**ABSTRACT**

**OBJECTIVE**

**FEATURES OF PYTHON USED**

**ER DIAGRAM**

**CHAPTER 1**

**INTRODUCTION**

“Sales Made Easy” is a project which depicts the recommendation process in any OTT platform or online shopping that we see today. To say in simple words, the sales of products have been increased effectively only because of the recommending algorithms which showcases the similar products which may be liked by the customer based on their choice. We will have a look at the entire structure of recommendation process and data storage of the customers details and credentials. This project considers mobile showroom as an example to see if the right products are being recommended to right customer.

**1.1 PROBLEM DEFINITION**

In the current scenario there are various technologies evolving at immense speed trying to solve real-time problems. One usual thing we would have noticed is how the online shopping platforms recommends a variety of specific products upon the products chosen by the customer. Usually, customers when on their way to buy a product may be successful in buying it or not for many reasons, one of which, they wouldn’t have come across multiple variants of the product to choose among to experience it even if it was available or in other words they would miss out on products of different brand which they would have liked it but just that they didn’t have an idea about that brand. The right product doesn’t reach out to the right customer’s hand and hence, they miss out on that product.

**1.2 COURSE OBJECTIVES**

In this project we will try to accomplish by finding a solution to the above-mentioned problem. The main objective is to recommend the products with similar features of various brands which can also be liked by the customer based on the user choice of products.

**1.3 METHODOLOGY TO BE FOLLOWED**

This project uses the python and databases concepts to run and maintain the application. We will be implementing the front end using python’s Tkinter and back-end as MySQL database. The necessary inputs for this project are figured out and we will prepare a ER diagrams and schema models to have a better understanding of the application program flow.

**1.4 EXPECTED OUTCOMES**

On successful implementation, this project produces various outcomes as:

* The user can have an experience of several varieties of the product instead of just one or two, by which he/she can take home the worthy and satisfied product.
* The admin can predict the outcome of the best brands of product being sold and can take advantage of it.
* Secure transaction of payment with OTP.
* By this data we can train a machine learning model which can enhance itself to recommend the best suitable products to specific customers based on their likes.

**CHAPTER 2**

**REQUIREMENT SATISFACTION**

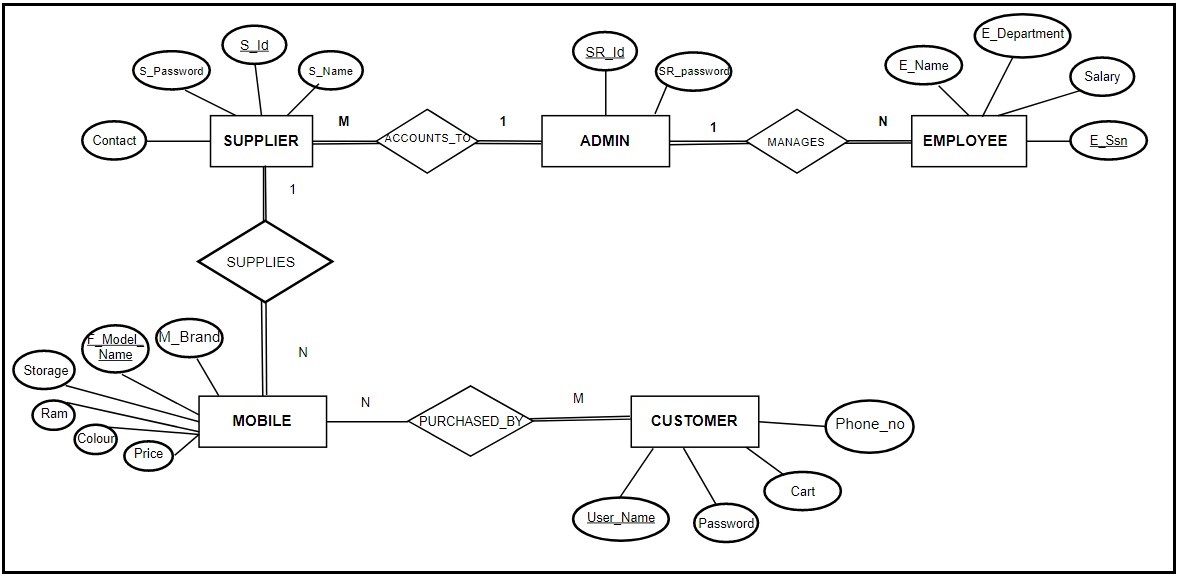
**2.1 HARDWARE REQUIREMENTS:**

* **RAM**: 4GB and above
* **ROM**: Minimum 50GB

**2.2 SOFTWARE REQUIREMENTS:**

* **Operating System**: Microsoft Windows 10
* **Programming Language**: Python
* **Front-end**: Tkinter
* **Back-end**: MySQL Database
* **IDE**: PyCharm
* **Processor**: Processor Intel(R) Core i5 8th Gen

**2.3 ER MODEL**

****

**Fig 2.3.1 ER diagram**

**ASSUMPTIONS**

Project involves 5 entities named SUPPLIER, ADMIN, EMPLOYEE, MOBILE, CUSTOMERwith their respective attributes as mentioned in above diagram.

1. Each supplier is uniquely identified by S\_id and have other attributes like

E\_department, E\_name and salary.

1. Each employee is uniquely identified by E\_ssn and have other attributes

like S\_password, S\_name and contact.

1. Each customer is uniquely identified by User\_name and have other

attributes like Password, cart and Phone\_no.

1. Each mobile is uniquely identified by F\_model\_name and have other

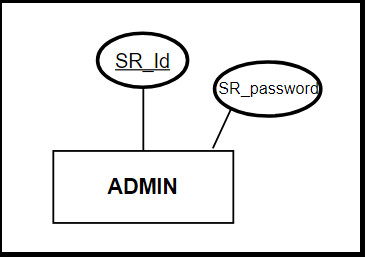
attributeslike M\_brand, Storage, Ram, Colour and Price.

1. Admin has a Username and password.
2. “m” number of mobiles can be bought by “n” number of customers.
3. There can be a mobile which is not bought by any customer.

**CHAPTER 3**

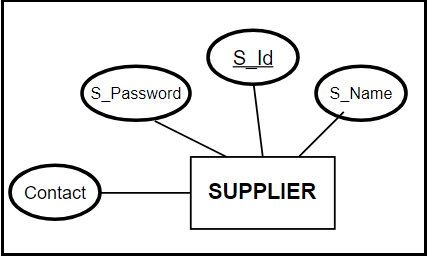
**DATAMODELS/ER MODEL**

**3.1 ENTITY AND ATTRIBUTES**



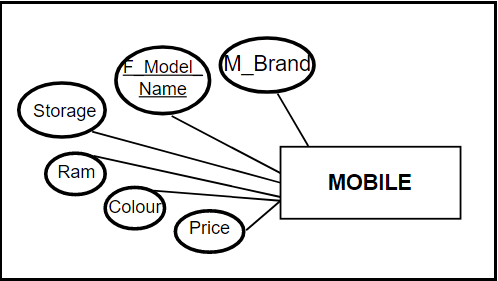
**Fig 3.1.1** Entity ADMIN

Admin is an entity with attributes SR\_Id and SR\_password which represents admin username and password for login. SR\_Id is the primary key which can be changed or updates by the admin when necessary.



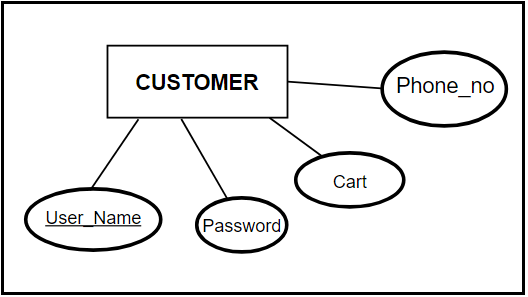
**Fig 3.1.2** Entity SUPPLIER

Supplier is an entity with attributes S\_Password, S\_Id, S\_name and contact which represents supplier password , username, supplier name and contact number for login. S\_Id is the primary key which is unique to every supplier and has to enter it while login.



