

## **TABLE OF CONTENTS**

<b>Chapter no</b>	<b>Chapter name</b>	<b>Page no</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	INTRODUCTION TO FILE STRUCTURE	1
1.1.1	HISTORY	2
1.1.2	ABOUT THE FILE	2
1.1.3	VARIOUS STORAGE KIND OF FIELDS AND RECORDS	3
1.1.4	APPLICATION OF FILE STRUCTURE	4
<b>2</b>	<b>SYSTEM ANALYSIS</b>	<b>6</b>
2.1	ANALYSIS OF APPLICATION	6
2.2	STRUCTURE USED TO STORE FIELDS AND RECORDS	6
2.3	OPERATION PERFORMED ON A FILE	7
2.4	INDEXING USED	8
<b>3</b>	<b>SYSTEM DESIGN</b>	<b>9</b>
3.1	DESIGN OF FIELDS AND RECORDS	9
3.2	USER INTERFACE	9
3.2.1	ADMIN LOGIN	10
3.2.2	INSERTION OF A PRODUCT	11
3.2.3	DELETION OF A PRODUCT	12
3.2.4	UPDATE A PRODUCT	13
3.2.5	VIEW THE PRODUCT	14
3.2.6	SEARCH A PRODUCT	15
<b>4</b>	<b>IMPLEMENTATION</b>	<b>16</b>

4.1	ABOUT PYTHON	16
4.2	PSEUDOCODE	17
4.2.1	INSERTION MODULE PSEUDOCODE	17
4.2.2	DELETION MODULE PSEUDOCODE	17
4.2.3	UPDATE MODULE PSEUDOCODE	17
4.2.4	VIEW MODULE PSEUDOCODE	18
4.2.4	SEARCH MODULE PSEUDOCODE	18
4.2.5	SORTING MODULE PSEUDOCODE	18
4.2.6	BUFFER MODULE PSEUDOCODE	18
4.2.7	COMPRESSION MODULE PSEUDOCODE	18
4.3	TESTING	19
4.3.1	UNIT TESTING	18
4.3.2	INTEGRATION TESTING	20
4.3.3	SYSTEM TESTING	20
4.4	DISCUSSION OF RESULTS	21
4.4.1	MENU	21
4.4.2	INSERTION	21
4.4.3	DELETION	22
4.4.4	MODIFICATION	22
4.4.5	SORTED FILE CONTENT	23
4.4.6	BUFFERED FILE CONTENT	24
4.4.7	COMPRESSED FILE CONTENT	24
5	<b>CONCLUSION AND FUTURE ENHANCEMENT</b>	<b>25</b>
6	<b>REFERENCE</b>	<b>26</b>

## LIST OF FIGURES

<b>Figure no</b>	<b>Name of the figure</b>	<b>Page no</b>
3.2.1.1	Login Page	10
3.2.1.2	Admin Login	11
3.2.2.1	Insertion of a Product	11
3.2.2.2	After Insertion of a Product	12
3.2.3.1	Deletion of a Product	12
3.2.3.2	Conformation of Deleted Product	13
3.2.4.1	Update a Product	13
3.2.4.2	Product to be Updated	14
3.2.5.1	View all the Products	14
3.2.6.1	Search a Product	15
4.3.1.1	Admin Login	19
4.3.1.2	Record.txt	19
4.3.2.1	Authorization success	20
4.4.1.1	Menu	21
4.4.2.1	Insertion	21
4.4.3.1	Before Deletion	22
4.4.3.2	After Deletion	22
4.4.4.1	Before Modification	23
4.4.4.2	After Modification	23
4.4.5.1	Sorted File	23
4.4.6.1	Buffered File	24
4.4.7.1	Compressed File	24

# CHAPTER 1

## INTRODUCTION

### 1.1. Introduction to File Structure

File Structures is the Organization of Data in Secondary Storage Device in such a way that minimizes the access time and the storage space. A File Structure is a combination of representations for data in files and of operations for accessing the data. File Structure allows applications to read, write and modify data. It also supports finding the data that matches some search criteria or reading through the data in some particular order.[7] An improvement in file structure design may make an application hundreds of time faster. The details of the representation of the data and the implementation of the operations determine the efficiency of the file structure for particular applications.

