Headphones	Bose	NULL
13	Headphones	Boat	NULL
14	Headphones	Apple	NULL
15	Laptop	Apple	NULL
16	Phone	One Plus	NULL
17	Phone	Nokia	NULL
18	Phone	Motorola	NULL
19	Headphones	Boat	NULL
20	Headphones	Apple	NULL
21	Headphones	Apple	NULL
22	Laptop	Asus	NULL
23	Laptop	HP	NULL
24	Phone	Samsung	NULL
25	Headphones	Bose	NULL

25 rows in set (0.00 sec)

4. INNER JOIN OF (INNER JOIN OF USER AND PURCHASE) AND REPAIR

- This is for the user feedback of all those users who have purchased a product.
- We perform a join with item table to retrieve information about the item purchased like the brand and version.

```
mysql> SELECT user.user_name,user.user_feedback,purchase.purchase_id,purchase.item_id,purchase.item_type,item.item_brand,item.item_version,item.item_warranty FROM ((user INNER JOIN purchase ON user.user_name = purchase.user_name) INNER JOIN item ON purchase.item_id=item.item_id);
```

user_name	user_feedback	purchase_id	item_id	item_type	item_brand	item_version	item_warranty
Samanvitha	Good	1	1	Laptop	Lenovo	Yoga	2
Sumukh	Good	3	3	Phone	Apple	12	2
Anirudh	Okay	4	4	Phone	Samsung	s6	1
Sumedh	Bad	5	7	Phone	Motorola	5a	1

Aggregate Functions

Highlight at least 4 Aggregate function queries

Write the query in English Language, Show the equivalent SQL statement and a screenshot of the query and the results

1. MIN

- To find least amount sold

```
mysql> SELECT * FROM purchase WHERE purchase_amt=(SELECT MIN(purchase_amt) FROM purchase);
+-----+-----+-----+-----+-----+-----+-----+
| user_id | user_name | item_id | item_type | purchase_id | date_of_purchase | purchase_amt |
+-----+-----+-----+-----+-----+-----+-----+
|      5 | Sumedh   |      7 | Phone    |      5 | 2022-06-02      |      20000 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

2. MAX

- To find maximum amount sold

```
mysql> SELECT * FROM purchase WHERE purchase_amt=(SELECT MAX(purchase_amt) FROM purchase);
+-----+-----+-----+-----+-----+-----+-----+
| user_id | user_name | item_id | item_type | purchase_id | date_of_purchase | purchase_amt |
+-----+-----+-----+-----+-----+-----+-----+
|      2 | Saanchita |      2 | Laptop   |      2 | 2022-10-14      |      60000 |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

3. GROUP BY

- Total amount sold for each item type

```
mysql> SELECT item_type,SUM(purchase_amt) FROM electronic_shop_and_repair_management_364.purchase GROUP BY item_type;
+-----+-----+
| item_type | SUM(purchase_amt) |
+-----+-----+
| Laptop   |          90000 |
| Phone    |         100000 |
+-----+-----+
2 rows in set (0.01 sec)
```

4. SUM

- Total sales

```
mysql> SELECT SUM(purchase_amt) AS Total_Sales FROM electronic_shop_and_repair_management_364.purchase;
+-----+
| Total_Sales |
+-----+
|         190000 |
+-----+
1 row in set (0.00 sec)
```

5. AVG

- Average Sales

```
mysql> SELECT AVG(purchase_amt) AS Average_Sales FROM electronic_shop_and_repair_management_364.purchase;
+-----+
| Average_Sales |
+-----+
|      38000.0000 |
+-----+
1 row in set (0.00 sec)
```

Set Operations

Showcase at least 4 Set Operations queries

Write the query in English Language, Show the equivalent SQL statement and also a screenshot of the query and the results

1. INTERSECTION OF REPAIR AND PURCHASE TABLES

- To find users who have purchased an item from the shop and have returned to the same store for servicing of item

```
mysql> SELECT user_id,item_id,item_type,purchase_id,date_of_purchase FROM electronic_shop_and_repair_management_364.repair INTERSECT SELECT user_id,item_id,item_type,purchase_id,date_of_purchase FROM electronic_shop_and_repair_management_364.purchase;
```

user_id	item_id	item_type	purchase_id	date_of_purchase
4	4	Phone	4	2020-05-04
5	7	Phone	5	2022-06-02
2	2	Laptop	2	2022-10-14

3 rows in set (0.00 sec)

2. MINUS OF PURCHASE AND REPAIR TABLES

- To find users who have only purchased an item and have not come for servicing an item.

```
mysql> SELECT user_id,item_id,item_type,purchase_id,date_of_purchase FROM electronic_shop_and_repair_management_364.purchase EXCEPT SELECT user_id,item_id,item_type,purchase_id,date_of_purchase FROM electronic_shop_and_repair_management_364.repair;
```

user_id	item_id	item_type	purchase_id	date_of_purchase
1	1	Laptop	1	2022-11-27
3	3	Phone	3	2021-09-12

2 rows in set (0.00 sec)

3. MINUS OF USER AND PURCHASE TABLES

- To find users in the users list who have not purchased any item yet

```
mysql> SELECT user_id,user_name FROM electronic_shop_and_repair_management_364.user EXCEPT  
SELECT user_id,user_name FROM electronic_shop_and_repair_management_364.purchase;
```

user_id	user_name
6	Megha
7	Adithi
8	Samhitha
9	Pramithi
10	Druthi
11	Samarth
12	Anagha
13	Spoorthi
14	Sanmathi
15	Supriya
16	Sudith
17	Chinmay
18	Yashas
19	Keerthi
20	Kruthika
21	Vedika
22	Prajna

17 rows in set (0.00 sec)


Functions and Procedures

Create a Function and Procedure. State the objective of the function / Procedure. Run and display the results.