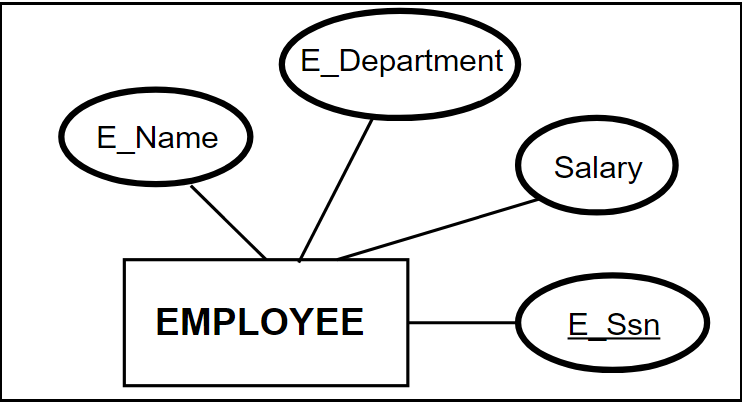
**Fig 3.1.3** Entity MOBILE

Mobile is an entity with attributes M\_Brand, F\_Model\_Name, Storage, Ram, Colour and Price which represents mobile brand, mobile name, storage, Ram, colour and price. F\_Model\_Name is the primary key which is unique to every mobile.



**Fig 3.1.4** Entity CUSTOMER

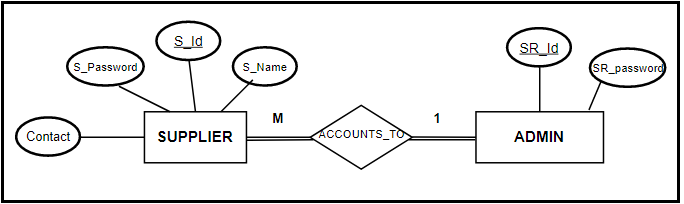
Customer is an entity with attributes User\_name, Password, cart and Phone\_no which represents customer username, password, cart and contact number for login. User\_name is the primary key which is unique to every customer and has to enter it while login. Cart is an attribute in which the customer can view all the purchased mobiles and also can give feedback for the products.



**Fig 3.1.5** Entity EMPLOYEE

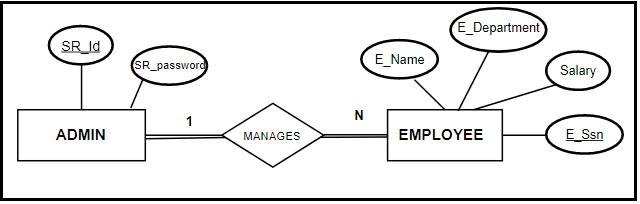
Employee is an entity with attributes E\_Name, E\_department, salary and E\_ssn which represents employee name, department of work, salary and employee number. E\_ssn is the primary key which is unique for every employee which is generated by the application when the admin adds a new employee.

**3.2 RELATIONSHIP AND PARTICIPATION**



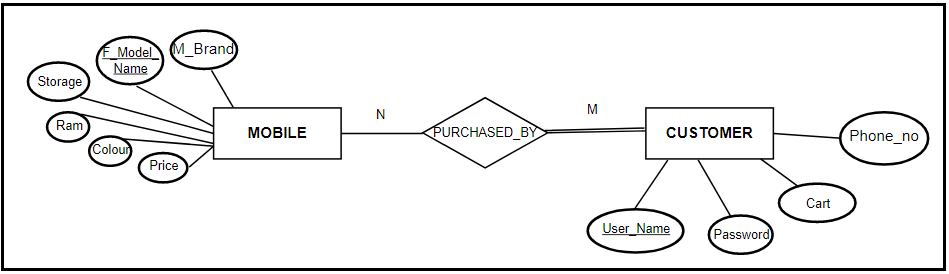
**Fig 3.2.1 Supplier Admin relationship**

The above figure represents a strong entity relationship between supplier and admin which is represented as “**accounts\_to**”. The role of supplier is to supply/register new mobiles to the admin and participation ration being M:1 resembles that there can be M number of suppliers supplying to only one admin. The participation of both the entities in the relation is considered to be FULL.



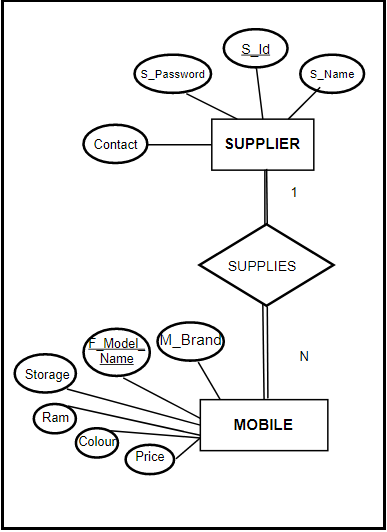
**Fig 3.2.2 Admin Employee relationship**

This is a strong entity relationship between employee and admin which is represented as “**manages**”. Participation ration being M:1 resembles that there can be N number of employees being managed by only one admin. The participation of both the entities in the relation is considered to be FULL.



**Fig 3.2.3 Mobile Customer relationship**

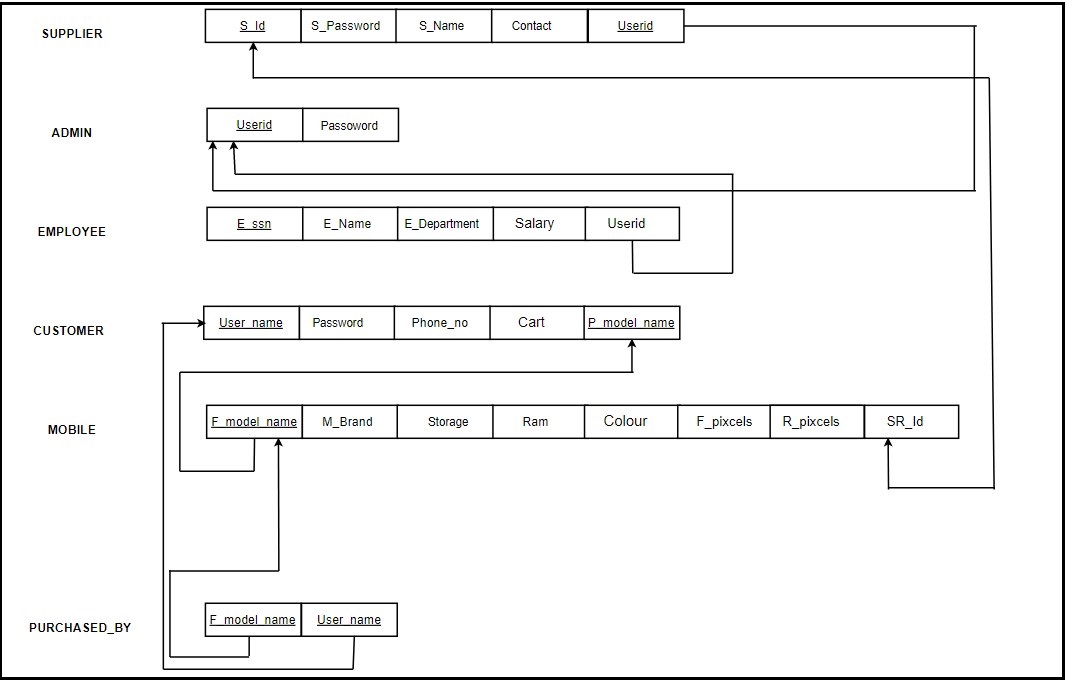
The above figure represents a strong entity relationship between supplier and admin which is represented as “**purchased\_by**”. Participation ration being M:N resembles that there can be N number of mobiles being purchased by M customers but there is a partial participation of mobile entity in the relation as there can be a mobile which is not bought by any customer. On the other side there is FULL participation of customer entity because there can not be any phone which is sold without it bought by a customer.



**Fig 3.2.4 Supplier Mobile relationship**

The above figure represents a strong entity relationship between supplier and mobile which is represented as “supplies”. The role of supplier is to supply/register new mobiles by specifying the features and quantity. Participation ration being 1:M represents that there can be only one supplier supplying N number of mobiles of same brand. The participation of both the entities in the relation is considered to be FULL.

