UNIVERSITI TEKNOLOGI MARA

ICT501 - DATABASE MANAGEMENT SYSTEM
FINAL PROJECT REPORT
(ENTERPRISE MANAGEMENT DATABASE SYSTEM)

NAME	MATRIC NO
MUHAMMAD AL-AMIN BIN MOHD ZAINI	2021100537
MUHAMAD FAIDI AKIF BIN MD ALI	2021196337
MUHAMMAD SYAZWAN FIKRI BIN SAHRAN	2021100233
ZUBLI QUZAINI BIN ZUBLI	2021119947

PREPARED FOR

MADAM NORISAN BINTI ABDUL KARIM

Table of Contents

1.0 Company Background

Figure 1.0 PROVAPE SEMABOK Enterprise Logo

Figure 1.1 Mr Mustaqim Bin Anuar

We have decided to come out with an Enterprise Management Database System for Provape Semabok Enterprise which will help to improve and increase the effectiveness as well as efficiency of the business processes in the business. The founder of this company is Mr Mustaqim Bin Anuar, 23 years old who lives in Serkam, Melaka as shown in Figure 2.0. This is a new company that was established on 1 February 2021. It is located at No 10, Jalan PS3 Plaza Semabok, Melaka. Provape Semabok Enterprise sells e-cigarettes related products such as mods, coils, batteries, and e-liquids. Their operating hours is from 12:00 pm until 11:30 pm everyday including public holidays.

The company restocks their products weekly. These products include 30 types of mod (Caliburn, Vaporshark, Alladin, Trinity and Smoant), 15 types of coils, 5 types of battery and 120 types of e-liquid. They also came up with a way to tie a customer to the shop which is the card stamp system. They can redeem any device at the store once they receive all 10 stamps. Each stamp is received when purchases is above RM 50.00.

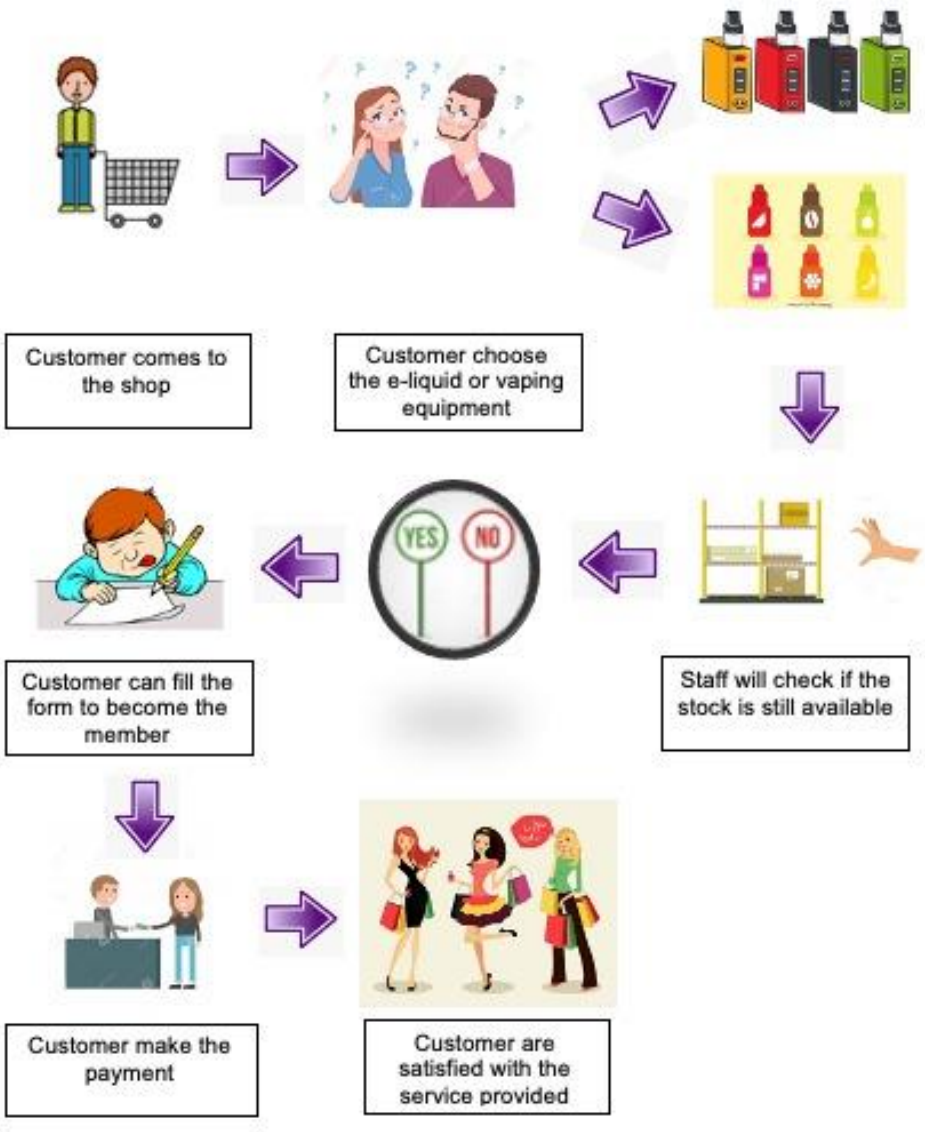
Vision

To become a leading, dynamic, and competitive source organization through a strong and solid team alliance. In effort to set the higher bar to compete with other stores the working environment and staffs should be able to tolerate and help others whenever they are in needs

Mission

To provide quality, fast, and flexible service to achieve one hundred percent consumer satisfaction. Staff need to serve the customer will full satisfaction. “Customers always right” should be embedded to the models.

1.1 Business Operation



Staff checking the stock

Staff will calculate the stock at the end of the week

Store manager contact the supplier to restock the stuff

Store manager will decide either restock or not

2.0 Problem Statement

Based on our interview, our group were able to detect that store will have a major problem which are missing file, miscalculating, and misunderstanding. All this problem may happen because they still use the old method to store data, which is handwritten in a piece of paper, manual system. The company is operating manually in recording their inventory data and tracking their sales.

The first problem is lack in information sharing. The store collects all information in a single book. Finding something will take some time as well as they are restricted to an outdated information. This means that if the stock is not updated in the book, staff may not know that there are more stocks but thought they might have run out of it. In addition to that, information that is stored in a single book or paper may be loss or damaged, such as misplacing it or a fire accident. This creates a situation that may make them lose profit and valuable information.

Furthermore, the risk of use all method is miscalculating. This problem may happen when they need to calculate the balance of the stocks and to add the stocks, also in calculating profits and lost. Manually calculating with calculators are trustworthy and easily controlled at the hands of an expert when it comes to managing and calculating something that is high in quantity, however a human will always make a mistake, and a mistake that is on a written piece of paper is hard to trace. This is one of the disadvantages of using an old method for keep the data. Whereas with a computer, it is more efficient and easier to use.

The third one is, the possibility of miscommunication. Handwritings will always be different, some may be easy to read, and some may not. The hard to read and the slightly different shape may be misleading, for example the number seven may be interpreted as two. Because of that, they always need to doublecheck the product to make sure that all is correct. In addition, they need to write properly to make sure all the staff understand the word or number. This can waste their time and will make the company losses some amount of money.

Finally, data redundancy, the company/store uses manual system and Microsoft Excel. When transferring information from one medium to another, it is prone to human error and may cause typographical error. If it keeps happening, it will cost time to assure data entered is correct. In addition, those data may be inconsistent due to it being changed unexpectedly. Anomalies such as update, insertion and deletion anomalies may occur due to Microsoft Excel is a flat-file database.

3.0 Objective

In the era of modernization, all companies in the world are competing to adopt the greatest and latest technology in the industries. A database help business to stay organized and keep information accessible, the big whales in the market can engage better with their customers. With sophisticated software, the data in the database could be used on the fly to make suggestions for business benefits. The data can also help to manage inventory levels, to know when inventory is getting low or when something is out of stock and completing calculations.

The first objective of proposing a database to Provape Semabok Enterprise is because they are heavy user of Microsoft Excel or papers to record all their business transactions and activities. A spreadsheet and a database can appear to be about the same thing. However, for vast amounts of data, a spreadsheet is not nearly as effective as a database. Bringing data into and out of spreadsheets can also be a pain. By the existence of database, Provape Semabok no longer need to do a lot of manual data entry, manual data exporting and importing data to other programs. As solution, Provape Semabok can easily manipulate the data that been stored in the database system as spreadsheet data cannot easily manipulated, with help of an integrated software.

Furthermore, recording sales transactions, inventory counting and profits and loss calculation on papers are very risky. This may lead to miscalculation. Calculations that are being done manually by using papers and calculators somehow are more satisfying and trustworthy. However, human is prone to mistakes. It is more stressful when the calculation involves huge number of data, values, or quantity. To eliminate the tendency to make mistakes and miscalculations, a software that programmed to make calculation based on the data in the database will make the chores. Hence, a database system will increase and improve the efficiency of many business processes in Provape Semabok.

The third objective is to help and assist Provape Semabok in providing accurate information to make decisions. Database provides different kinds of information for the company to expend its business. Information that produced by analysing the database can guide Provape Semabok towards strategic business decision that align with their goals, objectives, and initiatives. Establishing this core capabilities will allow Provape Semabok to move a step forward from its competitors.

To sum it up, Provape Semabok may get used to using the old-fashioned way in recording and calculating their business transactions. With the suggested database, it will be beneficial to the company in many ways. Inconsistencies and inefficiencies could be reduced and indirectly drive the company towards positive progression.

Objective of the system:

i. To improve sales management of the store by providing computerized system
ii. To allow staff record sales transaction at the store
iii. To manage customers membership and order details

4.0 Entity Relationship Diagram

The entity relationship diagram is designed where the CUSTOMER and STAFF entity will have one-to-many relationship connected with the ORDER entity. Customer can contribute to many orders and customerID will be the primary key. All transaction occurs at the store will update the value in ORDER entity.

In addition, the product entity connects with ORDER_PRODUCT entity via one-to-many relationship with productID as the primary key. The ORDER_PRODUCT entity will act as a bridge between PRODUCT and ORDER entity. The ORDER entity will have its own primary key, orderID and staffID, customerID as a support primary key and foreign key. The relationship between the entities is strong relationship due to weak form of entity.

The member and non-member entity are related to customer entity via generalization hierarchies.

The department entities will have one-to-one relationship with staff entity as one staff will on be positioned on one department.

During our interview, we are able to get the business rules from the company.

The following requirement are as **business rules** stated below:

- Staff is associated with their id, name, phone number and salary.
- One staff can handle many orders. Each order is included with order id, customer id, staff id, sales date.
- The staff will be arranged by each department. Each department will have the department ID and department name.
- One product can contribute to many orders. Each of product id, name, price, cost, type, minimum and current inventory will be update into the system.
- For each order, the quantity of product sold will be recorded
- One customer also can contribute to one or many orders. Each customer is associated with his/her unique id and their name.
- The customers able to register becoming members. Customer that did not register will be sorted as non-member where id and name of each member and non-member are recorded for future use.

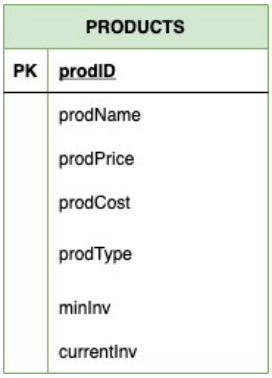


Figure 4.1 Product Entity

The Product Entity will have its own unique product ID (prodID) and store data of the following attribute, which is product name (prodName), product price (prodPrice), product cost (prodCost), product type (prodType) minimum inventory (minInv) and current inventory (currentInv). The purpose of this entity is to allow employee to list down all the products available at the store. The attribute of products such as name, price, type and cost will be recorded for ordering from supplier and selling to customers.

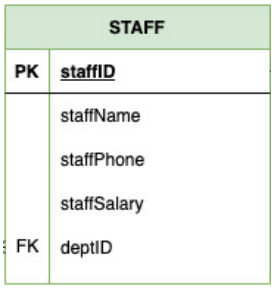


Figure 4.2 Staff Entity

On top of that, the Staff Entity will hold data about each staff including their staff ID (staffID) as primary key, name (staffName), phone (staffPhone), salary (staffSalary) and which department they work in based on the department ID (deptID) in Department Entity as foreign key. The main purpose of this entity to allow branch manager to manage their staffs in the store.