1. STORED PROCEDURE

- The stored procedure takes in the user feedback and performs a join operation on three tables and uses a cursor to iterate through each row of the resulting table to fetch only those rows with the feedback given as the input and returns all those rows where the feedback is equal to the input to the stored procedure.

```
211 • CREATE PROCEDURE userfeedback(IN fdbc TEXT)
212 BEGIN
213 DECLARE feedback TEXT;
214 DECLARE username TEXT;
215 DECLARE mail TEXT;
216 DECLARE pid INT;
217 DECLARE item_type TEXT;
218 DECLARE piid INT;
219 DECLARE riid INT;
220 DECLARE irc INT;
221 DECLARE finish INT DEFAULT 0;
222 DECLARE cur_1 CURSOR FOR SELECT user.user_feedback,user.user_name,user.user_mail,purchase.purchase_id,
223 purchase.item_type,purchase.item_id,repair.item_repair_code,repair.item_id
224 from ((electronic_shop_and_repair_management_364.user INNER JOIN repair ON user.user_id=repair.user_id)
225 INNER JOIN purchase ON user.user_id=purchase.user_id)
226 WHERE user_feedback=fdbc;
227 DECLARE CONTINUE HANDLER FOR NOT FOUND SET finish=1;
228 OPEN cur_1;
229 get_user:LOOP
230 FETCH cur_1 INTO feedback,username,mail,pid,item_type,piid,irc,riid;
231 SELECT feedback, username, mail, pid, item_type, piid, irc, riid;
232 IF finish=1 THEN
233 LEAVE get_user;
234 END IF;
235 END LOOP get_user;
236 END$$
```

 Call stored procedure electronic_shop_and_repair_management_364.userfeedback

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

fdbc [IN] TEXT

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [A](#)

	feedback	username	mail	pid	item_type	piid	irc	riid
▶	Good	Samanvitha	sam@gmail.com	1	Laptop	1	783	3

Result 1 x Result 2 Result 3 Result 4

queries_364* userfeedback x

Limit to 1000 rows

```

1
2
3
4
5 • call electronic_shop_and_repair_management_364.userfeedback('Good');

```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [A](#)

	feedback	username	mail	pid	item_type	piid	irc	riid
▶	Good	Saanchita	saanchita@gmail.com	2	Laptop	2	899	2

Call stored procedure electronic_shop_and_repair_management_364.userfeedback — □ ×

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

fdbc [IN] TEXT

Execute Cancel

queries_364* userfeedback x

Limit to 1000 rows

```

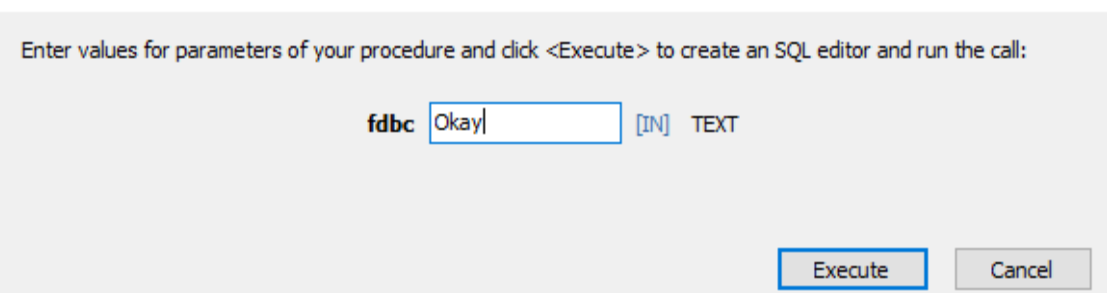
1 • call electronic_shop_and_repair_management_364.userfeedback('Bad');
2

```

<

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [A](#)

	feedback	username	mail	pid	item_type	piid	irc	riid
▶	Bad	Sumedh	sumedh@gmail.com	5	Phone	7	688	7



```
1 • call electronic_shop_and_repair_management_364.userfeedback('Okay');
2
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
feedback	username	mail	pid
item_type	piid	irc	riid
▶ Okay	Anirudh	anirudh@gmail.com	4
Phone	4	654	4

2. FUNCTION

a. FUNCTION 1

The function takes the previous purchase id and the current purchase id and calculates the discount on the purchase amount based on the amount spent by the user previously

```
DELIMITER $$  
• CREATE FUNCTION Discount_on_Purchase (pid_old INT,pid_new INT)  
  RETURNS TEXT DETERMINISTIC  
  BEGIN  
    DECLARE message TEXT;  
    DECLARE amt INT;  
    DECLARE amt2 INT;  
    SET message = 'Discount has been Applied';  
    SELECT purchase_amt INTO amt FROM purchase WHERE purchase_id = pid_old;  
    SELECT purchase_amt INTO amt2 FROM purchase WHERE purchase_id = pid_new;  
    IF amt<=30000 THEN  
      SET amt2 = amt2 - amt2*0.1;  
    ELSEIF (amt > 30000 and amt < 60000) THEN  
      SET amt2 = amt2 - amt2*0.2;  
    ELSEIF (amt >= 60000) THEN  
      SET amt2 = amt2 - amt2*0.3;  
    END IF;  
    UPDATE purchase SET purchase_amt=amt2 WHERE purchase_id = pid_new;  
    RETURN(message);  
  END$$  
  