**3.3 RELATIONAL SCHEMA**

****

**Fig 3.3.1 Relational schema**

Since the relational between mobile and customer is “M:N”, a new table is created in the name “PURCHASED\_BY” which consist of attributes that are primary key of the two entities selected

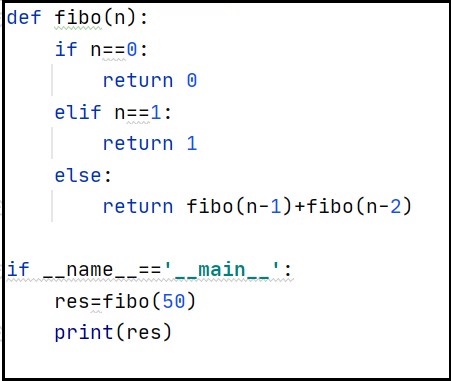
**3.4 PYTHON FUNDAMENTALS**

Python is a high-level dynamically typed programming language for general-purpose and very popular among developers. Another add on for python language is it is Object-oriented and is a muti-paradigm programming language. Python is immensely used in the field of web development, machine learning, Image processing and many more. Python is one of the simplified programming languages when compared to all of its parent languages. By this we mean to say that, lines of codes written in python is way lesser than other languages like java.

**Some fundamental topics in python:**

**Statements:** In python, statements are nothing but logical instructions that interpreter can read and execute. For example: ‘if’, ‘for’, ‘while’, ‘a==10’ all are statements which can be executed by the interpreter.

**Indentation:** Almost all the programming languages uses braces {} to indicate a block of code unlike python, which uses indentation to mark a block of code. Indentation are nothing but a block of spaces separating code from the margin.

Example: 

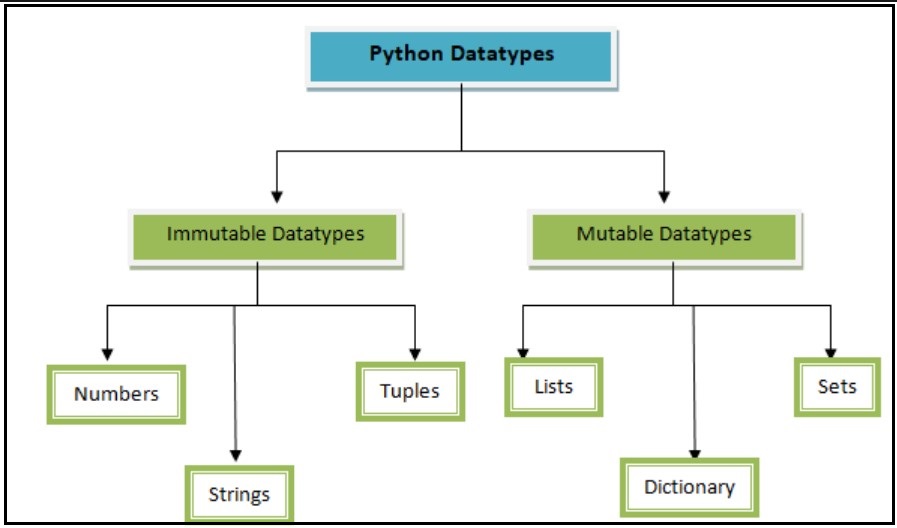
**Fig 3.4.1 Indentation example**

**Comments:** Comments are nothing but tagged lines of words in codes which increases the readability of the code and gives a better understanding for the reader.

**Variables:** A variable is a memory address which can be assigned values that can change and when memory address cannot change then it is known as constants. A variable can store different types of values. For example, students name should be stored as string whereas students USN should be stored as integer.

**PYTHON DATATYPES**

Python provides various standard data types that define the storage method on each of them.



**Fig 3.4.2 Python data-types**

**Number:** In Python, number data types are used to store numeric values. There are four different numerical types in Python:

* **int**: this one is pretty standard — plain integers are just positive or negative whole numbers.
* **long**: long integers are integers of infinite size. They look just like plain integers except they’re followed by the letter “L” (ex: 150L).
* **float**: floats represent real numbers, but are written with decimal points (or scientific notation) to divide the whole number into fractional parts.
* **complex**: represented by the formula a + bJ, where a and b are floats, and J is the square root of -1 (the result of which is an imaginary number). Complex numbers are used sparingly in Python.

**Strings:** String data-types are one of the most commonly used to store a value. Strings are initialized by just enclosing the value in quotes. It is represented by the **str** class in Python. We can use the **str ()** function to convert other data types to **str type**.

For example: title=’Sales Made Easy’ # title is a variable which stores string

**Tuple**: A tuple is a collection of objects which are ordered and immutable. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets. Creating a tuple is as simple as putting different comma-separated values.

For example:

tup1= ( ‘Sales’ , ’Made’ , ’Easy’ )

tup2= ( 201 , 202 , 203 , 204 )

**List**: List data-types are one among the mutable types which are commonly used to store data. A list contains elements separated by comma similar to tuples but enclosed with ‘[]’ brackets. One huge advantage of using list data-types is that all the items belonging in that list can be of different data type.

The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (\*) is the repetition operator.

For example:

List1= [‘Sales’, 5 , ‘made’ , ‘Easy’ , 10 , 493]

List2= [‘hello’ , ‘world’]

List3 = [ 123 , 234 , 456]

**Dictionary**: **Dictionary**in is an unordered collection of data values which is used to store data values like a map, which unlike other data-types that hold only single value as an element. Other key note for dictionary is that it holds key : value pair where each key represents or points to one value. Key value is provided in the dictionary to make it more optimized. Values in a dictionary can be of any datatype and can be duplicated, whereas keys can’t be repeated and must be immutable.

Dictionary can be created by placing sequence of elements within curly **{}** braces, separated by ‘comma’.

For example:

dictionary1 = {1: 'Sales', 2: 'Made', 3: 'Easy'}

dictionary1 = {'Title': 'Sales', 1: [1, 2, 3, 4]}

**Sets**: Sets in Python is an unordered collection of data which are mutable and has no duplicate elements. Python’s set class represents the mathematical notion of a set. The major advantage of using a set, as opposed to a list, is that it has a highly optimized method for checking whether a specific element is contained in the set. Since sets are unordered, we cannot access items using indexes like we do in [lists](https://www.geeksforgeeks.org/python-list/).There are numerous function that can operate on sets. Some of which are add, intersection, difference, clear and etc.

For example:

Set1 = {1,2,3,4}

Set2 = { “Sales” , “Made” , “Easy” }

Set1.add (6) output: set1= {1,2,3,4,6}

Set2.clear() output: set2={}

**3.5 DATABASE/ALGORITHM FUNDAMENTALS**

A database is a collection of data which is organized, which is also called as structured data. It can be accessed or stored at the computer system. It can be managed through Database management system (DBMS), which is a software which is used to manage data. Database refers to related data which is in a structured form.

In Database, data is organized into tables which consist of rows and columns and it is indexed so data gets updated, expanded and deleted easily. Computer databases typically contain file records data like transactions money in one bank account to another bank account, sales and customer details, fee details of student and product details. There are different kinds of databases like the relational database, distributed database, cloud database, NoSQL database and much more.

**RELATIONAL DATABASE**

A relational database is used to store and provide access to data points that are related to one another. It is based on relational model that are used to represent data in tables. In relational database, each row in a table is a record with a unique ID called key. The columns of the table hold attributes of the data and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

Each table in a relational database contains [rows](https://techterms.com/definition/row) also known as records and [columns](https://techterms.com/definition/column) also known as fields. Technically rows are sometimes called "tuples," columns may be referred to as "attributes," and the tables themselves may be called "relations.". It is "relational" since all records share the same fields.

Almost all the tables in a database include a [primary key](https://techterms.com/definition/primarykey), which acts as a unique identifier for each row within the table. This key value cannot have duplicate value. The key may be assigned to a column, or it may be comprised of multiple columns that together form a unique combination of values. When there is a relation between two tables then we represent or link it with another key called as foreign key. Through foreign key we can access or process the other table with the help of foreign key.

The standard way to access data from a relational database is through an [SQL](https://techterms.com/definition/sql) (Structured Query Language) query. SQL queries are classified into four types:

* DDL (Data Definition Language): creation of objects
* DML (Data Manipulation Language): manipulation of data
* DCL (Data Control Language): assignment and removal of permissions
* DQL (Data Query Language)

**DDL Commands:** It consists of SQL commands that can be used to define the database schema. It is used to describe the database schema and also involves in creating and modifying the structure of database objects.

* **CREATE:** This command is used to create a database or its objects like table, views and triggers.
* **DROP:** Used to delete objects from a database.
* **ALTER:** This command is used to alter the structure of a database.
* **TRUNCATE:** Removes all the data r records which are stored in the database including all spaces allocated to records.
* **RENAME:** this command is used to rename the objects exciting in a database.

**DML Commands:** These commands usually deal with manipulation of records in a database table. It doesn’t involve in anything like change of database object or its structure.