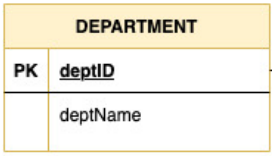


Figure 4.3 Department Entity

On top of that, the Department Entity will hold data of department ID (deptID) as primary key and department name (deptName). The main purpose of this entity to allow manager to manage their staffs in the store. This will helps ease the flow of organization for promotion or assigning task.

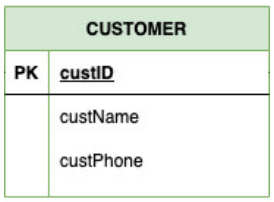


Figure 4.4 Customer Entity

Besides that, the Customer Entity will hold information regarding the customers such as customer id (custID), customer name (custName), customer phone number (custPhone). The customer is the core component to measure the demand of the market by calculating the product bought at the store. The staff will handle the operation of managing the customer information to get the accurate data for future use.

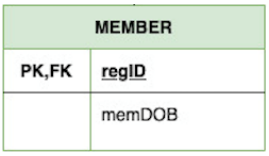


Figure 4.5 Member Entity

Besides that, the member entity will hold information of the customers those registers as a member such as id, name. The member entity will hold data of member's date of birth as member will receive promotional discount and free product on their birthday.

NONMEMBER	
PK,FK	regID
	nmemEmail

Figure 4.6 Non-member Entity

Finally, the non-member entity will hold information of the customers that do not register as a member. The non-member entity will help staff to differentiate customers which are members or the other as non-members. The non-member will receive promotional updates via email from the company.

ORDER	
PK	orderID
PK,FK1	staffID
PK,FK2	custID
	orderDate

Figure 4.7 Order Entity

Order entity will store all the information about the order made by the customer. The entity will have orderID as the primary key, data about staff managing the order (staffID) and customer(custID) that purchase the order will be stored in the table. In addition, ther order date also will be recorded for every transaction.

ORDER_PRODUCT	
PK,FK1	orderID
PK,FK2	prodID
	salesQuantity

Figure 4.8 Order-Product Entity

Finally, Order-Product entity will store information about the product purchased for each order made by the customer. The information such as quantity of the product also will be recorded together for calculation of total price.

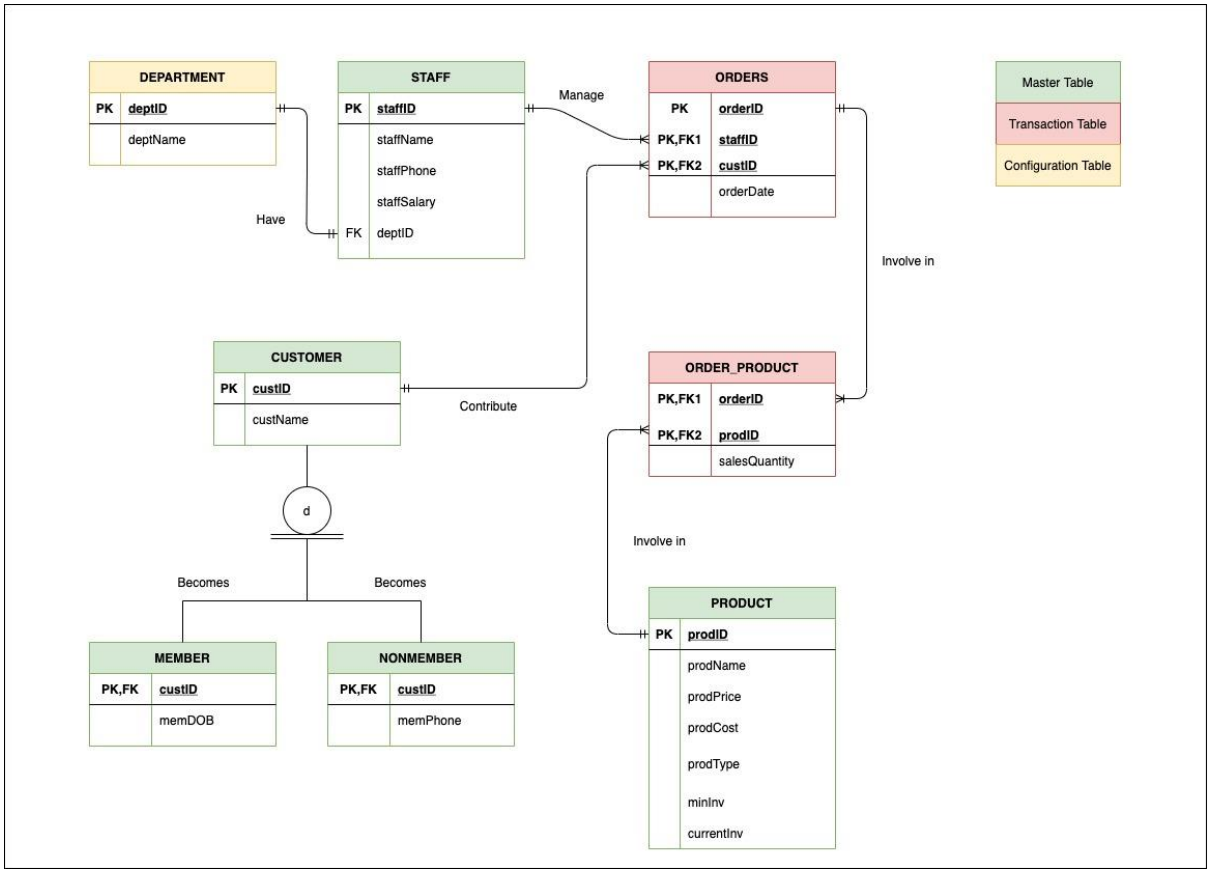


Figure 4.9 Revised Entity Relationship Diagram (ERD)

5.0 Revised 3NF Relational Schema

1NF
(**deptID**, deptName, **staffID**, staffName,
staffPhone, staffSalary, **custID**, custName, memDob, memPhone, **prodID**, prodName, prodPrice,
prodCost, prodType, minInv, currentInv**, orderID**, orderDate, salesQuantity)

Partial Dependencies:
(**staffID** → staffName, staffPhone, staffSalary, deptID, deptName**)**
(**custID** → custName)
(**custID** → memDOB)
(**custID** → memPhone)
(**orderID**, **staffID, custID** → orderDate)
(**prodID** → prodName, prodPrice, prodCost, prodType, minInv, currentInv)
(**orderID, prodID** → salesQuantity)

Transitive Dependencies
(**deptID** → deptName)

2NF
STAFF (**staffID**, staffName, staffPhone, staffSalary, deptID, deptName)
CUSTOMER (**custID,** custName)
MEMBER (**custID*,** memDOB)
NONMEMBER (**custID*,** memPhone)
ORDER (**orderID**, **staffID*, custID*,** orderDate)
PRODUCTS (**prodID,** prodName, prodPrice, prodCost, prodType, minInv, currentInv)
ORDER_PRODUCT (**orderID*, prodID***, salesQuantity)

3NF
STAFF (**staffID**, staffName, staffPhone, staffSalary, deptID*)
CUSTOMER (**custID,** custName)
MEMBER (**custID*,** memDOB)
NONMEMBER (**custID*,** memPhone)
ORDER (**orderID**, **staffID*, custID*,** orderDate)
PRODUCTS (**prodID,** prodName, prodPrice, prodCost, prodType, minInv. currentInv)
ORDER_PRODUCT (**orderID*, prodID***, salesQuantity)
DEPARTMENT (**deptID**, deptName)

6.0 Data Integrity

Table DEPARTMENT

Object Type **TABLE** Object **DEPARTMENT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPARTMENT	DEPTID	VARCHAR2	5	-	-	1	-	-	-
	DEPTNAME	VARCHAR2	50	-	-	-	✓	-	-

Table STAFF

Object Type **TABLE** Object **STAFF**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STAFF	STAFFID	VARCHAR2	5	-	-	1	-	-	-
	STAFFNAME	VARCHAR2	80	-	-	-	✓	-	-
	STAFFPHONE	VARCHAR2	12	-	-	-	✓	-	-
	STAFFSALARY	NUMBER	-	5	0	-	✓	-	-
	DEPTID	VARCHAR2	5	-	-	-	-	-	-

Table ORDERS

Object Type **TABLE** Object **ORDERS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDERS	ORDERID	VARCHAR2	5	-	-	1	-	-	-
	STAFFID	VARCHAR2	5	-	-	-	-	-	-
	CUSTID	VARCHAR2	5	-	-	-	-	-	-
	ORDERDATE	DATE	7	-	-	-	✓	-	-

Table ORDER_PRODUCT

Object Type **TABLE** Object **ORDER_PRODUCT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ORDER_PRODUCT	ORDERID	VARCHAR2	5	-	-	-	-	-	-
	PRODID	VARCHAR2	5	-	-	-	-	-	-
	SALESQUANTITY	NUMBER	-	5	0	-	✓	-	-

Table PRODUCT

Object Type **TABLE** Object **PRODUCT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
PRODUCT	PRODID	VARCHAR2	5	-	-	1	-	-	-
	PRODNAME	VARCHAR2	80	-	-	-	✓	-	-
	PRODPRICE	NUMBER	-	5	2	-	✓	-	-
	PRODCOST	NUMBER	-	5	2	-	✓	-	-
	PRODTYPE	VARCHAR2	20	-	-	-	✓	-	-
	MININV	NUMBER	-	5	0	-	✓	-	-
	CURRENTINV	NUMBER	-	5	0	-	✓	-	-

Table CUSTOMER

Object Type **TABLE** Object **CUSTOMER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	CUSTID	VARCHAR2	5	-	-	1	-	-	-
	CUSTNAME	VARCHAR2	80	-	-	-	✓	-	-

Table NONMEMBER

Object Type **TABLE** Object **NONMEMBER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
NONMEMBER	CUSTID	VARCHAR2	5	-	-	-	✓	-	-
	MEMPHONE	VARCHAR2	12	-	-	-	✓	-	-

Table MEMBER

Object Type **TABLE** Object **MEMBER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
MEMBER	CUSTID	VARCHAR2	5	-	-	-	-	-	-
	MEMDOB	DATE	7	-	-	-	✓	-	-

7.0 Data Dictionary

Table Department

Field Name	Data Type	Data Format	Field Size	Description	Example
deptid	varchar		5	Primary key	10030
deptName	varchar		50	Name of Department	Human Resource

Table Staff

Field Name	Data Type	Data Format	Field Size	Description	Example
staffID	varchar		5	Primary key	20210
staffname	varchar		80	Name of staff	Kamarul Akim
staffphone	varchar	NNNNNNNNNN	12	Staff phone number	0162433036
staffsalary	number	NNNNN	(5,0)	Salary of staff	1500.00
deptID	varchar		5	Foreign Key	10030

Table Orders

Field Name	Data Type	Data Format	Field Size	Description	Example
orderID	varchar		5	Primary key	50505
staffID	varchar		5	Primary key, Foreign Key	20210
custID	varchar		5	Primary key, Foreign Key	10101
orderDate	date	NNNN-NN-NN	7	Date of Order	2021-06-18

Table Order_Product

Field Name	Data Type	Data Format	Field Size	Description	Example
orderID	varchar		5	Primary key, Foreign Key	50505
prodID	varchar		5	Primary key, Foreign Key	80800
salesQuantity	number	NNNN	(5,0)	Quantity of Product	10

Table Product

Field Name	Data Type	Data Format	Field Size	Description	Example
prodID	varchar		5	Primary key	80800
prodName	varchar		80	Name of Product	Caliburn G
prodPrice	number	NN.NN	(5,2)	Price of Product	30.50
prodCost	number	NN.NN	(5,2)	Cost of Product	10.00
prodType	varchar		15	Type of Product	Mod
minInv	number		(5,0)	Minimum Inventory of Product	10
currentInv	Number		(5,0)	Current amount of stock	15

Table Customer

Field Name	Data Type	Data Format	Field Size	Description	Example
custID	varchar		5	Primary key	20210
custName	varchar		80	Name of customer	Kamarul Alim

Table Member

Field Name	Data Type	Data Format	Field Size	Description	Example
custID	varchar		5	Primary key, Foreign Key	20210
memDOB	date	NNNN-NN-NN	7	Customer birthdate	2000-06-18