DELIMITER ;
```


Initial Purchase Table

	user_id	user_name	item_id	item_type	purchase_id	date_of_purchase	purchase_amt
▶	1	Samanvitha	1	Laptop	1	2022-04-12	40000
	2	Saandhita	2	Laptop	2	2022-10-14	60000
	3	Sumukh	3	Phone	3	2021-09-12	50000
	4	Anirudh	4	Phone	4	2020-05-04	30000
	5	Sumedh	7	Phone	5	2022-06-02	18000
	7	Adithi	15	Laptop	6	2021-08-06	80000
	15	Supriya	21	Headphones	7	2019-03-04	7000
	12	Anagha	20	Headphones	8	2022-11-07	5000
	9	Pramithi	22	Laptop	9	2022-01-03	45000
	11	Samarth	16	Phone	10	2020-03-02	30000
	22	Prajna	23	Laptop	11	2019-07-04	60000
	17	Chinmay	18	Phone	12	2020-01-18	42000
	18	Yashas	17	Phone	13	2022-04-12	8500
	20	Kruthika	14	Headphones	14	2020-05-06	2000
	16	Suditha	12	Headphones	15	2021-09-11	5000
	1	Samanvitha	2	Laptop	16	2022-04-12	48000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Inserting a user who has already purchased an item previously

	1	Samanvitha	2	Laptop	16	2022-04-12	48000
...	1	Samanvitha	21	Headphones	17	2022-04-12	7000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Executing the function

 Call stored function electronic_shop_and_repair_manage... — □ ×



Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

pid_old INT
pid_new INT

Execute

Cancel

1 • `select electronic_shop_and_repair_management_364.Discount_on_Purchase(16, 17);`
2

<	
Result Grid	Filter Rows: <input type="text"/> Export:  Wrap Cell Content: 
	electronic_shop_and_repair_management_364.Discount_on_Purchase(16, 17)
▶	Discount has been Applied

Purchase Amount has been changed to the discounted amount

1	Samanvitha	2	Laptop	16	2022-04-12	48000
1	Samanvitha	21	Headphones	17	2022-04-12	5600
NULL	NULL	NULL	NULL	NULL	NULL	NULL

b. FUNCTION 2

The function takes in the purchase id (pid) as input and tells the user how much discount they will receive on their next purchase based on the amount spent on current purchase

```
242  -- FUNCTION
243  -- -----
244
245  DELIMITER $$
246 •  USE `electronic_shop_and_repair_management_364`$$
247 •  CREATE FUNCTION Discount_on_Purchase (pid INT)
248  RETURNS TEXT DETERMINISTIC
249  BEGIN
250  DECLARE message TEXT;
251  DECLARE amt INT;
252  SELECT purchase_amt INTO amt FROM purchase WHERE purchase_id=pid;
253  IF amt <= 30000 THEN
254      SET message = 'You get 10% discount on your next purchase';
255  elseif (amt > 30000 and amt < 60000) THEN
256      SET message = 'You get 20% discount on your next purchase';
257  elseif (amt >= 60000) THEN
258      SET message = 'You get 30% discount on your next purchase';
259
260  END IF;
261
262  RETURN(message);
263  END$$
```

Call stored function electronic_shop_and_repair_management_364.Discount_on_Purchase

Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

pid INT

Execute Cancel

queries_364 Discount_on_Purchase x

Limit to 1000 rows

```
1 • select electronic_shop_and_repair_management_364.Discount_on_Purchase(2);
```

2

<

Result Grid Filter Rows: Export: Wrap Cell Content:

electronic_shop_and_repair_management_364.Discount_on_Purchase(2)
▶ You get 30% discount on your next purchase

Call stored function electronic_shop_and_repair_management_364.Discount_on_Purchase — □ ×

Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

pid 4 INT

Execute Cancel

queries_364 Discount_on_Purchase x

Limit to 1000 rows

```
1 • select electronic_shop_and_repair_management_364.Discount_on_Purchase(4);
```

2

<

Result Grid Filter Rows: Export: Wrap Cell Content:

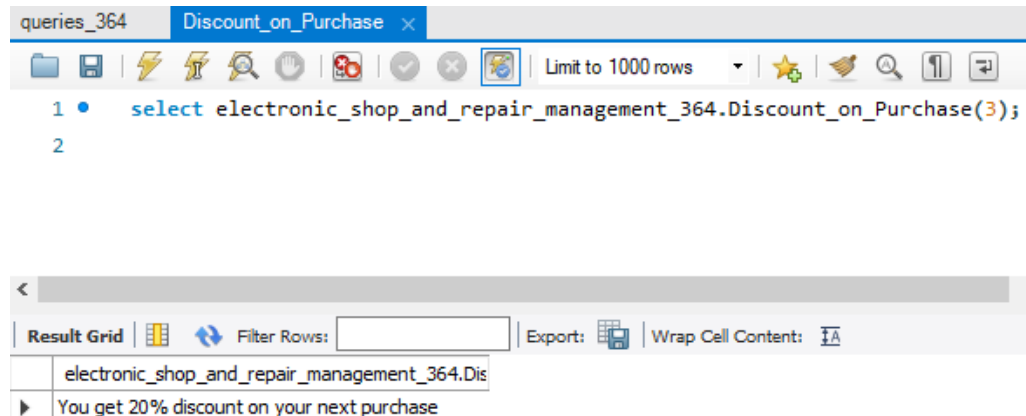
electronic_shop_and_repair_management_364.Discount_on_Purchase(4)
▶ You get 10% discount on your next purchase

Call stored function electronic_shop_and_repair_management_364.Discount_on_Purchase — □ ×

Enter values for parameters of your function and click <Execute> to create an SQL editor and run the call:

pid 3 INT

Execute Cancel



Triggers and Cursors

Create a Trigger and a Cursor. State the objective. Run and display the results.

1. Before Insert and Before Update Trigger which gives an error message when the user's age is less than 18 years.
 - Before insert Trigger – checks if user's age is greater than eighteen

-- 1. Before Insert Trigger