* **INSERT:** Used to insert data into tables
* **UPDATE:** Used to update the value of an existing data in tables.
* **DELETE:** Used to delete records from a table.

**DCL commands:** These commands commonly deal with the rights, permissions and other controls of the database system.

* **GRANT:** It give user the right to access privileges the database.
* **REVOKE:** It is the command used to withdraw the user access privilege given to them by GRANT.

**DQL Commands:** When use database not just to store data or records but even to retrieve data based on the query statement. Hence DQL commands come in handy for this purpose where we can get some schema relation based on the query passed to it.

* **SELECT:** This command is used to retrieve records from database according to a specified query.

**CHAPTER 4**

**DESIGN**

**4.1 DESIGN GOALS**

In the process of developing this application we are making use of various concepts provided and supported by PYTHON. Some of that majorly used python concepts are lists, sets, tuples, functions and strings. Apart from this python's Tkinter module (package) and its widgets are more prominently used fro front-end to enhance the user experience and interaction with the application. At some instances it also deals with handling exceptions and returning the appropriate exception message.

This project imitates the functioning of online shopping websites like which it has the same admin, user, employee and several other portals. The admin portal consist of various sections which overall has access to all the functionalities of the application. Similarly, it has customers and supplier’s portal which consist of the necessary details and inputs for the user to shop and supplier to register a mobile. Customers cart section is used to keep track of their purchases. Apart from this the user can give a feedback in form of ratings for the mobile they purchased and according the application recommends few mobiles to the next user. Users has 5 various options to sort the mobiles and choose their best of choice mobiles.

**4.2 ALGORITHM/ PSEUDOCODE**

**1. Home page for the application:**

Step 1: Start

Step 2: Create a root window for the home page of the application which displays the appropriate login buttons for admin, customer and supplier.

Step 3: When clicked on the admin login button or customer button or supplier button it corresponds to respective portal page.

Step 4: END

1. **Admin page for the application:**

Step 1: Start

Step 2: Establish DBMS connection.

Step 3: Create admin\_login page() for admin to enter username and password.

Step 4: After successful login admin portal consist of “admin section”, “employee section” and “production section”. Admin section consist of change\_username() under which username name of admin can be changed using old username and change\_password() function consist of code to change password using old password.

Step 5: Employee section has add\_employee() function which executes the code add new\_user() and generate unique ESSN for the employee as well as delete\_employee().

Step 6: Employee section also consist of search\_employee() function which executes code to search for a particular employee as well as list all the employees registered.

Step 7: Production section has list\_of\_mobiles() which displays list of all mobiles registered by the supplier.

Step 8: Log Out

Step 9: END

1. **Customer page for the application:**

Step 1: Start

Step 2: Establish DBMS connection.

Step 3: Create customer\_login page() for customer to enter username and password.

Step 4: If user not registered then an option is created to register a new customer with OTP authentication of the user.

Step 5: After successful login/registration customer portal consist of “shopping section”, “billing section” and “rating section”. Shopping section is created with 5 options among which the customer can choose and filter their choice.

Step 6: After selecting a mobile the proceed to bill function executes a code to proceed to final billing section.

Step 7: After confirming the billing the product is purchased and queued in the cart.

Step 8: In the shopping history() function the tree-view data-type is used to display the mobiles purchased by the respective customer.

Step 9: In the rating section, rating() function is used to execute a code where the user can give a feedback rating for the mobile purchased.

Step 10: Log Out

Step 11: END

1. **Supplier page for the application:**

Step 1: Start

Step 2: Establish DBMS connection.

Step 3: Create supplier\_login page() for supplier to enter username and password.

Step 4: After successful login supplier portal pops up and the supplier can register a mobile by entering the necessary features and quantity.

Step 5: After entering valid details the products gets added successful.

Step 6: Log Out.

Step 7: END

**CHAPTER 5**

This project involves 3 main sections through which all the user interface happens.

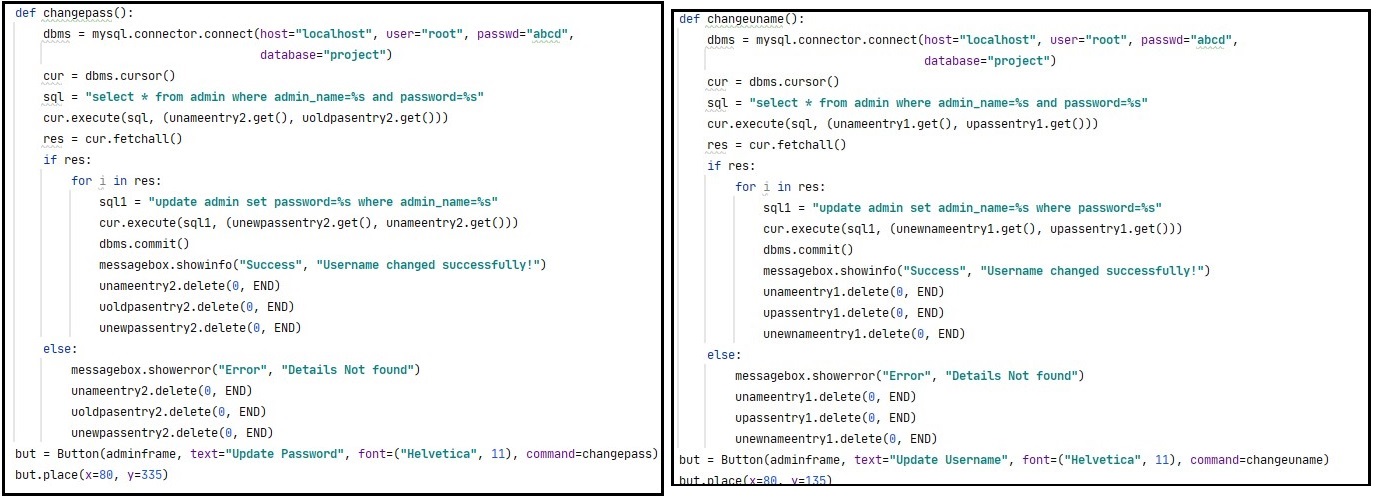
These 3 sections are namely Admin login, Customer login and Supplier login.

**Module 1 Functionality:**

****

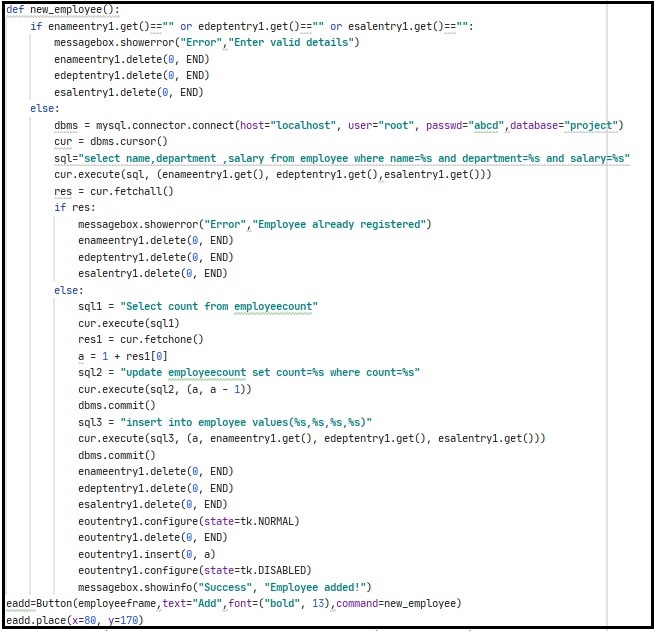
**Fig.5.1.1 Admin login**

Admin is a function which is designed to hold all the necessary information for the admin portal. This function is implemented with different options for admin such as the changing username and password, add ,delete and search for employee and finally list of all the mobiles registered. In the login page when the admin enter his username and password he will be directed to the next page of the portal.



**Fig.5.1.2 Change username or password (admin)**

The above two functions defined as changepass and changename are implemented for the use of admin if in case he has to change his username or password.



**Fig.5.1.3 Add employee (admin)**

The above function new\_employee helps in adding a new employee to the organization with the necessary details like name, department, salary and ESSN for the respective employee will be generated by the application based on the previously registered employee.



**Fig.5.1.4 Delete employee (admin)**

The above function del\_employee is implemented to delete an employee from the organization database. This function accepts an input from admin i.e, ESSN of the employee. On an entry of invalid ESSN the application throws an error saying “Invalid ESSN”.