Table Non-Member

Field Name	Data Type	Data Format	Field Size	Description	Example
custID	varchar		5	Primary key, Foreign Key	20210
memPhone	number	NNNNNNNN	12	Customer phone number	0197649482

8.0 Data Definition Language

```
CREATE TABLE  "DEPARTMENT"
  (    "DEPTID" VARCHAR2(5) NOT NULL ENABLE,
   "DEPTNAME" VARCHAR2(50),
    CONSTRAINT "DEPARTMENT_PK" PRIMARY KEY ("DEPTID")
 USING INDEX  ENABLE
  )
```

Column Name	Data Type	Nullable	Default	Primary Key
DEPTID	VARCHAR2(5)	No	-	1
DEPTNAME	VARCHAR2(50)	Yes	-	-

```
CREATE TABLE  "STAFF"
  (    "STAFFID" VARCHAR2(5) NOT NULL ENABLE,
   "STAFFNAME" VARCHAR2(80),
   "STAFFPHONE" VARCHAR2(12),
   "STAFFSALARY" NUMBER(5,0),
   "DEPTID" VARCHAR2(5) NOT NULL ENABLE,
    CONSTRAINT "STAFF_PK" PRIMARY KEY ("STAFFID")
 USING INDEX  ENABLE
  )
/
ALTER TABLE  "STAFF" ADD CONSTRAINT "STAFF_FK" FOREIGN KEY ("DEPTID")
     REFERENCES  "DEPARTMENT" ("DEPTID") ENABLE
/
```

Column Name	Data Type	Nullable	Default	Primary Key
STAFFID	VARCHAR2(5)	No	-	1
STAFFNAME	VARCHAR2(80)	Yes	-	-
STAFFPHONE	VARCHAR2(12)	Yes	-	-
STAFFSALARY	NUMBER(5,0)	Yes	-	-
DEPTID	VARCHAR2(5)	No	-	-

```
CREATE TABLE  "ORDERS"
  (    "ORDERID" VARCHAR2(5) NOT NULL ENABLE,
   "STAFFID" VARCHAR2(5) NOT NULL ENABLE,
   "CUSTID" VARCHAR2(5) NOT NULL ENABLE,
   "ORDERDATE" DATE,
    CONSTRAINT "ORDERS_PK" PRIMARY KEY ("ORDERID")
 USING INDEX  ENABLE
  )
/
ALTER TABLE  "ORDERS" ADD CONSTRAINT "ORDERS_FK" FOREIGN KEY ("CUSTID")
    REFERENCES  "CUSTOMER" ("CUSTID") ENABLE
/
ALTER TABLE  "ORDERS" ADD CONSTRAINT "ORDERS_FK2" FOREIGN KEY ("STAFFID")
    REFERENCES  "STAFF" ("STAFFID") ENABLE
/
```

Column Name	Data Type	Nullable	Default	Primary Key
ORDERID	VARCHAR2(5)	No	-	1
STAFFID	VARCHAR2(5)	No	-	-
CUSTID	VARCHAR2(5)	No	-	-
ORDERDATE	DATE	Yes	-	-

```
CREATE TABLE  "ORDER_PRODUCT"
  (    "ORDERID" VARCHAR2(5) NOT NULL ENABLE,
   "PRODID" VARCHAR2(5) NOT NULL ENABLE,
   "SALESQUANTITY" NUMBER(5,0)
  )
/
ALTER TABLE  "ORDER_PRODUCT" ADD CONSTRAINT "ORDER_PRODUCT_FK" FOREIGN KEY ("ORDERID")
    REFERENCES  "ORDERS" ("ORDERID") ENABLE
/
ALTER TABLE  "ORDER_PRODUCT" ADD CONSTRAINT "ORDER_PRODUCT_FK2" FOREIGN KEY ("PRODID")
    REFERENCES  "PRODUCT" ("PRODID") ENABLE
/
```

Column Name	Data Type	Nullable	Default	Primary Key
ORDERID	VARCHAR2(5)	No	-	-
PRODID	VARCHAR2(5)	No	-	-
SALESQUANTITY	NUMBER(5,0)	Yes	-	-

```
CREATE TABLE  "PRODUCT"
  (    "PRODID" VARCHAR2(5) NOT NULL ENABLE,
   "PRODNAME" VARCHAR2(80),
   "PRODPRICE" NUMBER(5,2),
   "PRODCOST" NUMBER(5,2),
   "PRODTYPE" VARCHAR2(20),
   "MININV" NUMBER(5,0),
   "CURRENTINV" NUMBER(5,0),
    CONSTRAINT "PRODUCT_PK" PRIMARY KEY ("PRODID")
 USING INDEX  ENABLE
  )
/
```

Column Name	Data Type	Nullable	Default	Primary Key
PRODID	VARCHAR2(5)	No	-	1
PRODNAME	VARCHAR2(80)	Yes	-	-
PRODPRICE	NUMBER(5,2)	Yes	-	-
PRODCOST	NUMBER(5,2)	Yes	-	-
PRODTYPE	VARCHAR2(20)	Yes	-	-
MININV	NUMBER(5,0)	Yes	-	-
CURRENTINV	NUMBER(5,0)	Yes	-	-

```
CREATE TABLE  "CUSTOMER"
  (
   "CUSTID" VARCHAR2(5) NOT NULL ENABLE,
   "CUSTNAME" VARCHAR2(80),
    CONSTRAINT "CUSTOMER_PK" PRIMARY KEY ("CUSTID")
 USING INDEX  ENABLE
  )
```

Column Name	Data Type	Nullable	Default	Primary Key
CUSTID	VARCHAR2(5)	No	-	1
CUSTNAME	VARCHAR2(80)	Yes	-	-

```
CREATE TABLE  "MEMBER"
  (    "CUSTID" VARCHAR2(5) NOT NULL ENABLE,
   "MEMDOB" DATE
  )
/
ALTER TABLE  "MEMBER" ADD CONSTRAINT "MEMBER_FK" FOREIGN KEY ("CUSTID")
     REFERENCES  "CUSTOMER" ("CUSTID") ENABLE
/
```

Column Name	Data Type	Nullable	Default	Primary Key
CUSTID	VARCHAR2(5)	No	-	-
MEMDOB	DATE	Yes	-	-

```
CREATE TABLE  "NONMEMBER"
  (    "CUSTID" VARCHAR2(5),
   "MEMPHONE" VARCHAR2(12)
  )
/
ALTER TABLE  "NONMEMBER" ADD CONSTRAINT "NONMEMBER_FK" FOREIGN KEY ("CUSTID")
     REFERENCES  "CUSTOMER" ("CUSTID") ENABLE
/
```

Column Name	Data Type	Nullable	Default	Primary Key
CUSTID	VARCHAR2(5)	Yes	-	-
MEMPHONE	VARCHAR2(12)	Yes	-	-

9.0 Data Manipulation Language

Create new customer

```
INSERT INTO CUSTOMER
VALUES('10016', 'Zub Quzaini')
```

EDIT	CUSTID	CUSTNAME
	10004	Ahmad Zahir
	10005	Faiz Syakir
	10010	Amzar Adam
	10015	Nik Qistina
	10001	Syahirah Ros
	10007	Fayyad Bashir
	10000	Arfan Zaini
	10003	Aina Farhana
	10011	Ros Farhana
	10012	Adam Amir
	10006	Fazlina Rusdina
	10008	Nur Amalina
	10013	Izwan Shah
	10016	Zub Quzaini
	10002	Naqeeb Rizal

Read all members

```
SELECT * FROM MEMBER
```

CUSTID	MEMDOB
10000	03/02/2000
10001	04/02/2000
10002	06/02/2000
10003	06/13/1987
10004	06/02/1989
10005	12/12/1989
10006	11/13/1990
10007	02/21/1991
10008	06/13/1991
10009	03/21/1992

Update product price

```
UPDATE PRODUCT
SET PRODPRICE = 35
WHERE PRODID = '30001'
```

PRODID	PRODNAME	PRODPRICE	PRODCOST	PRODTYPE	MININV	CURRENTINV
30001	Juicy Mango	35	18	Flavour	10	25
30002	Juicy Grape	30	18	Flavour	20	55
30003	Juicy Lychee	30	18	Flavour	50	120
30004	Milo Tabur	35	21	Flavour	40	80
30005	Creamy Vanilla	35	21	Flavour	50	70
30006	Popcorn Boom	25	14	Flavour	20	20
30007	Minty Watermelon	20	9	Flavour	20	52
30008	Minty Salak	20	9	Flavour	40	90
30009	Minty Mentos	20	9	Flavour	50	30
30010	Cosmo Coil	15	8	Parts	1	200
30011	Batteries	32.5	28.2	Parts	20	23
30012	Cotton	9.2	4.8	Parts	20	55
30013	Smok Pod	80.8	69.2	Mod	10	22
30014	FreeMax Pod	100.5	75.5	Mod	15	20
30015	VaporFi Pod	95.9	65.5	Mod	20	20

Delete Customer

```
DELETE FROM CUSTOMER
WHERE CUSTID = '10016'
```

10.0 System User Interface

Login Page

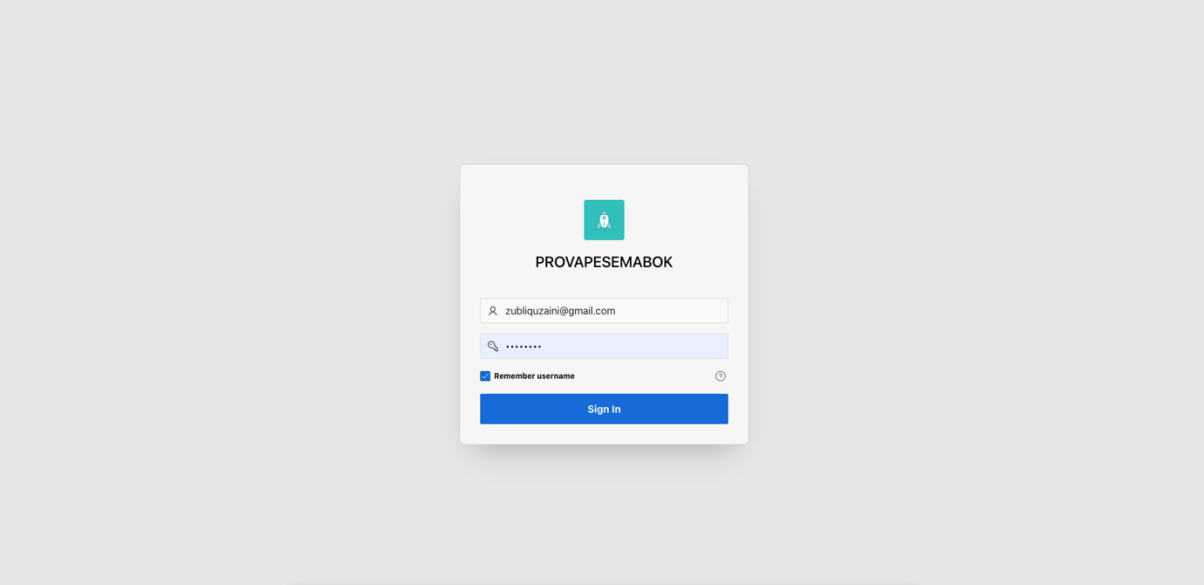


Figure 10.1 Users need to enter the login credentials such as email and password to enter the system.