```
delimiter //
CREATE TRIGGER add_user BEFORE INSERT
ON user
FOR EACH ROW
IF NEW.user_age < 18 THEN
  SIGNAL SQLSTATE '50001' SET MESSAGE_TEXT = "User must be older than 18 ";
END IF; //
delimiter;
```

```
mysql> INSERT INTO `electronic_shop_and_repair_management_364`.`user`(`user_id`,`user_gender`,`user_age`,`user_name`,`user_phone`,`user_mail`,`user_address`,`user_feedback`)VALUES(23,'Female',12,'purvi','877654329','purvi@gmail.com','banashankari','good');
ERROR 1644 (50001): User must be older than 18
mysql>
```

- Before update Trigger – checks if user's age is greater than eighteen

```
-- 2. Before Update Trigger
delimiter //
CREATE TRIGGER update_user BEFORE UPDATE
ON user
FOR EACH ROW
> IF NEW.user_age < 18 THEN
  SIGNAL SQLSTATE '50002' SET MESSAGE_TEXT = "User must be older than 18 ";
~ END IF; //
delimiter;
```

```
mysql> UPDATE `electronic_shop_and_repair_management_364`.`user` SET `user_id` = 22, `user_gender` = 'Female'
, `user_age` = 16, `user_name` = 'Prajna', `user_phone` = '9087732115', `user_mail` = 'praj@gmail.com', `user_a
ddress` = 'jayanagar', `user_feedback` = 'Good' WHERE `user_id` = 22;
ERROR 1644 (50002): User must be older than 18
mysql>
```

2. Before delete Trigger for item and user table. Whenever a record is deleted from either of the two tables, the deleted record is added to the backup tables item backup and person backup.

- User table

```
-- 3. Before Delete Trigger - User Table
delimiter //
CREATE TRIGGER store_user_bd BEFORE DELETE
ON user
FOR EACH ROW
INSERT INTO person_backup (user_name,user_age,user_id,user_gender,user_phone)
VALUES (OLD.user_name, OLD.user_age, OLD.user_id, OLD.user_gender, OLD.user_phone);//
```

```
mysql> select * from person_backup;
+-----+-----+-----+-----+-----+
| user_name | user_age | user_id | user_gender | user_phone |
+-----+-----+-----+-----+-----+
| Isiri | 21 | 7 | Female | 7033237649 |
| Dhruti | 20 | 7 | Female | 9876543211 |
| Anagha | 21 | 8 | Female | 9016653411 |
+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)
```

- Item Table

```
-- 4. Before Delete Trigger - Item Table
delimiter //
CREATE TRIGGER store_item_bd BEFORE DELETE
ON item
FOR EACH ROW
INSERT INTO item_backup (item_type,item_id,item_brand,item_version,item_repair_code)
VALUES (OLD.item_type, OLD.item_id, OLD.item_brand, OLD.item_version, OLD.item_repair_code);//
delimiter;
```

```
mysql> select * from item_backup;
```

item_type	item_id	item_brand	item_version	item_repair_code
Laptop	8	Dell	6	78
Laptop	8	HP	S360	831

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

CURSOR