**Fig.5.1.5 List of mobiles registered (admin)**

The above few set of lines are implemented to list the set of mobiles registered by the organization. These mobiles updated as and when the supplier logs In and supplies mobile.

Finally, the admin has an option to securely LOG OUT of his portal.

**Module 2 Functionality:**

****

**Fig.5.2.1 Customer login**

Customer is a function which is designed to hold all the necessary information for the customer portal. This function is implemented with different options for customer such as the login, shopping page, cart, feedback and rating. In the login page when the customer enter their valid username and password then they will be directed to the next page of the portal. If he/she is new then they will have to register first with a valid phone number along with OTP authentication.

****

**Fig.5.2.2 Options for choosing mobile.**

Above are the set of functions which are the set of recommending options which suggests the user to pick the best mobile of their choice. There are 5 options i.e, Latest, price(low-high), 4GB RAM, 6GB RAM and the most bought.

* Latest: This function is implemented in a way such that it displays a set of mobiles based on its date of release.
* Price (low-high): This function is implemented in a way such that it displays all the mobiles in increasing order of price.
* 4GB and 6GB RAM: These functions are implemented in a way such that it displays set of mobiles with 4 GB or6GB RAM.
* Most Bought: This function is implemented in a way such that it displays mobiles which are bought by most of their customers.

Finally when the customer has chosen a mobile they can continue to billing where he/she can purchase the mobile.



**Fig.5.2.3 Shopping history**

After purchasing the product, the products will be added to the customers shopping history. Here the customer can have a view of all the mobiles that they have purchased in the past.

By this, customers have an option to rate the purchased products based on their experience with product and satisfaction.

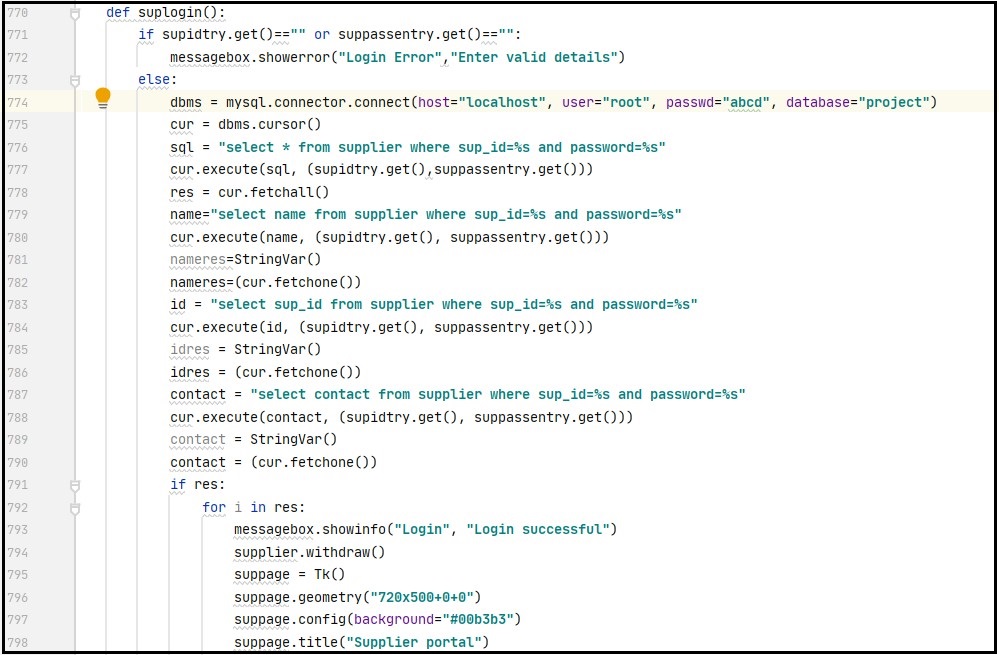
****

**Fig.5.2.4 OTP code**

This is code which is implemented as a function for executing the OTP authentication.

When a customer registers for the first time then they have to enter a valid phone number for which they will receive an OTP. Only when they enter the correct OTP they will be registered as a valid customer**.**

**Module 3 Functionality:**

****

**Fig.5.3.1 Supplier login**

Supplier is a function which is designed to hold all the necessary information for the supplier portal. This function is implemented in a way such that the supplier can register a mobile with all its necessary features along with it. Supplier login is same as others where he/she has to enter valid credentials in order to login.

****

**Fig.5.3.2 Adding mobile to organization**

This is the supplier section where all the mobiles are registered for the organization.

Supplier after logging in he/she has to enter all the necessary features of mobile mentioned in the form and then submit it along with the quantity.

If there is an invalid entry for any of the section the application will throw an error message saying “Invalid details”.

**CHAPTER 6**

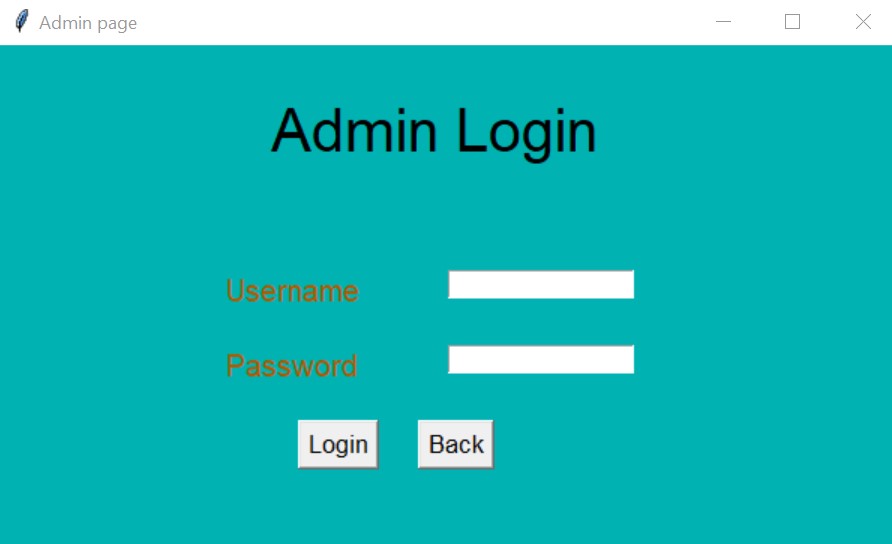
**RESULTS**

Below are the outputs for the module wise implementation of the project.



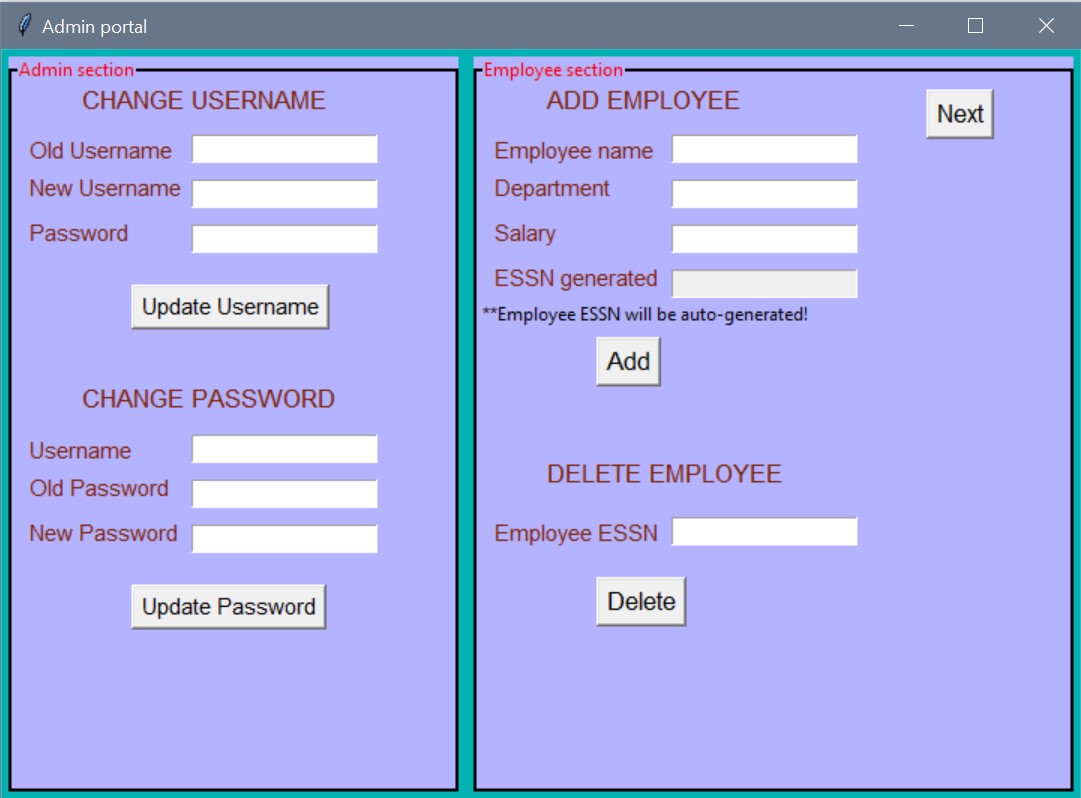
**Fig.6.1 Front page**

This is the opening page of the application which displays the 3 login buttons admin, customer and supplier login respectively.