Home Page

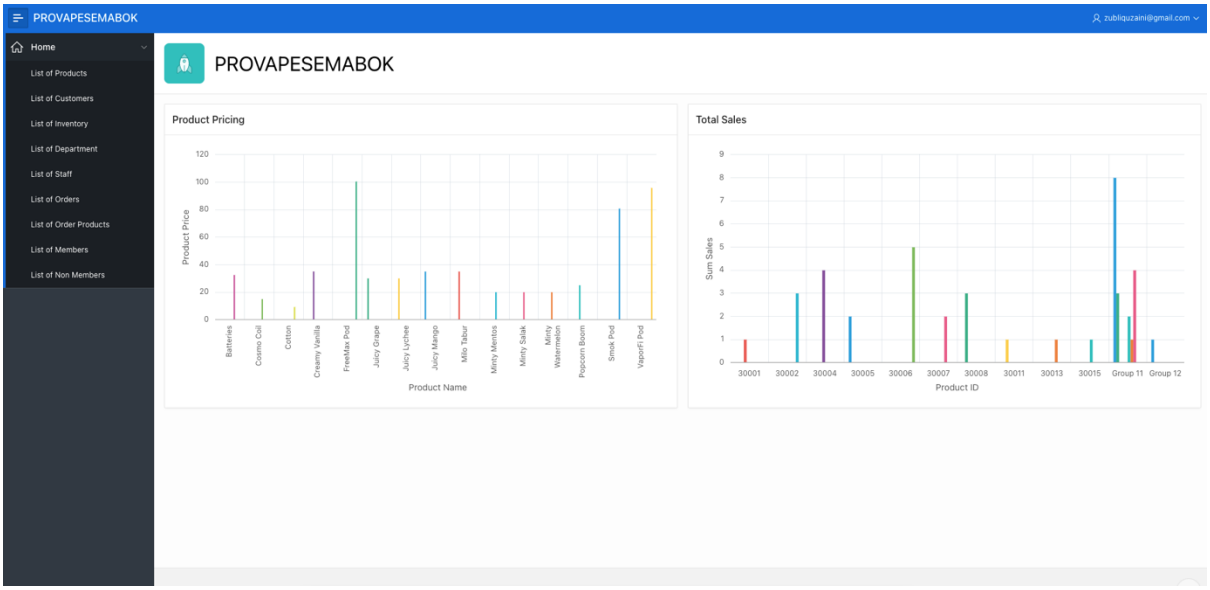


Figure 10.2 Users will be presented with the homepage. At the homepage, users can view the reporting of Product Pricing and the Total Sales.