```
CREATE PROCEDURE set_warranty()
BEGIN
    DECLARE res1 INT;
    DECLARE res INT;
    DECLARE irc INT;
    DECLARE iw INT;
    DECLARE dop DATE;
    DECLARE dos DATE;
    DECLARE finish INT DEFAULT 0;
    DECLARE cur2 CURSOR FOR SELECT item_repair_code, item_warranty, date_of_purchase,date_of_service FROM repair;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET finish=1;
    OPEN cur2;
    a:LOOP
        FETCH cur2 INTO irc,iw,dop,dos;
        SET res = DATEDIFF(dos,dop) DIV 365;
        SET res1 = DATEDIFF(NOW(),dop) DIV 365;
        IF finish!=1 THEN
            IF res<iw THEN
                UPDATE repair SET repair.warranty_usage = 'Applied' WHERE item_repair_code = irc;
            ELSEIF res>=iw THEN
                UPDATE repair SET repair.warranty_usage = 'Not Applied' WHERE item_repair_code = irc;
            END IF;
            IF res1<iw THEN
                UPDATE repair SET repair.current_item_warranty_status = 'Warranty is still Valid' WHERE item_repair_code = irc;
            ELSEIF res1>=iw THEN
                UPDATE repair SET repair.current_item_warranty_status = 'Warranty has Expired' WHERE item_repair_code = irc;
            END IF;
        END IF;
        IF finish=1 THEN
            LEAVE a;
        END IF;
    END LOOP a;
END $$
```


- The cursor is implemented inside a stored procedure. The cursor updates if the warranty has been used on the date of service of an item and also updates the warranty status as of current date.

	user_id	item_id	item_type	purchase_id	date_of_purchase	date_of_service	item_repair_code	item_warranty	warranty_usage	current_item_warranty_status
▶	4	4	Phone	4	2020-05-04	2021-10-20	654	1	Not Applied	Warranty has Expired
	5	7	Phone	5	2022-06-02	2022-08-02	688	1	Applied	Warranty is still Valid
	1	3	Phone	3	2020-09-07	2022-09-05	783	2	Applied	Warranty has Expired
	2	2	Laptop	2	2022-10-14	2022-11-26	899	2	Applied	Warranty is still Valid
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Adding a new repair item

	user_id	item_id	item_type	purchase_id	date_of_purchase	date_of_service	item_repair_code	item_warranty	warranty_usage	current_item_warranty_status