**Fig.6.2 Admin login**

When clicked on admin button the above page pops up and when the admin enters the right credentials he/she will be directed to the admin portal.

****

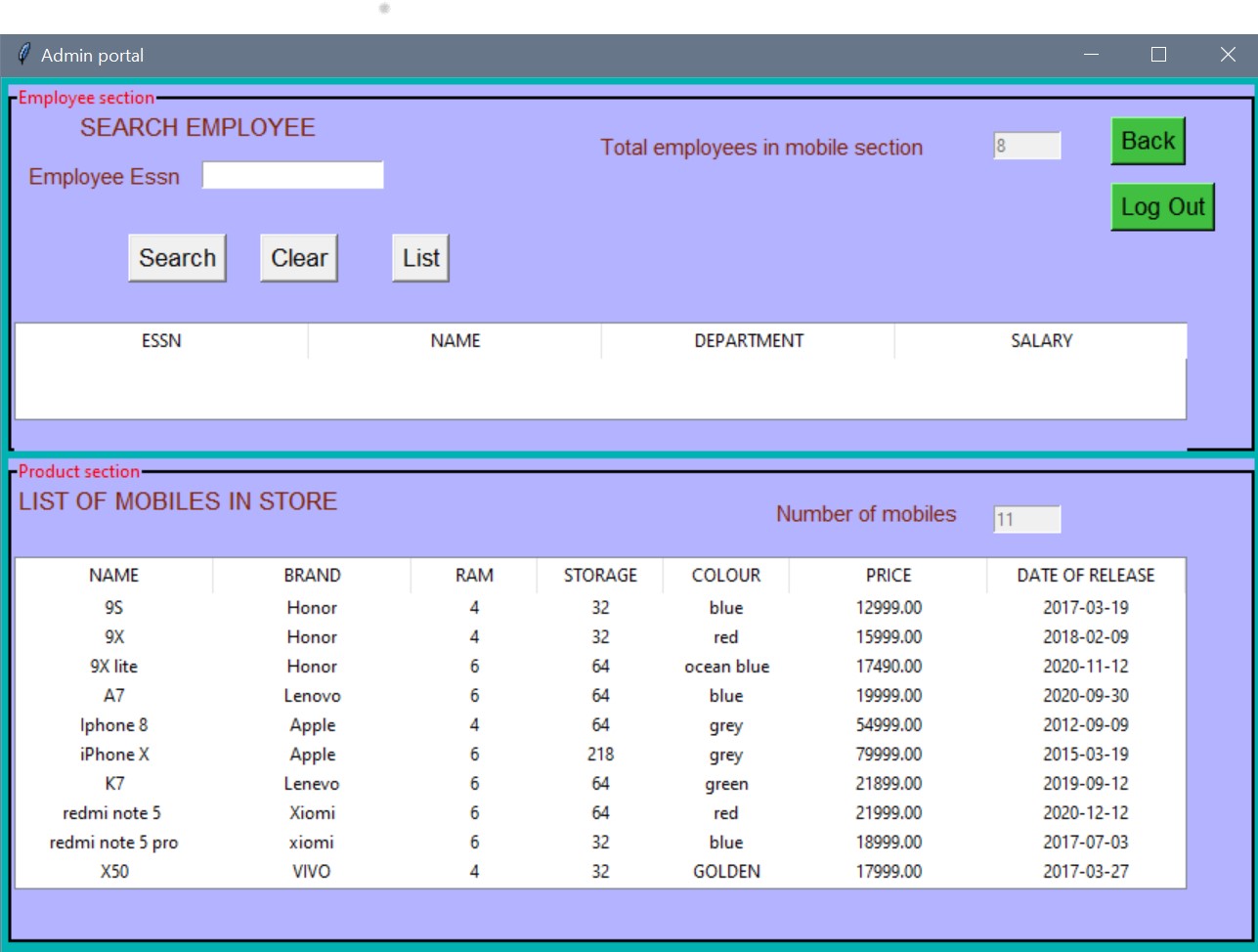
**Fig.6.3 Admin portal page 1**

Module 1 functionality deals with the illustrations of admin portal. Admin portal consist of several options to view and manage the necessary data, by data we mean information of customer, products, employee and admin themselves.

In the admin section, the admin has an option of either changing their existing username or password.

In the employee section, the admin has an option of adding or deleting employee details.

When the admin adds a new employee their ESSN is automatically generated based on the previously registered employee

****

**Fig.6.4 Admin portal page 2**

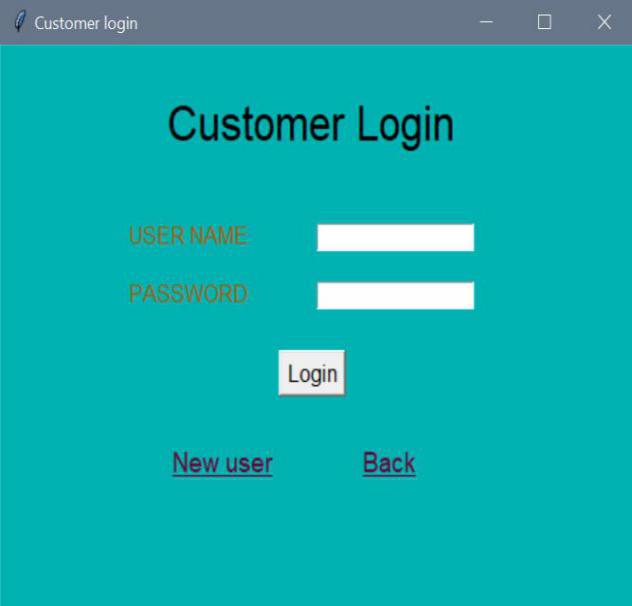
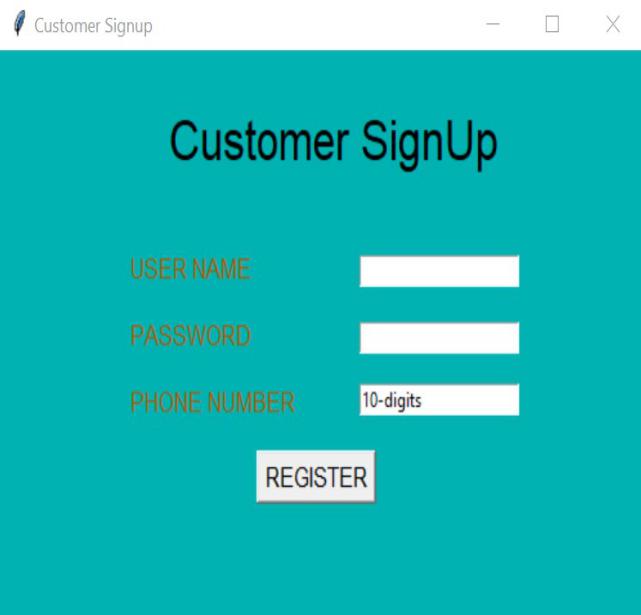
In the page 2 of admin portal, admin can search for a particular employee who have registered with the company. Admin can also see the list of all registered employees with their details.

In the production section admin can view all the mobiles registered by the supplier with all its feature. When a mobile is registered by the supplier this section of mobile list gets updated and gives a full list based on the data in database.

Similarly the admin can view the count of total number of employees registered as well as the total count of mobiles. The Tkinter widget used for viewing the list of items is “Tree-view”.

Finally, admin can securely log out after all the necessary check and data retrieval.

Module 2 functionality deals with the illustrations of customer portal. This module will completely explain how users can interact with the application and buy products.