Product Page

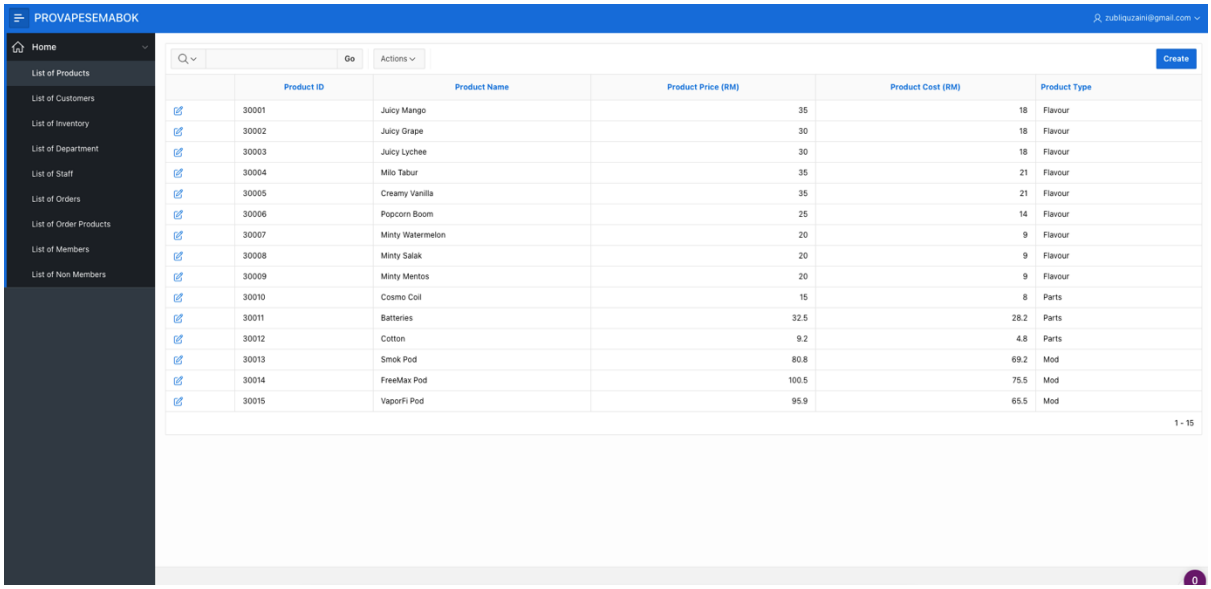


Figure 10.3 Users can view all the product available at the store

Figure 10.4 Users can enter product using the form

Figure 10.5 Users can update the product details

List of Customer

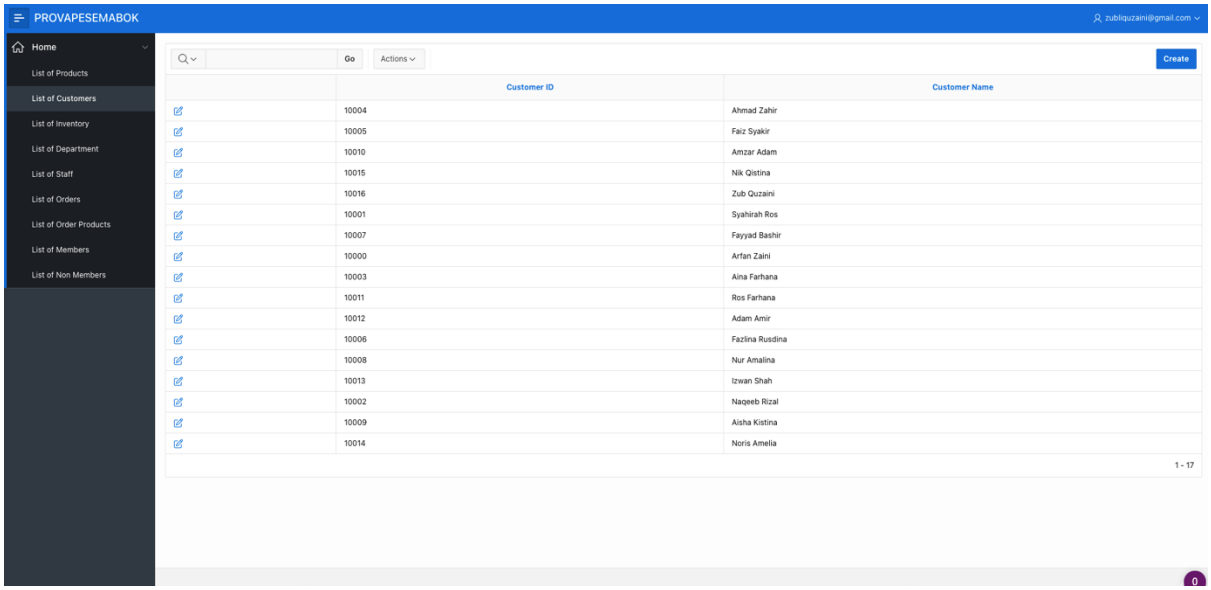


Figure 10.6 Users can view all the customer registered at the store

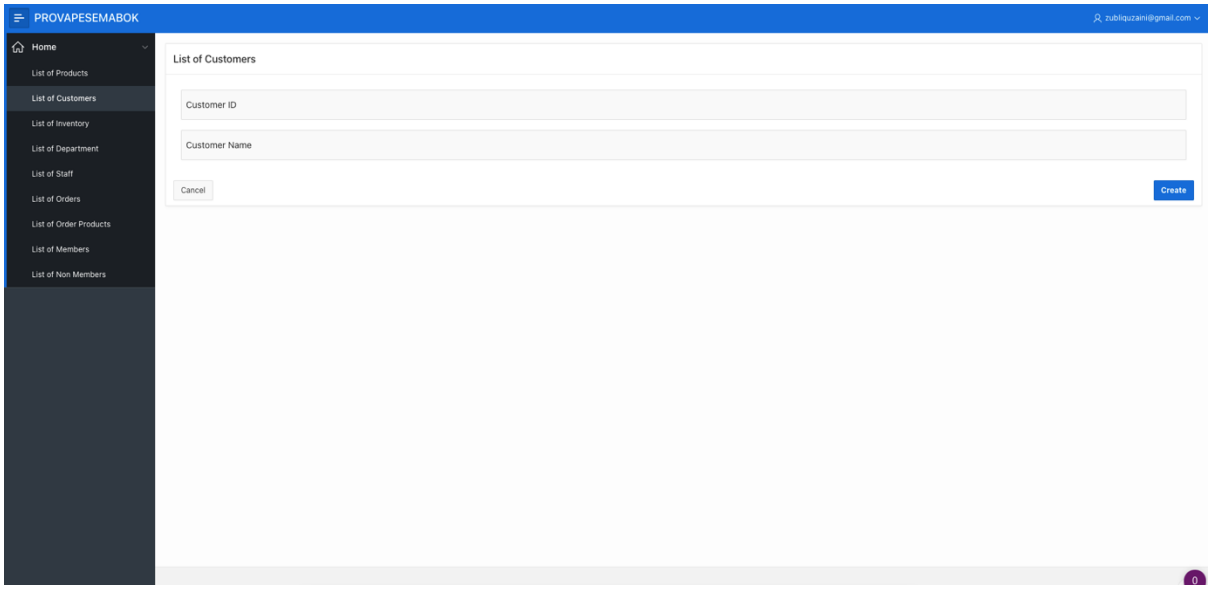


Figure 10.7 Users can register customer using the form

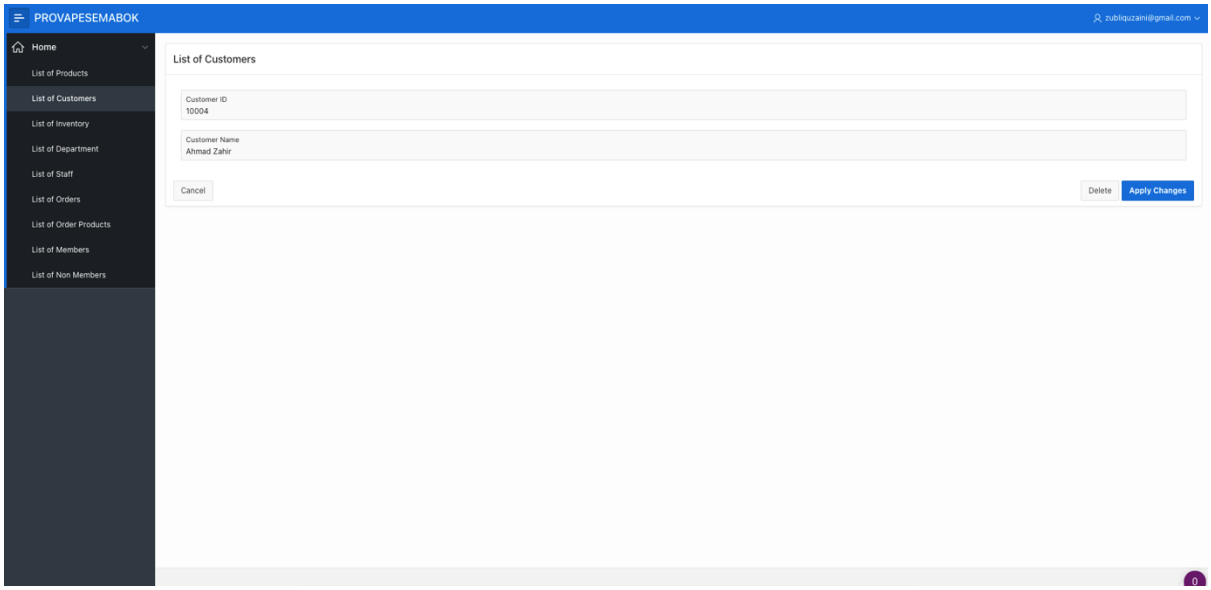


Figure 10.8 Users can update the customer details

List of Department

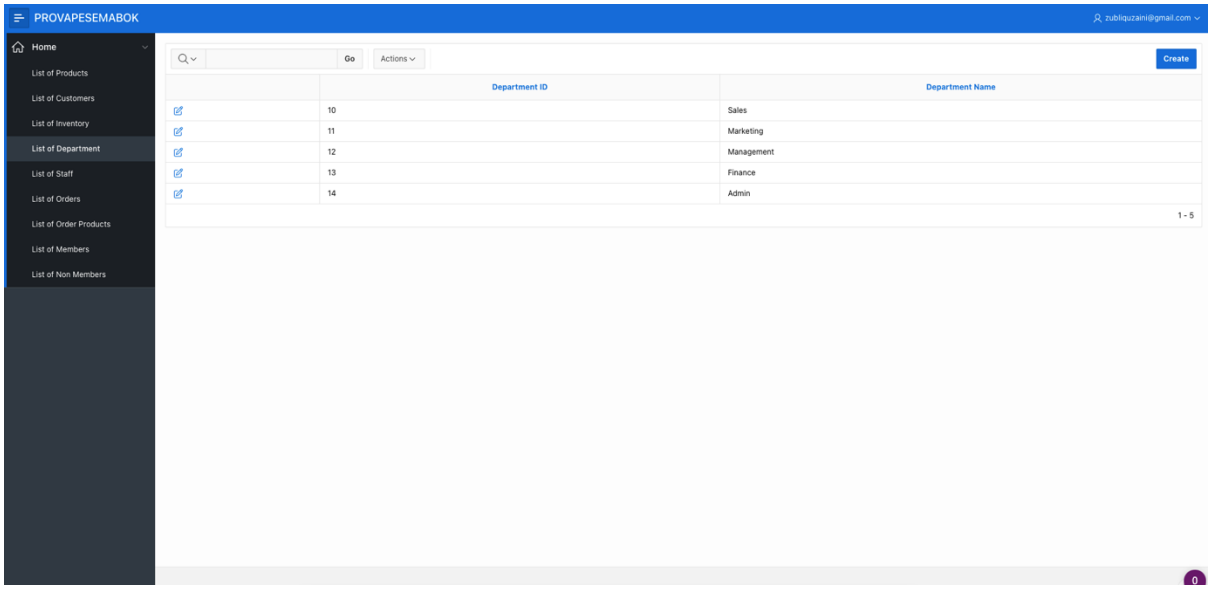


Figure 10.9 Users can view all the department at the store

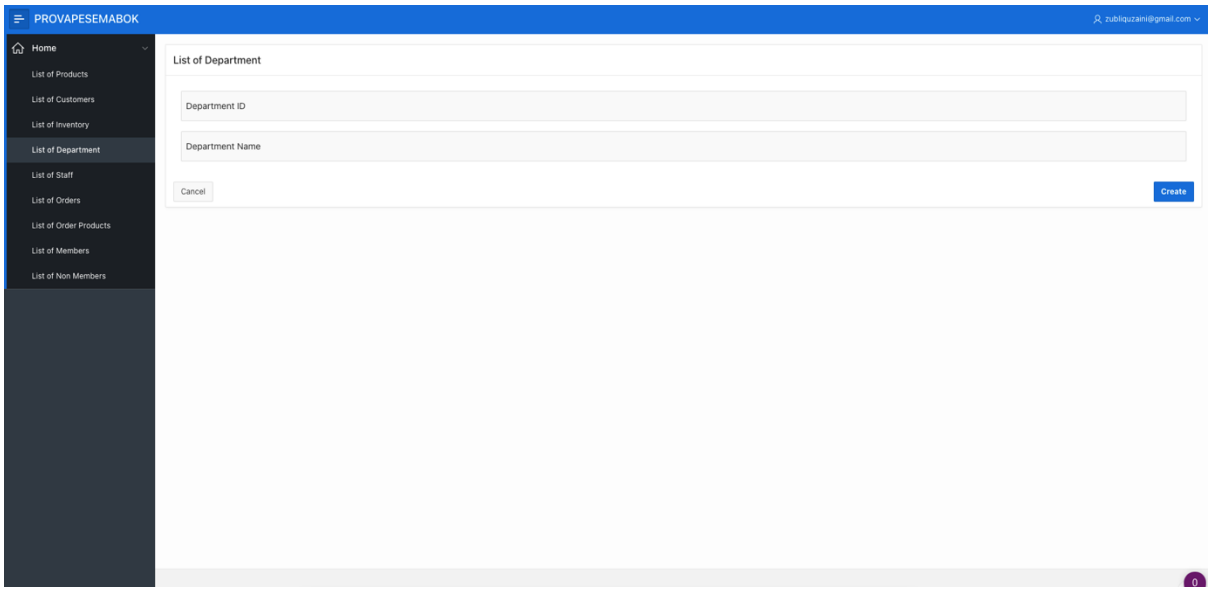


Figure 10.10 Users can add department using the form

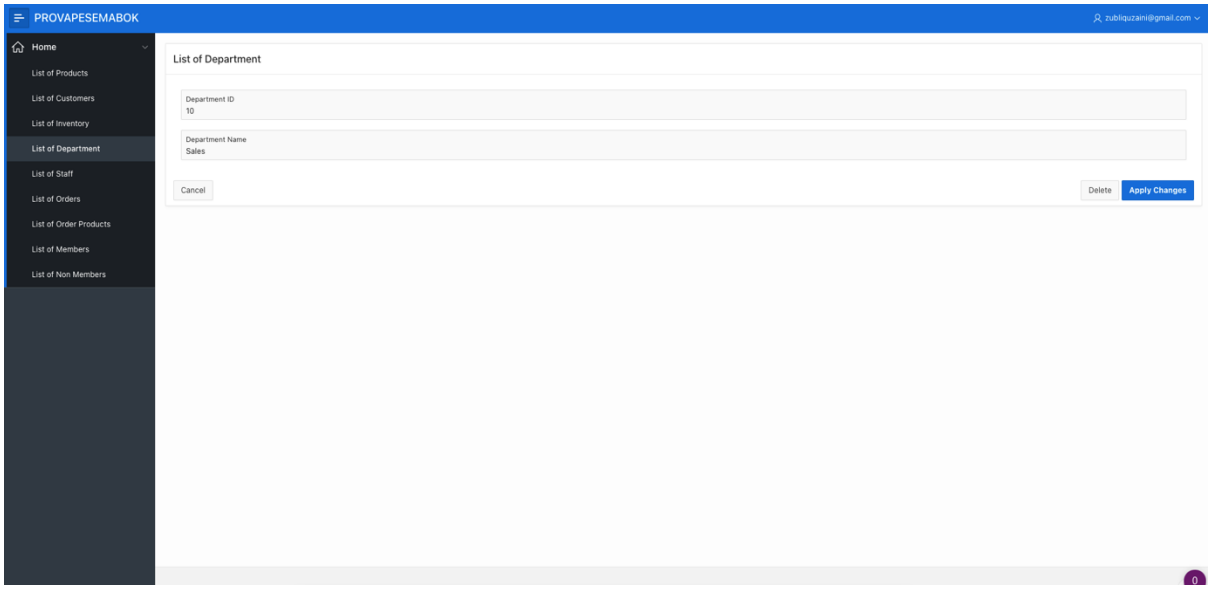


Figure 10.11 Users can update the department details

List of Staff

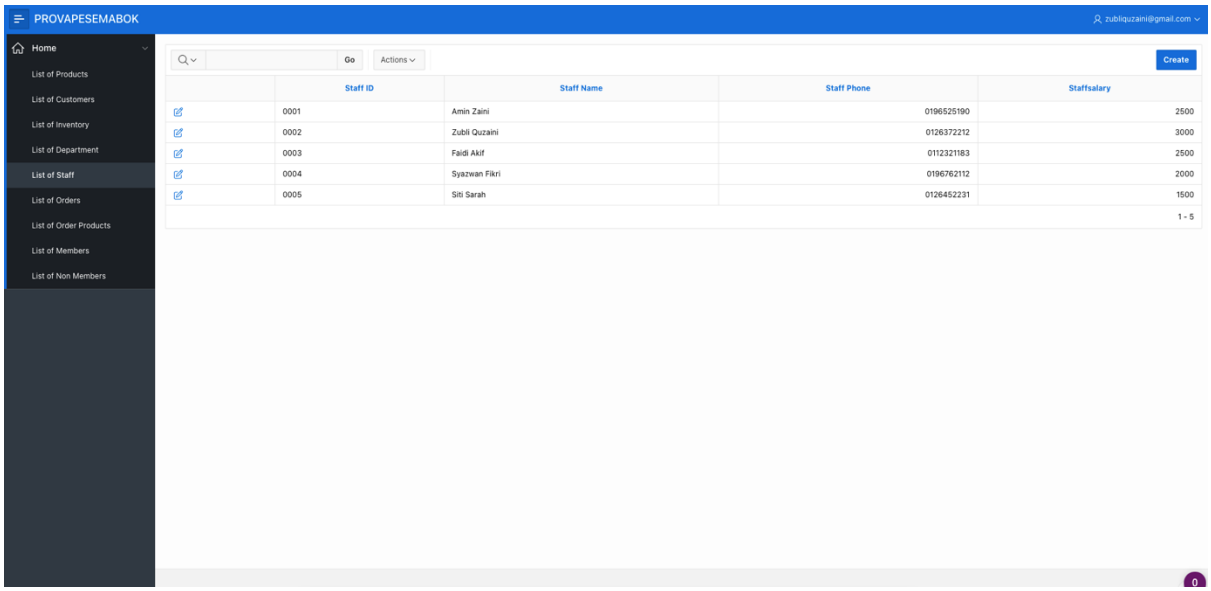


Figure 10.12 Users can view all the staff registered at the store