	15	21	Headphones	7	2019-03-04	2020-01-03	221	1	NULL	NULL
	22	23	Laptop	11	2019-07-04	2020-03-01	455	1	Applied	Warranty has Expired

Now we call the function set warranty () to automatically update warranty status

	user_id	item_id	item_type	purchase_id	date_of_purchase	date_of_service	item_repair_code	item_warranty	warranty_usage	current_item_warranty_status
▶	15	21	Headphones	7	2019-03-04	2020-01-03	221	1	Applied	Warranty has Expired
	22	23	Laptop	11	2019-07-04	2020-03-01	455	1	Applied	Warranty has Expired
	4	4	Phone	4	2020-05-04	2021-10-20	654	1	Not Applied	Warranty has Expired
	5	7	Phone	5	2022-06-02	2022-08-02	688	1	Applied	Warranty is still Valid
	1	3	Phone	3	2020-09-07	2022-09-05	783	2	Applied	Warranty has Expired
	2	2	Laptop	2	2022-10-14	2022-11-26	899	2	Applied	Warranty is still Valid
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Warranty Status has been updated for item with repair code 221

Developing a Frontend

The frontend should support

1. Addition, Modification and Deletion of records from any chosen table

×

User and Item Menu

- Add user
- Read user
- Update user
- Remove user
- Add item
- Read item
- Update item
- Remove item
- View Charts

⌘

Electronic shop and repair management 364

Enter user Details:

User Id:

Phone:

Gender:

Mail:

Age:

Address:

Name:

Feedback:

Add user

INSERT

Electronic shop and repair management 364

Enter user Details:

User Id:

Phone:

Gender:

Mail:

Age:

Address:

Name:

Feedback:

Add user

Electronic shop and repair management 364

Enter user Details:

User Id:	Phone:
<input type="text" value="24"/>	<input type="text" value="9044576221"/>
Gender:	Mail:
<input type="text" value="Male"/>	<input type="text" value="abhinav@gmail.com"/>
Age:	Address:
<input type="text" value="20"/>	<input type="text" value="12,Vijayanagar"/>
Name:	Feedback:
<input type="text" value="Abhinav"/>	<input type="text" value="Good"/>
<input type="button" value="Add user"/>	

Successfully added user: Abhinav

READ

Electronic shop and repair management 364

Read user details

View all users