**Fig.6.5 Customer login and sign-up**

The above figure is a login page for customers and a sign-up page for customers who haven’t registered. This includes a OTP authenticated sign-up where the customer has to enter a valid phone number in order to receive a OTP and only when they enter the OTP they will be registered and be considered as a valid customer.



**Fig.6.6 Shopping portal**

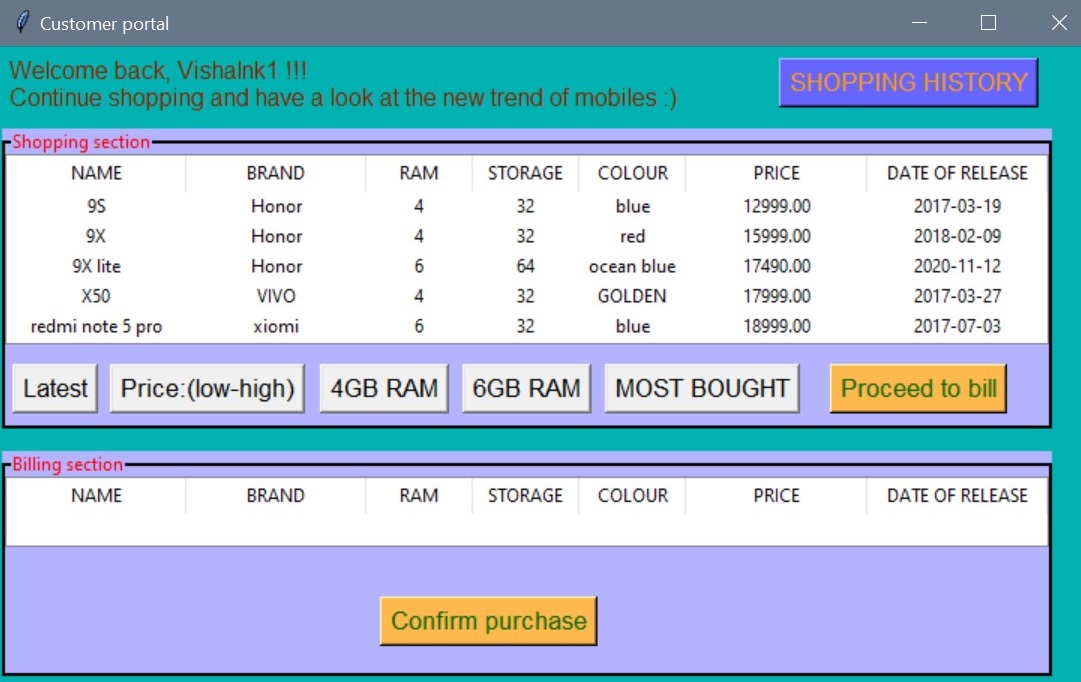
As soon as the user log’s-in with their username and password then a page as shown in the above figure shows up. This portal guides the user to purchase mobile of different brands based on certain inputs from the user.

There are 5 options with which it can help the user to select a mobile of their suitable choice.



**Fig.6.7 Mobiles based on latest release dates**

The above figure shows the set of mobile collection which has the latest release dates. From this option users can get an idea about the phone/brand in current trend or release. Latest the release dates of mobile, best will be its features. Features can be in terms of processor, performance, price and many more.



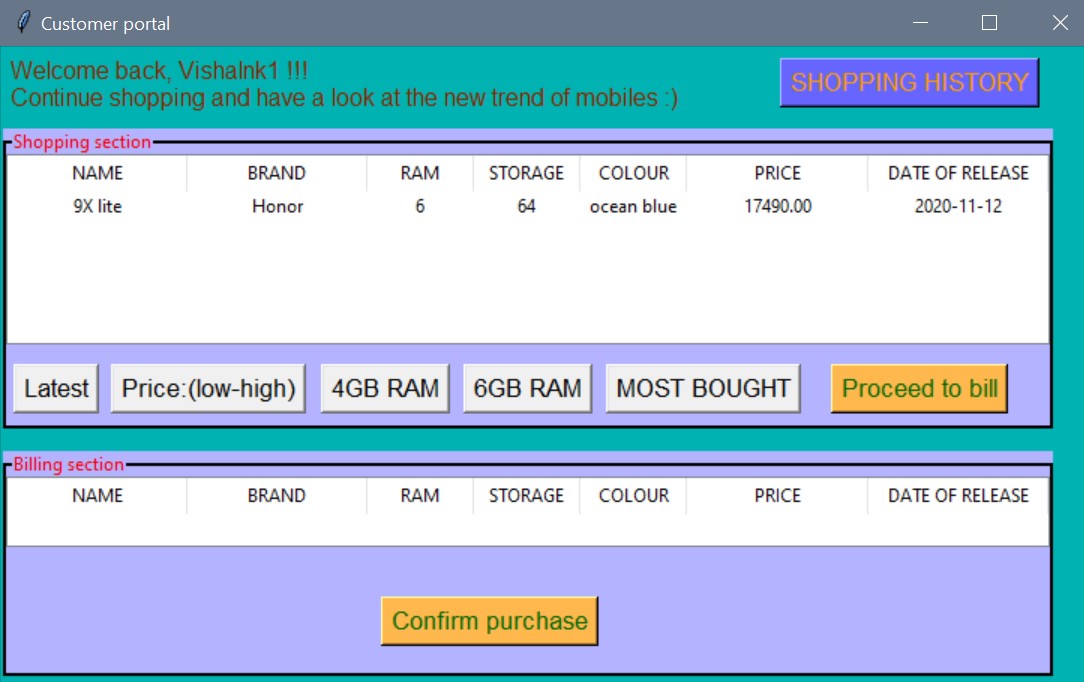
**Fig.6.8 Mobiles based on price (low-high)**

The above figure shows the set of mobile collection in ascending order of their price.

****

**Fig.6.9 Mobiles with 4GB or 6GB RAM**

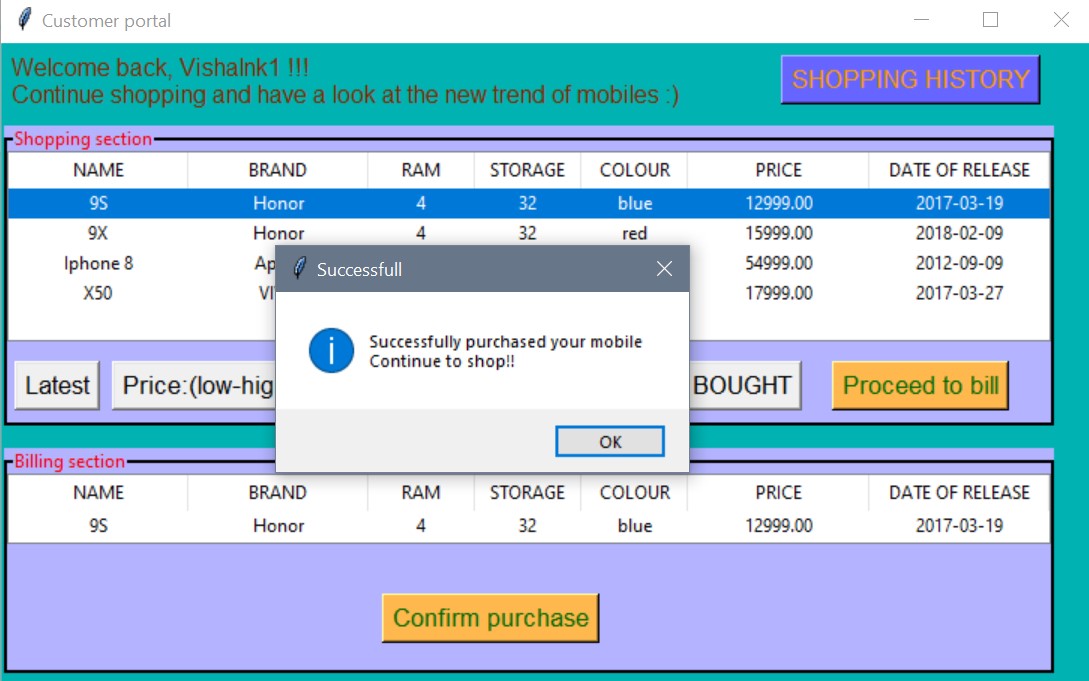
The above figure shows the set of mobile collection which has the feature of 4GB or 6GB RAM.