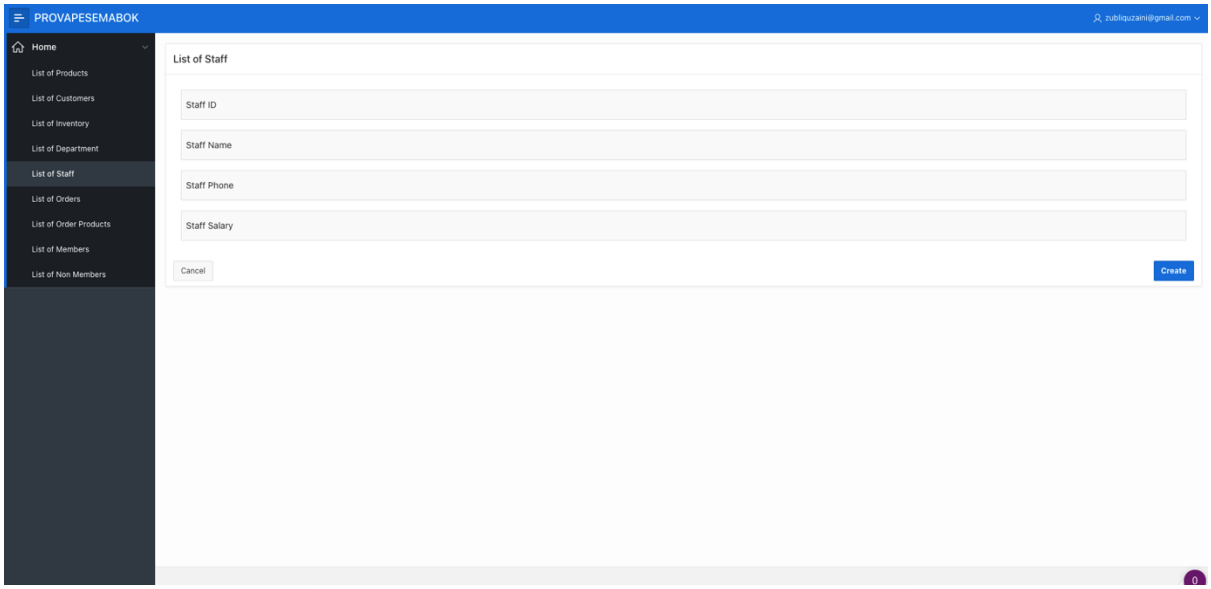


Figure 10.13 Users can register staff using the form

List of Orders

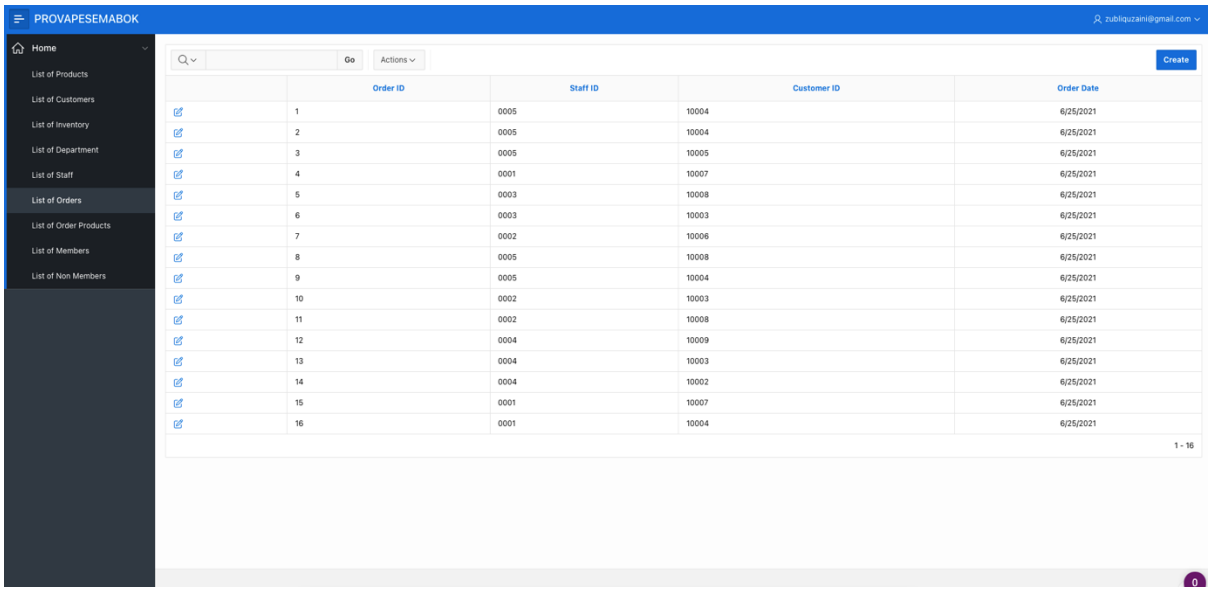


Figure 10.14 Users can view all the orders made by the customer

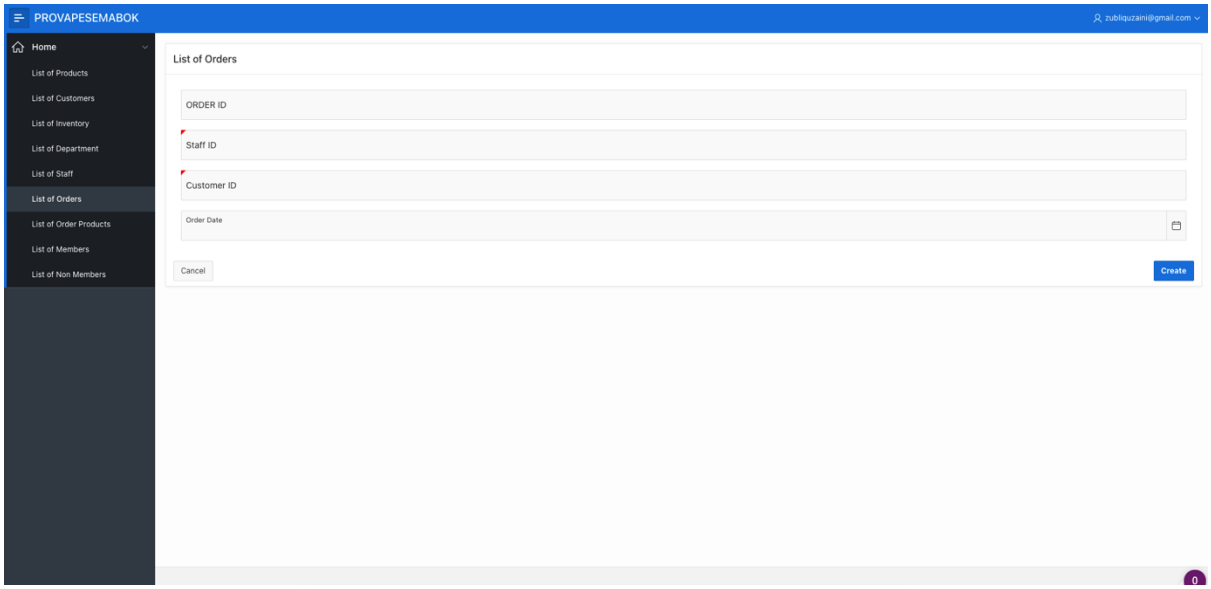


Figure 10.15 Users can add order using the form

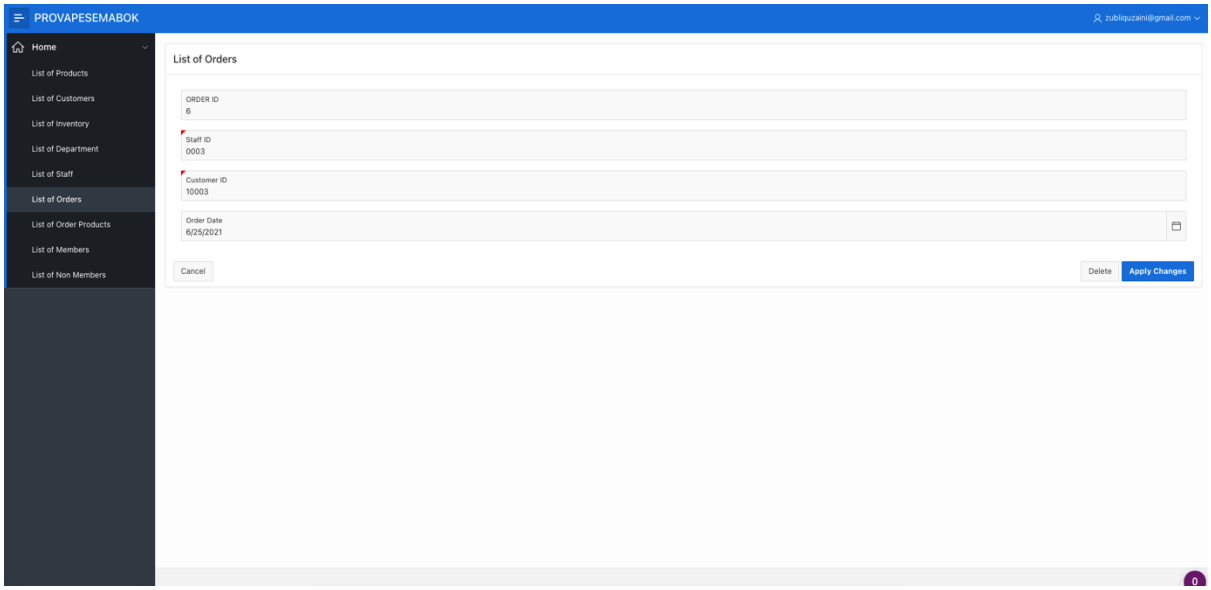


Figure 10.16 Users can update order using the form

List of Order Product

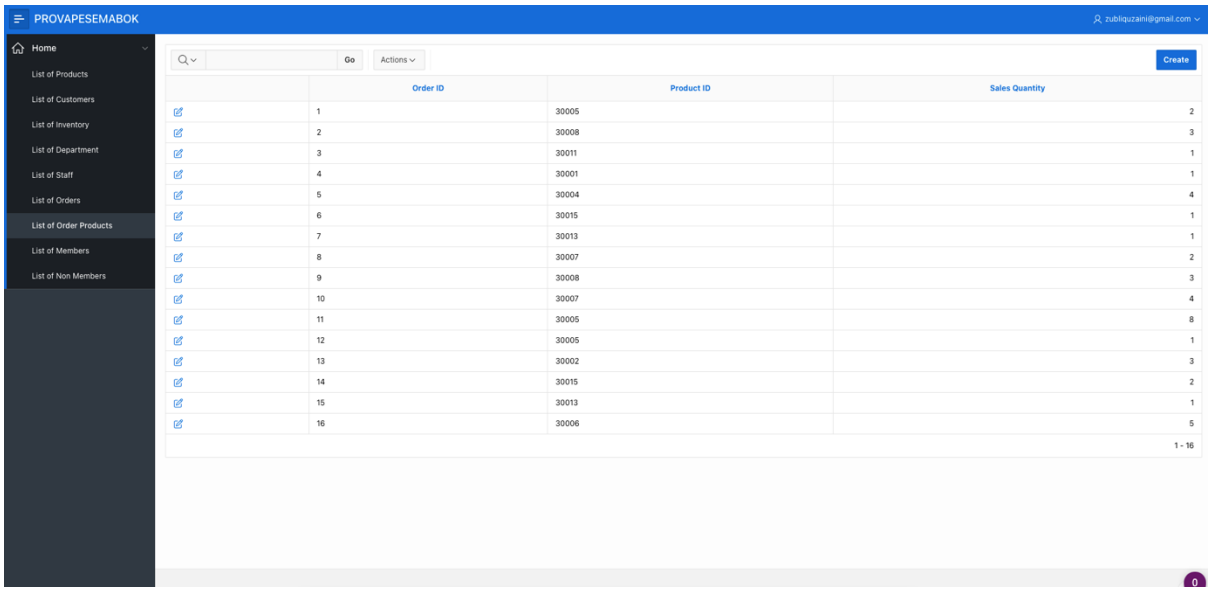


Figure 10.20 Users can view all the product for each orders at the store

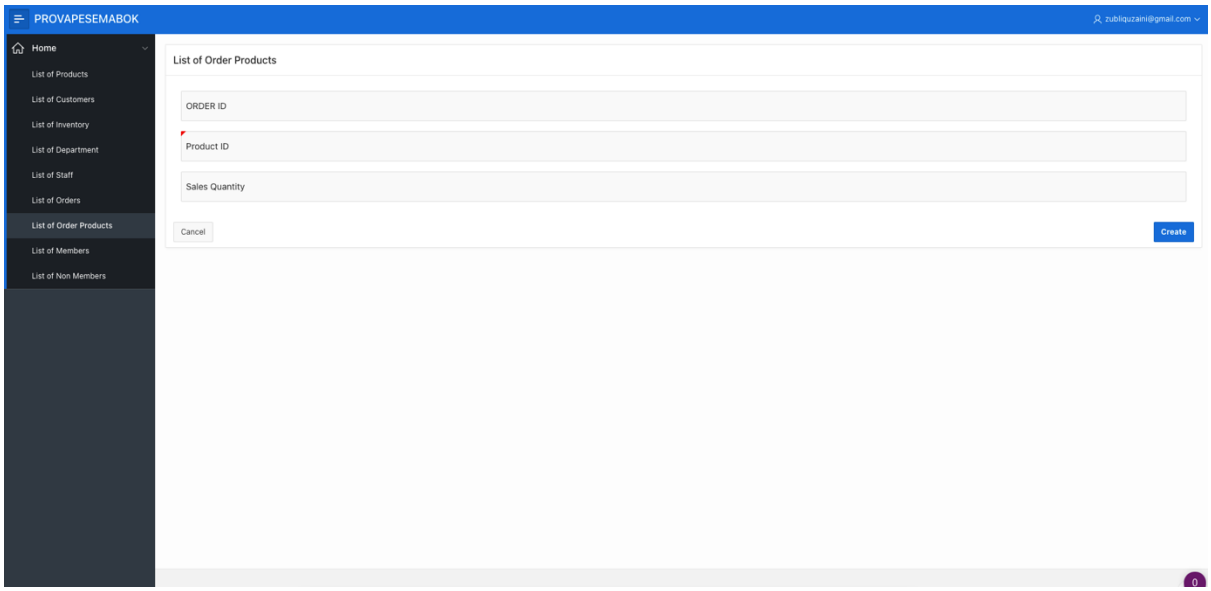


Figure 10.21 Users can add product for each order using the form

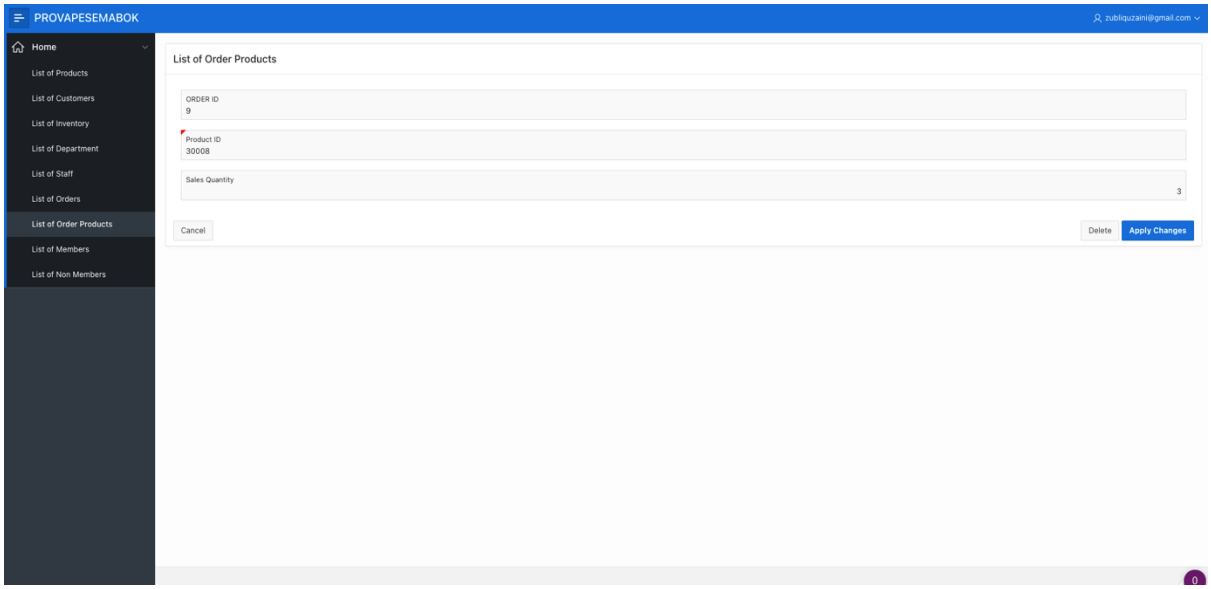


Figure 10.22 Users can update products for each order

List of Members

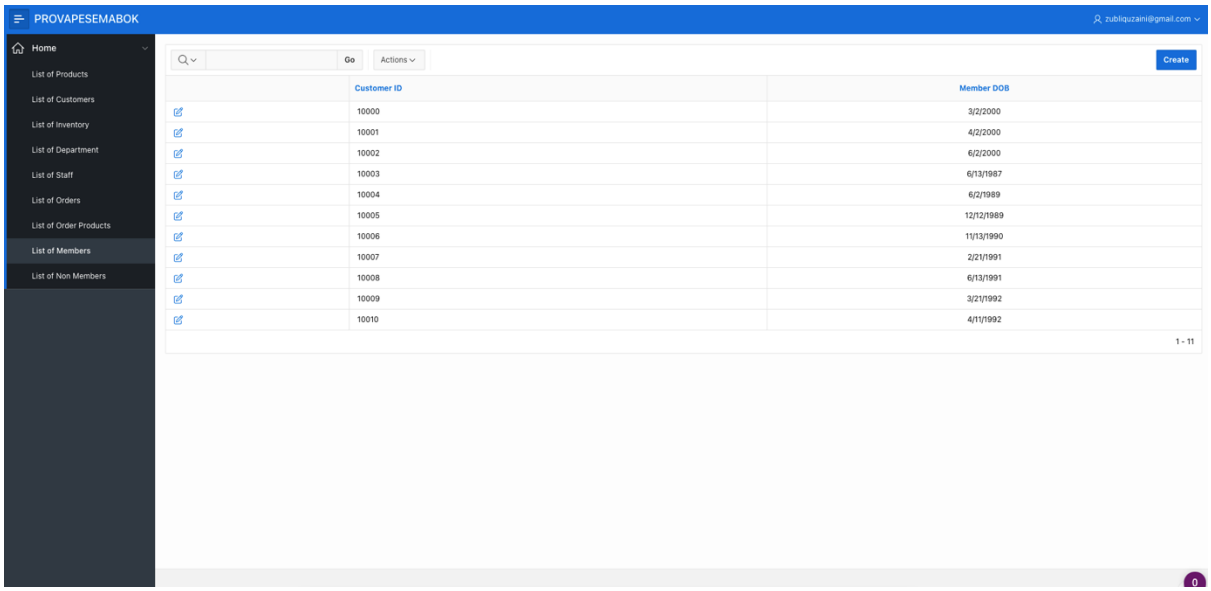


Figure 10.23 Users can all members registered at the store

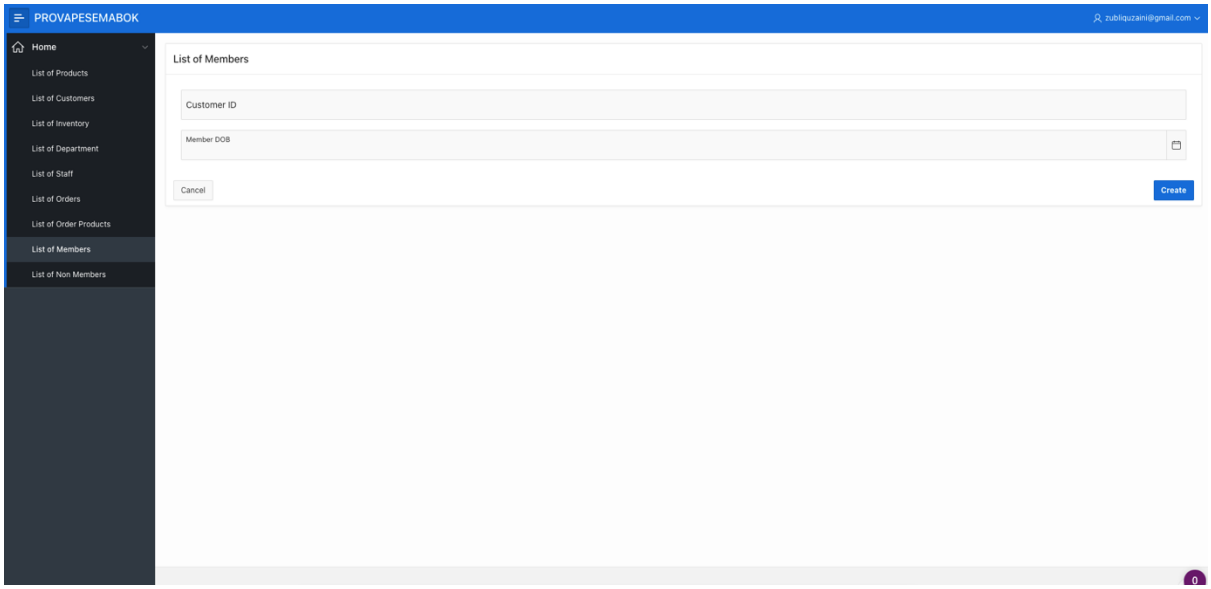


Figure 10.24 Users can add new members using the form

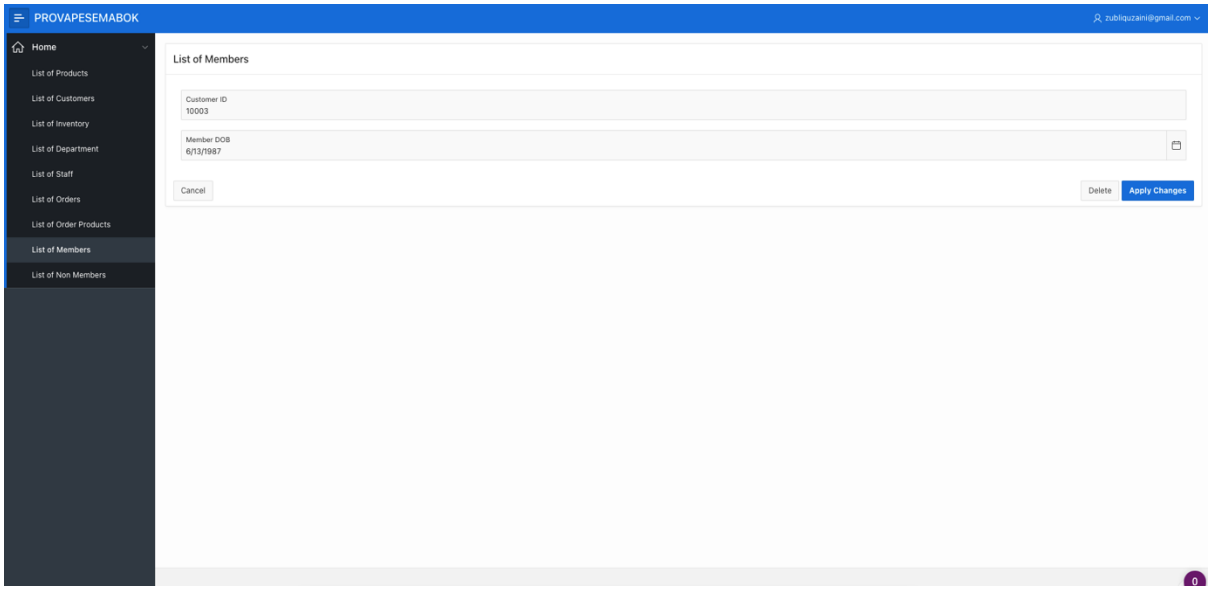