	user_id	user_gender	user_age	user_name	user_phone	user_mail	user_address
0	1	Female	20	Samanvitha	9012245677	sam@gmail.com	100,4th cross,I
1	2	Female	20	Saanchita	9019978254	saanchita@gmail.com	100,2nd phase
2	3	Male	19	Sumukh	6542289711	sumukh@gmail.com	20,Jayanagar
3	4	Male	30	Anirudh	765543221	anirudh@gmail.com	Kengeri,Benga
4	5	Male	26	Sumedh	4533216778	sumedh@gmail.com	Jayanagr,Beng
5	6	Female	25	Megha	9023345632	megha@gmail.com	Girinagar,Beng
6	7	Female	19	Adithi	7862234977	adithi@gmail.com	12,Jayanagar,I
7	8	Female	19	Samhitha	8762213988	samhitha@gmail.com	29,BTM Layout
8	9	Female	20	Pramithi	9017756322	pram@gmail.com	24-Jayanagar
9	10	Female	20	Druthi	9034456711	drut@gmail.com	35-Jayanagar

UPDATE

×

User and Item Menu

- ☐ Add user
- ☐ Read user
- ☒ Update user
- ☐ Remove user
- ☐ Add item
- ☐ Read item
- ☐ Update item
- ☐ Remove item
- ☐ View Charts

Electronic shop and repair management 364

Enter user Details:

Current users

▼

user to Edit

Abhinav

▼

User Id:

24

Phone:

9044576221

Gender:

Male

Mail:

abhinav@gmail.com

×

User and Item Menu

- ☐ Add user
- ☐ Read user
- ☒ Update user
- ☐ Remove user
- ☐ Add item
- ☐ Read item
- ☐ Update item
- ☐ Remove item
- ☐ View Charts

Current users

▼

user to Edit

Abhinav

▼

Chinnay

Yashas

Keerthi

Kruthika

Vedika

Prajna

Ananya

Abhinav

Abhinav

Good

Update user

Updated data

▼

user to Edit

Abhinav

User Id:

24

Phone:

9044576221

Gender:

Male

Mail:

abhinava@gmail.com

Age:

20

Address:

10,Vijayanagar

Name:

Abhinav A

Feedback:

Good

Update user

Successfully updated Abhinav

Updated data

Updated data

	er_id	user_gender	user_age	user_name	user_phone	user_mail	user_address
14	15	Female	21	Supriya	9238856122	sup@gmail.com	120-RR Nagar
15	16	Male	28	Sudith	9900345611	sudith@gmail.com	23-Jayanagar
16	17	Male	29	Chinmay	8781234904	chin@gmail.com	18-Jayanagar
17	18	Male	24	Yashas	7022456113	yashas@gmail.com	56-Vijaynagar
18	19	Female	25	Keerthi	8214790235	kee@gmail.com	5-Banashankari
19	20	Female	20	Kruthika	9920104566	kruthika@gmail.com	44-Whitefield
20	21	Female	19	Vedika	9927786433	dved@gmail.com	44-Whitefield