**Fig.6.10 Mobiles most bought**

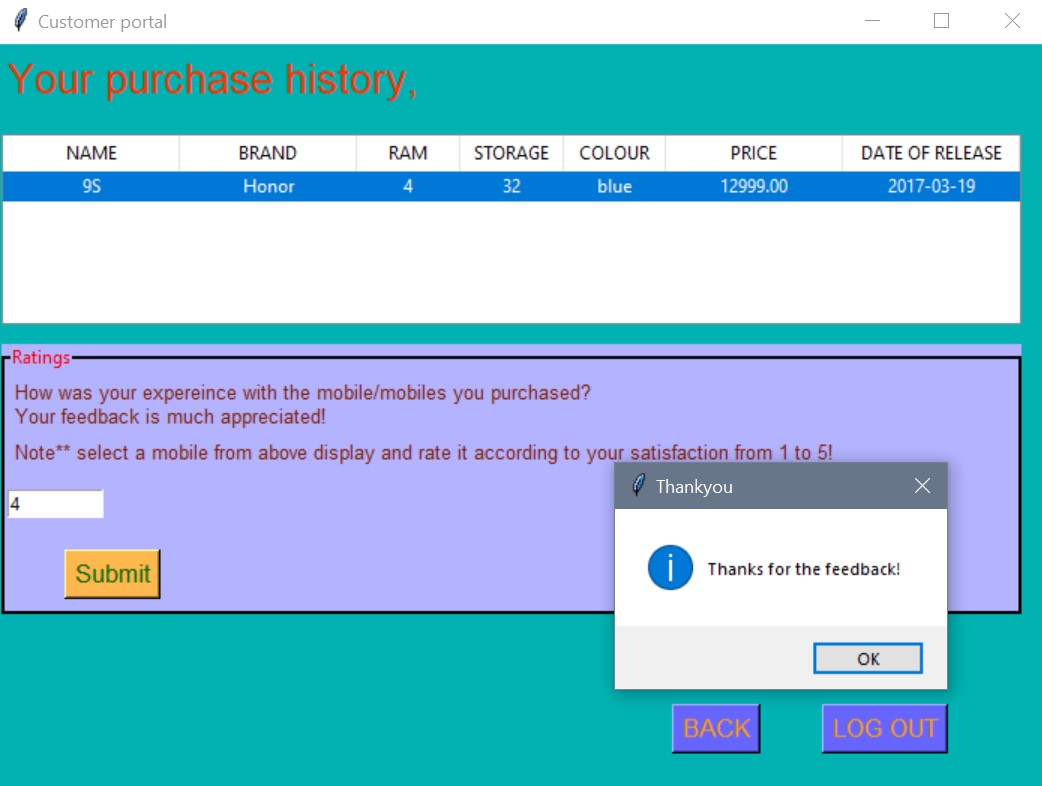
The above figure shows the set of mobiles which are bought by most of the customer.

From this option.



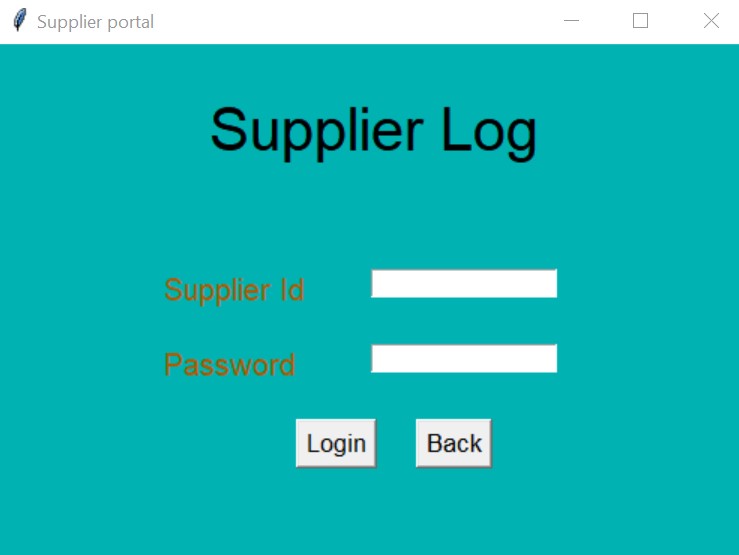
**Fig.6.11 mobile billing**

When the user selects a mobile and proceeds to bill he/she can confirm the purchase and the product will be added to the shopping history.



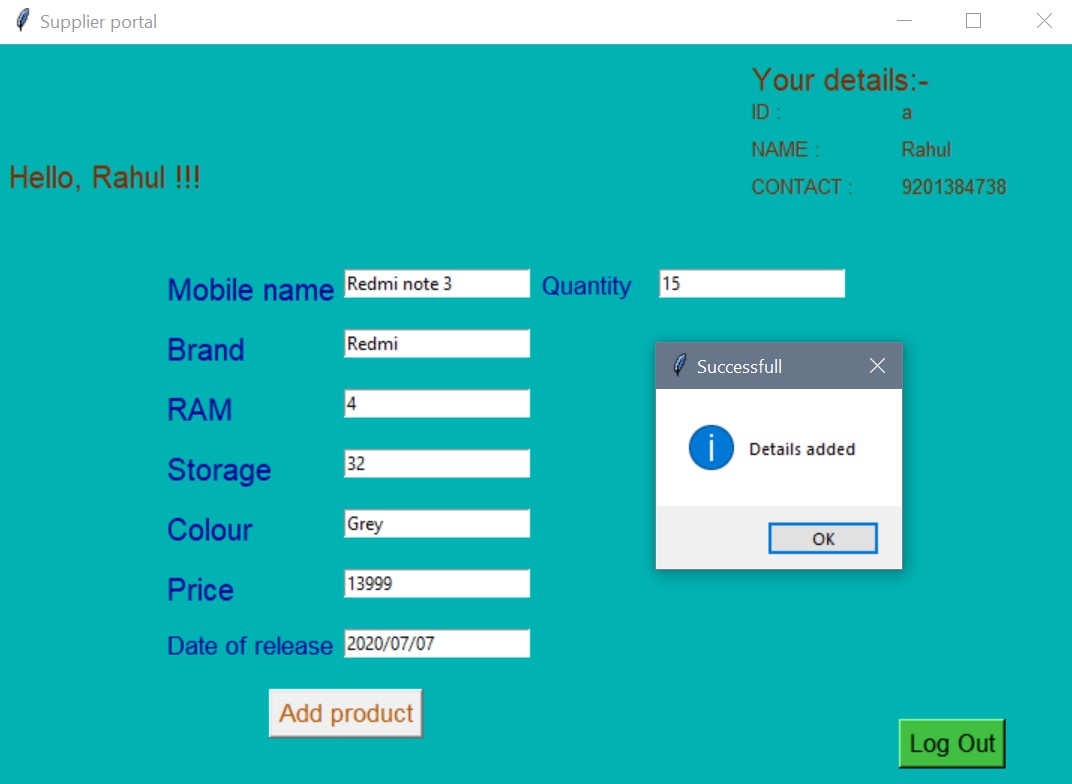
**Fig.6.12 rating feedback**

Once the product is bought by the customer it will be available in their shopping history. Later on the customer can rate the purchased mobile based on their satisfaction and experience. When the user is done with all the shopping and other necessary information they can finally log out securely.



**Fig.6.13 Supplier login**

When clicked on supplier button the above page pops up and when the supplier enters the right credentials they will be directed to the supplier portal.



**Fig.6.14 Mobile added by supplier**

When the supplier enters the portal they will be allowed to register a new mobile to the organization after successful adding of the product they can log out.

**CHAPTER 7**

**CONCLUSION:**

SALES MADE EASY project has been successfully implemented with the help of python and Tkinter modules. This project is developed in a very user-interactive way and easy to understand. This project has various scopes in terms of recommending best mobiles to the user based on their selection of options. It is also developed by keeping in mind the registration of valid customers through OTP authenticated login system. Finally, this project has helped me personally understand the concepts of python language very well and also the working of online shopping platforms. Addition of GUI to the project has made it more user-interactive and **efficient.**

**Hence Python programming language provide us several methodologies and ideas to develop real time projects.** **Therefore, this project relates to a real-world problem and provides a solution for it.**

**8. REFERENCES:**

The following websites were referred to develop this project:

1. <https://www.tutorialspoint.com/python/python_gui_programming.htm>
2. <https://www.geeksforgeeks.org/python-gui-tkinter/>
3. <https://coderslegacy.com/python/list-of-tkinter-widgets/>