Figure 10.25 Users can update member details

List of Non-Members

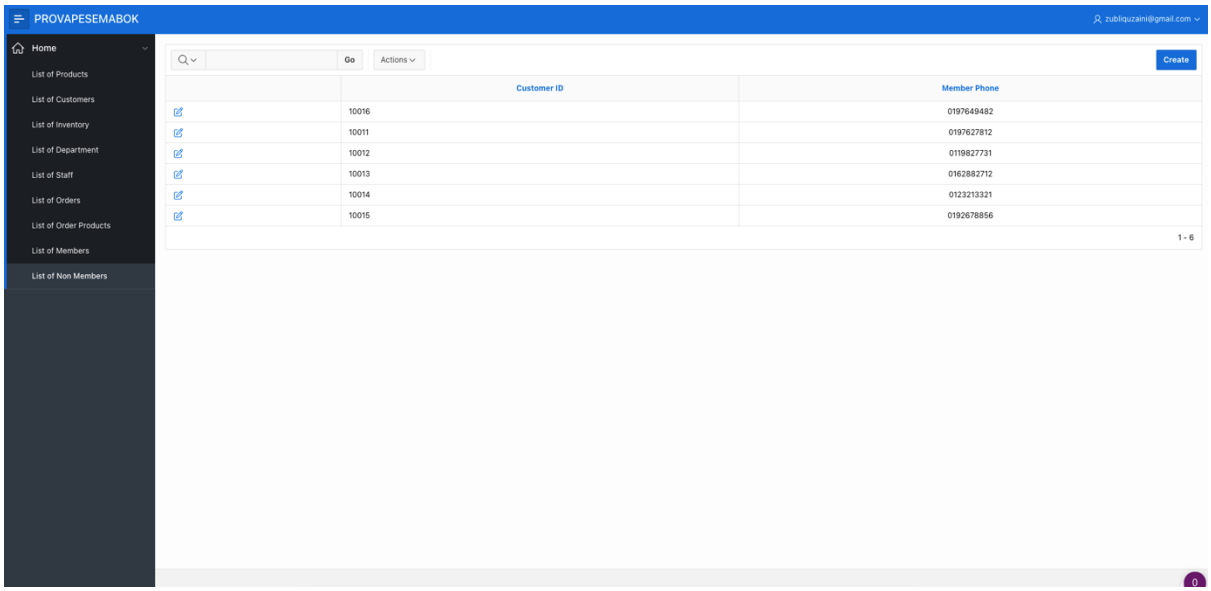


Figure 10.26 Users can view all the non-member registered at the store

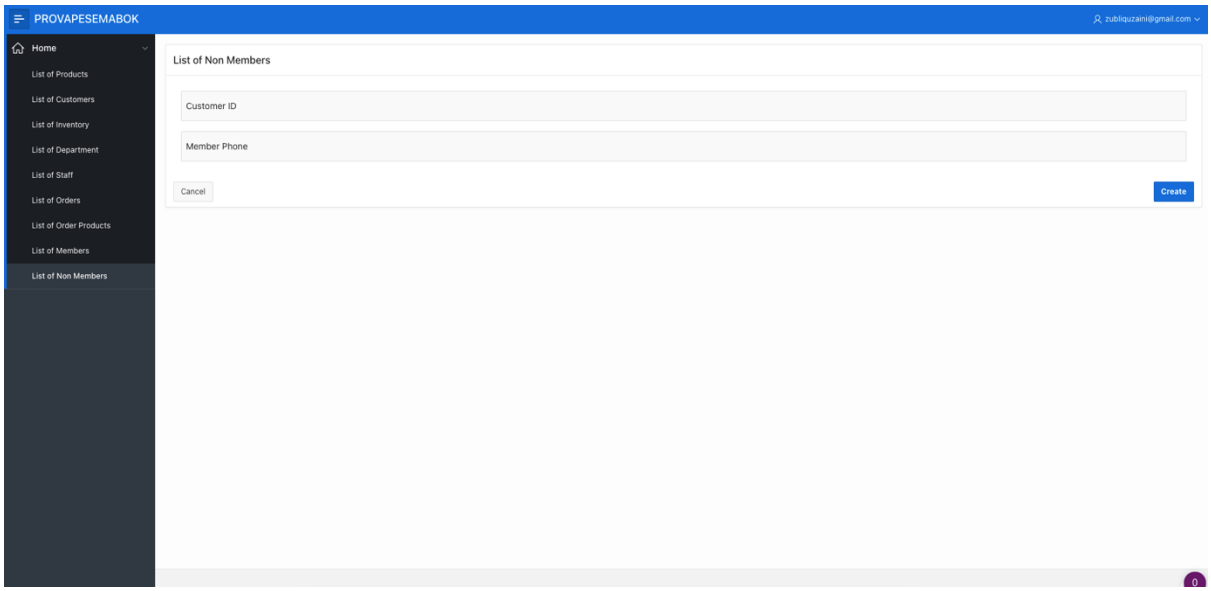


Figure 10.27 Users can add new non-member using the form

11.0 SQL Queries Questions

QUESTION 1

List staff id, staff name, and staff salary that is more than or equal to 2500 and does not have the character 'u' in their name. Name the column headings "Id", "Name", "Salary" Respectively. Sort the result by staff salary in descending order.

```
SELECT STAFFID "Id", STAFFNAME "Name", STAFFSALARY "Salary"
FROM STAFF
WHERE STAFFSALARY >= 2500 AND STAFFNAME NOT LIKE '%u%'
ORDER BY STAFFSALARY DESC
```

Id	Name	Salary
0003	Faidi Akif	2500
0001	Amin Zaini	2500

QUESTION 2

List staff id, staff name, staff salary, and department id, who are not in department 11

Increase staff salary by 50.5%, express the final value as a whole number. Name the generated column heading to "New salary".