21	22	Female	23	Prajna	8827885462	prajna@gmail.com	24-Jayanagar
22	23	Female	20	Ananya	9023365421	ananya@gmail.com	121,Banashankari
23	24	Male	20	Abhinav A	9044576221	abhinava@gmail.com	10,Vijayanagar

DELETE

×

User and Item Menu

- ☐ Add user
- ☐ Read user
- ☐ Update user
- ☒ Remove user
- ☐ Add item
- ☐ Read item
- ☐ Update item
- ☐ Remove item
- ☐ View Charts

Electronic shop and repair management 364

🔗 Delete created users

Current data ▾

Deleting record

Samanvitha ▾

Do you want to delete ::Samanvitha

Delete user record

Updated data ▾

Delete created users

Current data ▾

Deleting record

Abhinav A ▾

Do you want to delete ::Abhinav A

Delete user record

Updated data ▾

Delete created users

Current data



Deleting record

Abhinav A



Do you want to delete ::Abhinav A

Delete user record

user record has been deleted successfully

Updated data



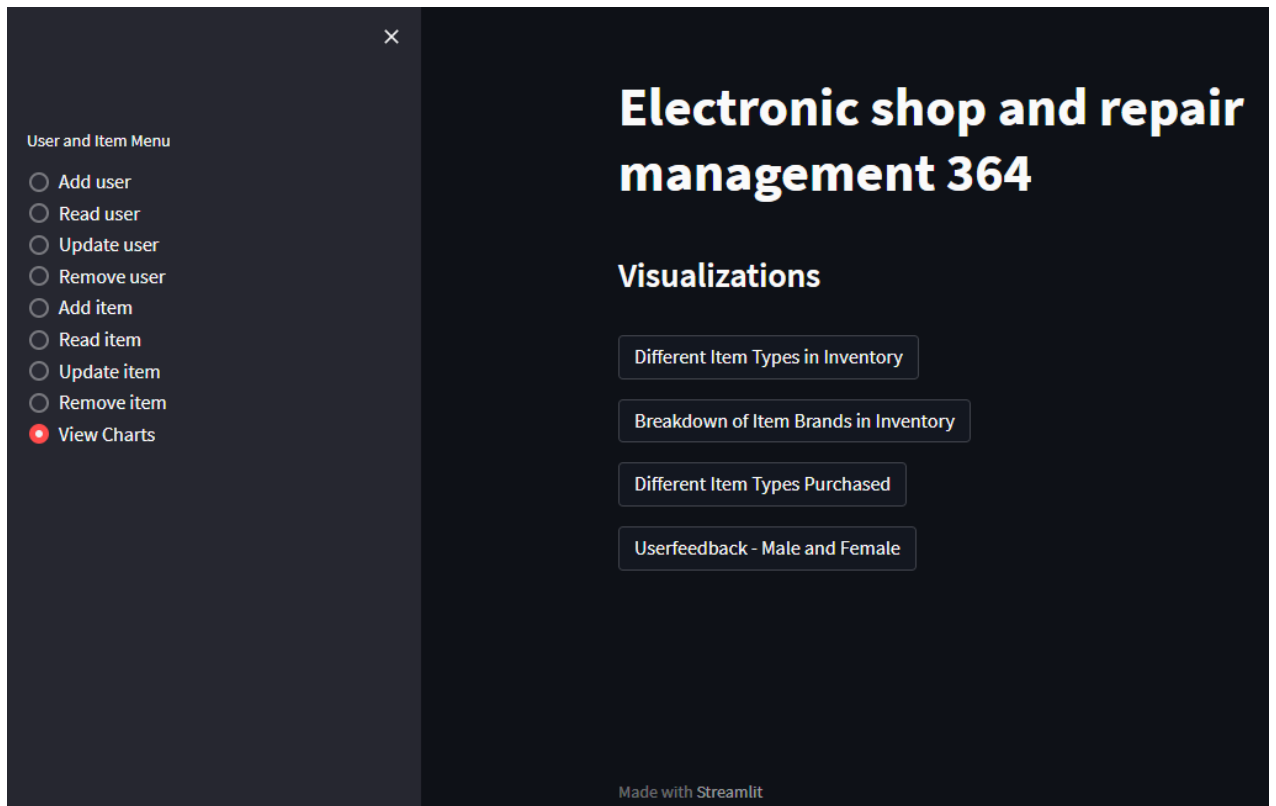
user record has been deleted successfully

Updated data

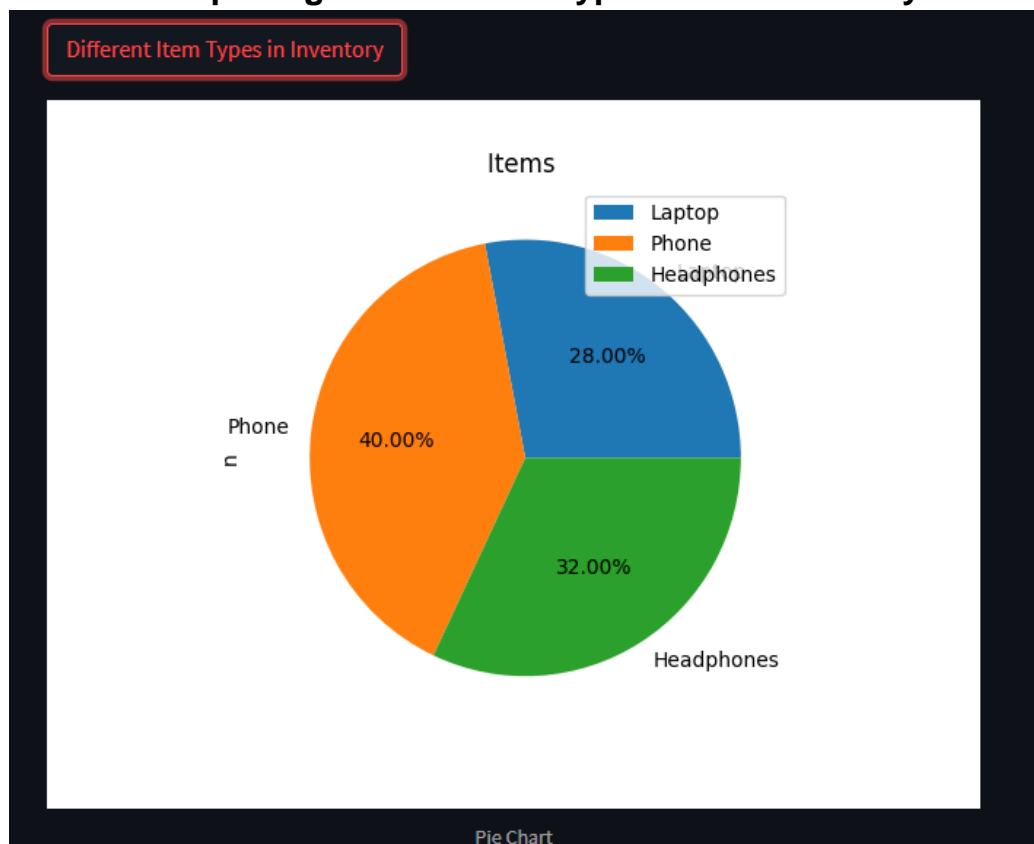


	user_id	user_gender	user_age	user_name	user_phone	user_mail	user_address
13	14	Female	20	Sanmathi	9019978652	san@gmail.com	25-Girinagar
14	15	Female	21	Supriya	9238856122	sup@gmail.com	120-RR Nagar
15	16	Male	28	Sudith	9900345611	sudith@gmail.com	23-Jayanagar
16	17	Male	29	Chinmay	8781234904	chin@gmail.com	18-Jayanagar
17	18	Male	24	Yashas	7022456113	yashas@gmail.com	56-Vijaynagar
18	19	Female	25	Keerthi	8214790235	kee@gmail.com	5-Banashankar
19	20	Female	20	Kruthika	9920104566	kruthika@gmail.com	44-Whitefield
20	21	Female	19	Vedika	9927786433	dved@gmail.com	44-Whitefield
21	22	Female	23	Prajna	8827885462	prajna@gmail.com	24-Jayanagar
22	23	Female	20	Ananya	9023365421	ananya@gmail.com	121,Banashankar

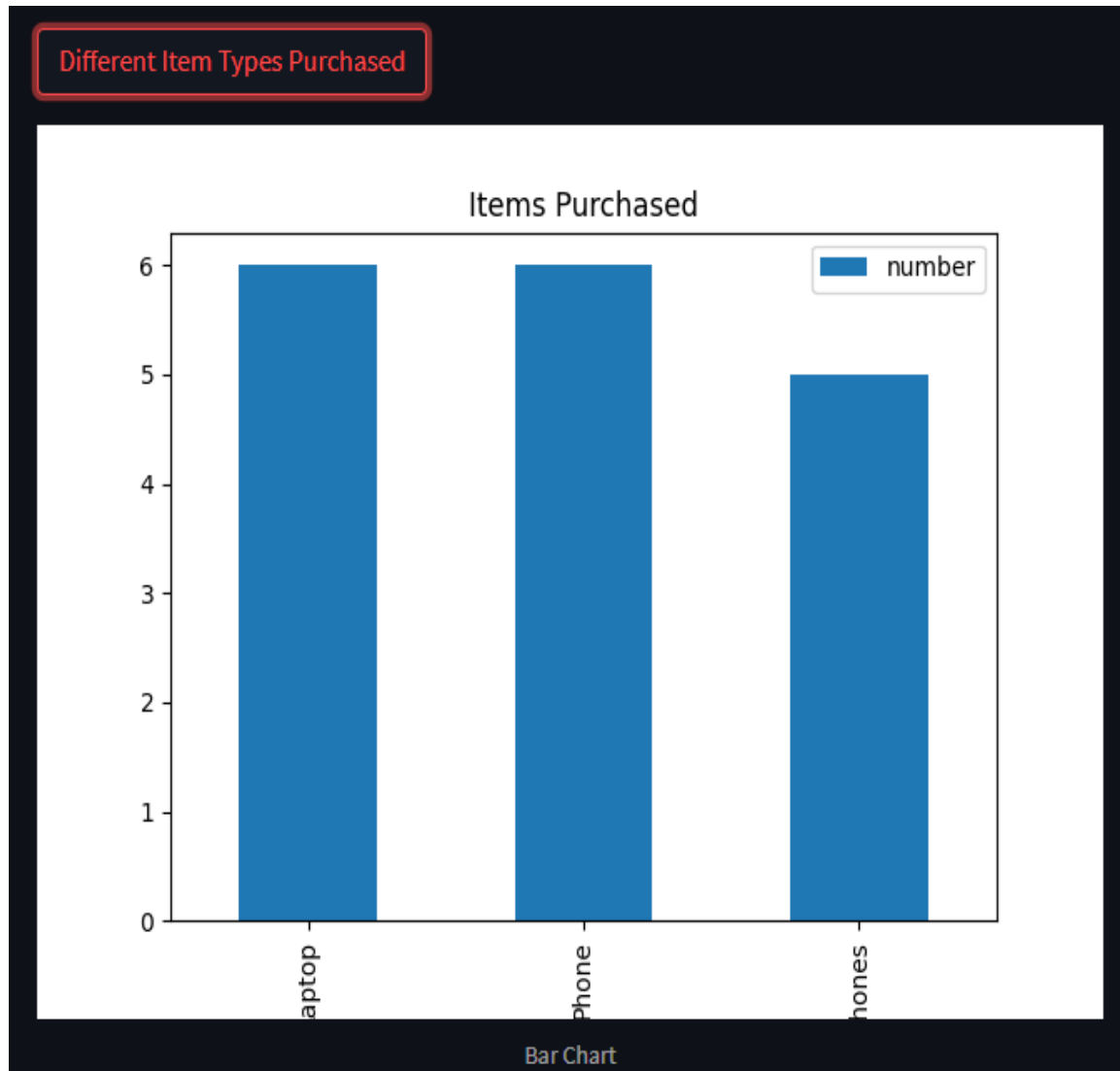
CHARTS - Real Time Visualizations



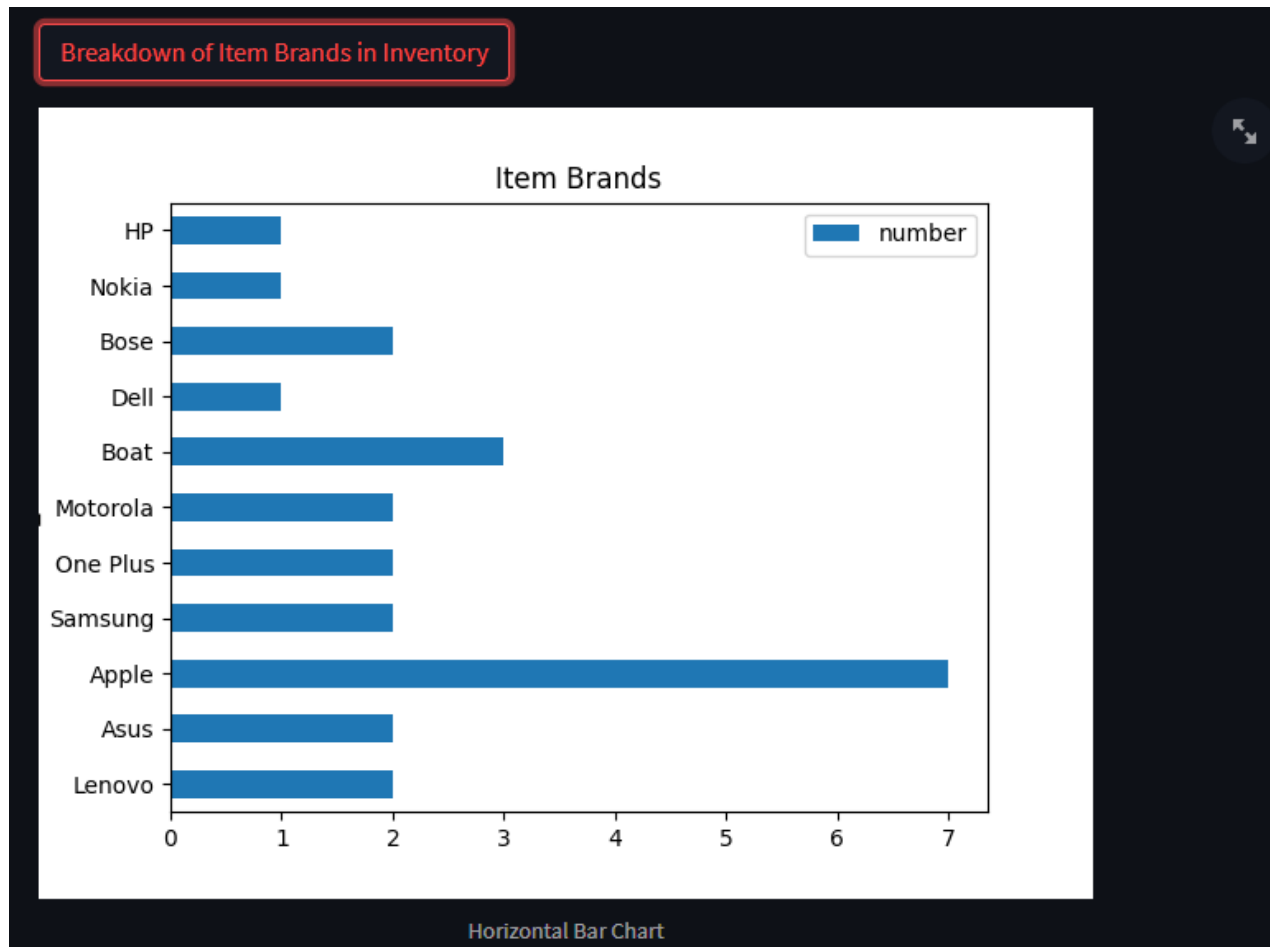
1. Pie Chart depicting different item types in the inventory



2. Bar Chart depicting the number of each type of item purchased



3. Horizontal Bar Chart depicting the number of items in each brand



4. Horizontal Bar Chart depicting the user feedback based on gender