Generate a new column to find how much salary is increased. Name the generated column heading to "Amount Increased"

```
SELECT STAFFID, STAFFNAME, STAFFSALARY, DEPTID,
      ROUND(STAFFSALARY+(STAFFSALARY*50.5/100),0) "New Salary",
      ROUND(STAFFSALARY+(STAFFSALARY*50.5/100),0) - STAFFSALARY
"Amount Increased"
FROM STAFF
WHERE DEPTID <> 11
```

STAFFID	STAFFNAME	STAFFSALARY	DEPTID	New Salary	Amount Increased
0003	Faidi Akif	2500	12	63125	60625
0004	Syazwan Fikri	2000	13	40400	38400
0005	Siti Sarah	1500	10	22725	21225

3 rows returned in 0.01 seconds Download

QUESTION 3

List product id, product name, product type and product price for product type mod or flavour only. Format the product price to appear in the format similar to “$99,999.00”. Name product price column to “Price”.

Create a price that gives a sale to the product price of 20.8%. Display the value as whole number. Format the generated column to appear in the format similar to “$99,999.00”. Name the generated column heading to “Sale”.

```
SELECT PRODID, PRODNAME, PRODTYPE, TO_CHAR(PRODPRICE,
      '$99,999.00')    "Price",
       TO_CHAR(ROUND( (PRODPRICE - PRODPRICE*20.8/100),0),
      '$99,999.00')    "Sale"
FROM PRODUCT
WHERE PRODTYPE IN ('Mod','Flavour')
ORDER BY PRODPRICE DESC
```

PRODID	PRODNAME	PRODTYPE	Price	Sale
30014	FreeMax Pod	Mod	$100.50	$80.00
30015	VaporFi Pod	Mod	$95.90	$76.00
30013	Smok Pod	Mod	$80.80	$64.00
30001	Juicy Mango	Flavour	$35.00	$28.00
30004	Milo Tabur	Flavour	$35.00	$28.00
30005	Creamy Vanilla	Flavour	$35.00	$28.00
30002	Juicy Grape	Flavour	$30.00	$24.00
30003	Juicy Lychee	Flavour	$30.00	$24.00
30006	Popcorn Boom	Flavour	$25.00	$20.00
30008	Minty Salak	Flavour	$20.00	$16.00
30007	Minty Watermelon	Flavour	$20.00	$16.00
30009	Minty Mentos	Flavour	$20.00	$16.00

QUESTION 4

The manager wants to view customer id, customer name, product ID and product name that customer bought for an order with any product that contains letter J. Change the product name column to 'Fast Selling'.

```
SELECT O.CUSTID, C.CUSTNAME, OP.PRODID, P.PRODNAME AS "Fast
Selling" FROM ORDER_PRODUCT OP
JOIN PRODUCT P
ON OP.PRODID = P.PRODID
JOIN ORDERS O
ON OP.ORDERID = O.ORDERID
JOIN Customer C
ON O.CUSTID = C.CUSTID
WHERE OP.PRODID IN (
SELECT PRODID FROM PRODUCT
WHERE PRODNAME LIKE '%J%'
)
```

CUSTID	CUSTNAME	PRODID	Fast Selling
10007	Fayyad Bashir	30001	Juicy Mango
10003	Aina Farhana	30002	Juicy Grape

2 rows returned in 0.01 seconds Download

QUESTION 5

List customer name and the product bought for order 1 and 2. Display the customer's name, order ID and the product name by using the set operators.

```
SELECT DISTINCT C.CUSTNAME, OP.ORDERID, P.PRODNAME FROM
ORDER_PRODUCT OP
JOIN PRODUCT P
ON P.PRODID = OP.PRODID
JOIN ORDERS O
ON OP.ORDERID = O.ORDERID
JOIN CUSTOMER C
ON C.CUSTID = O.CUSTID
WHERE OP.ORDERID = 1

UNION ALL

SELECT DISTINCT C.CUSTNAME, OP.ORDERID, P.PRODNAME FROM
ORDER_PRODUCT OP
JOIN PRODUCT P
ON P.PRODID = OP.PRODID
JOIN ORDERS O
ON OP.ORDERID = O.ORDERID
JOIN CUSTOMER C
ON C.CUSTID = O.CUSTID
WHERE OP.ORDERID = 2
```

CUSTNAME	ORDERID	PRODNAME
Ahmad Zahir	1	Creamy Vanilla
Ahmad Zahir	1	Batteries
Ahmad Zahir	1	Minty Salak
Ahmad Zahir	2	Cosmo Coil
Ahmad Zahir	2	Minty Salak

5 rows returned in 0.03 seconds Download

QUESTION 6

List the five-customer name with the highest number of orders made at the store. Show the column as 'Top Customers'

```
SELECT C.CUSTNAME AS "TOP CUSTOMERS" FROM CUSTOMER C
WHERE C.CUSTID IN (SELECT CUSTID FROM ORDERS O
GROUP BY O.CUSTID
ORDER BY COUNT(O.CUSTID) DESC
FETCH FIRST 5 ROWS ONLY)
```

TOP CUSTOMERS
Ahmad Zahir
Fayyad Bashir
Aina Farhana
Fazlina Rusdina
Nur Amalina

5 rows returned in 0.01 seconds Download

QUESTION 7

List department id, department name, and total salary per department. Concatenate department name and total salary and place ' department is RM' in the middle and rename the column header to "Total Salary per department". Sort the list by department id in ascending order.

```
SELECT S.DEPTID , D.DEPTNAME || ' department is RM' ||
SUM(S.STAFFSALARY) "Total Salary per department"
FROM STAFF S JOIN DEPARTMENT D ON S.DEPTID = D.DEPTID
GROUP BY S.DEPTID, D.DEPTNAME
ORDER BY DEPTID
```

DEPTID	Total Salary per department
10	Sales department is RM1500
11	Marketing department is RM5500
12	Management department is RM2500
13	Finance department is RM2000

4 rows returned in 0.01 seconds Download

QUESTION 8

Find product id, product name, minimum inventory and total sales that has accumulate over time with the lowest minimum inventory. Rename the generated column to “Total Sales”.

```
SELECT P.PRODID, P.PRODNAME, P.MININV, SUM(O.SALESQUANTITY) "Total
Sales"
FROM ORDER_PRODUCT O JOIN PRODUCT P ON P.PRODID = O.PRODID
WHERE MININV = (SELECT MIN(MININV) FROM PRODUCT)
GROUP BY P.PRODID, P.PRODNAME, P.MININV
```

PRODID	PRODNAME	MININV	Total Sales
30010	Cosmo Coil	1	3

QUESTION 9

List customer id and name, who are members and less than 22 years old as of the current date.

```
SELECT CUSTID, CUSTNAME
FROM CUSTOMER
WHERE CUSTID IN (SELECT CUSTID FROM MEMBER WHERE ROUND((SYSDATE -
TO_DATE(MEMDOB))/365.25,0) < 22 )

/*OBTAINED ON 28TH JUNE 2021*/
```

CUSTID	CUSTNAME
10000	Arfan Zaini
10001	Syahirah Ros
10002	Naqeeb Rizal

QUESTION 10

List product id, product name and product cost that have greater than the average cost. Rename the product cost heading to “Above average cost”.

```
SELECT PRODID, PRODNAME, PRODCOST “Above average cost”
FROM PRODUCT
WHERE PRODCOST > (SELECT AVG(PRODCOST) FROM PRODUCT)
```

PRODID	PRODNAME	Above average cost
30011	Batteries	28.2
30013	Smok Pod	69.2
30014	FreeMax Pod	75.5
30015	VaporFi Pod	65.5

4 rows returned in 0.01 seconds Download

References

List of references that have been involved while completing the proposal report are listed as below.

For Website:

1. Provapesemabok. (2020, Jun 24). *Home* [Facebook page]. Retrieved from May 17, 2021, from https://en-gb.facebook.com/provapesemabok/
2. PROVAPE OFFICIAL [@provapesemabok]. (n.d). Posts [Instagram profile]. Retrieved May 17, 2021, from https://www.instagram.com/provapesemabok/?hl=en

For Book:

Coronel, C., Morris, S., & Rob, P. (2010). Database systems. *Design Implementation and Management*.

For Interview

Mr Mustaqim Bin Anuar, Manager, May 7. 2021.

Appendices

Figure 3.0: Screenshot of interview session

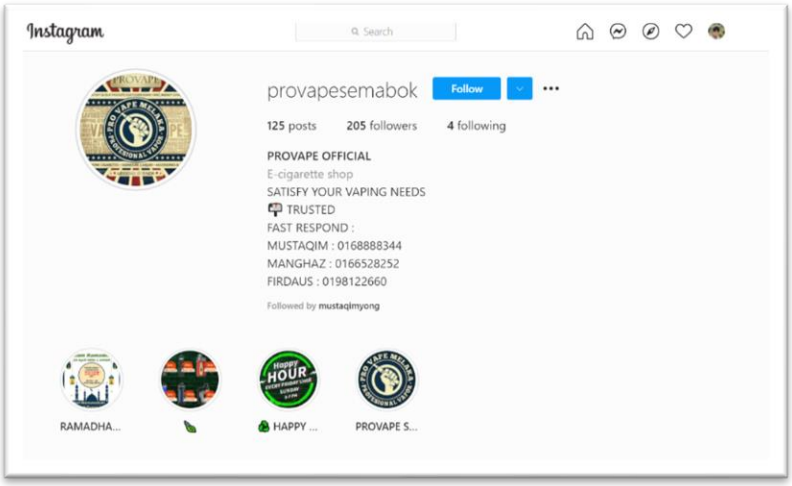


Figure 4.0: Screenshot of Instagram page

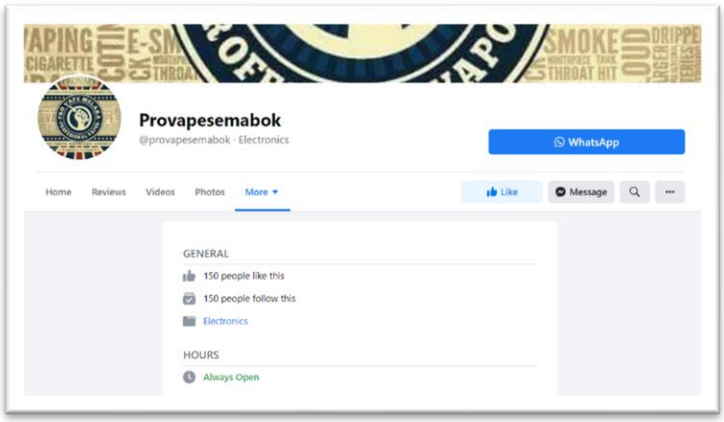


Figure 5.0: Screenshot of facebook page