The goal of File Structure is to get the information we need with one access to the disk. If it is not possible, then get the information with as few accesses as possible. Group information so that we are likely to get everything we need with only one trip of the disk. It is relatively easy to come up with File Structure designs that meet the general goals when the files never change. When files grow or shrink when information is added and deleted, it is much more difficult.

Goal of this course is with reference to time and space is to first minimize number of trips to the disk in order to get desired information. Ideally get what we need in one disk access or get it with a few disk accesses as possible. Secondly grouping related information so that we are likely to get everything we need with only one trip to the disk for example name, address, phone number, account balance.

Good File Structure design must have:

- Fast access to great capacity.
- Reduce the number of disk accesses.
- By collecting data into buffers blocks or buckets.
- Manage growth by splitting these collections.
- Waiting time for the applications becomes less for the disk.

### 1.1.1 History

- History of File Structure design, In the beginning the file access was sequential, and the cost of access grew in direct proportional to the size of the file. So Indexes were added to files.
- Indexes made it possible to keep a list of keys and pointers in a smaller file that could be searched more quickly.
- Simple indexes become difficult to manage for dynamic files in which the set of keys changes. Hence Tree Structures were introduced.
- Trees grew unevenly as records were added and deleted, resulting in long searches requiring multiple disk accesses to find a record. Hence an elegant, self-adjusting binary tree structure called an AVL Tree was developed for data in memory.[3]
- Even with a balanced binary tree, dozens of accesses were required to find a record in moderate sized files.
- A method was needed to keep a tree balanced when each node of the tree was not a single record as in a binary tree, but a file block containing hundreds of records. Hence B-trees were introduced.
- AVL trees grow from top down as records are added, B-trees grow from the bottom up.
- B-trees provided excellent access performance but, a file could not be accessed sequentially with efficiency.
- The above problem was solved using B+ tree which is a combination of a B-tree and a sequential linked list added at the bottom level of the B-tree.
- To further reduce the number of disk accesses, Hashing was introduced for files that do not change size greatly overtime.

### 1.1.2 About the File

A File is an object on a computer that stores data, information, settings, or commands used with a computer program. In a graphical user interface such as Microsoft Windows, files display as icons that relate to the program that opens the file. For example, the picture is an icon associated with adobe acrobat PDF files, if the file was on your computer, double clicking the icon in Windows would open that file in adobe acrobat or the PDF reader installed on the computer.

A File is created using a software program on the computer. For example, to create a text file we use text editor, to create an image file we use an image editor, and to create a document we use a word processor.

Files are not made for just reading the contents, we can also perform some operations on the Files.

- **Read operation:** Meant to read the information which is stored into the files.
- **Write operation:** For inserting some new contents into a file.
- **Rename or Change the Name of the file.**
- **Copy the file from one location to the other.**
- **Sorting or arrange the contents of the file.**
- **Move or cut the file from one place to another.**
- **Delete a file.**
- **Execute Means to Run Means File Display Output.** We can also link a file with any other File.

These are also called Symbolic Links, in the symbolic links all the files are linked by using some text Alias.

We can also link a file with any other File. These are also called Symbolic Links, in the symbolic links all the files are linked by using some text Alias.

### **1.1.3 Various storage kinds of fields and records**

#### **Field Structures:**

There are many ways of adding structure to files to maintain the identity of fields. Four most common methods follow:

- Method 1: Force the fields into a predictable length. Fix the length of fields. The fields of the file vary in length to make the fields fixed length we have to predict lengths.
- Method 2: Begin each field with a length indicator, store the field length just ahead of the field.
- Method 3: Place a delimiter at the end of each field to separate it from the next field.
- Method 4: Use a “keyword = Value” expression to identify each field and its contents.

## Record Structures:

A record can be defined as a set of fields that belong together when the file is viewed in terms of a higher level organization.[4] Five most common methods follow:

- Method 1: Make the records be a predictable number of bytes in length.
- Method 2: Make the record be a predictable number of fields in length.
- Method 3: Begin each record with a length indicator consisting of a count of the number of bytes that the record contains.
- Method 4: Use a second file to keep track of the beginning byte address for each record.
- Method 5: Place a delimiter at the end of each record to separate it from the next record.

### 1.1.4. Application of File Structure

#### Amaze files manager:

Amaze File Manager is a newer app comparatively speaking and it's a pretty good. It's open source and focuses on a lighter experience for those who just need to do some light file browsing. It features Material Design, SMB file sharing, a built-in app manager to uninstall apps, root explorer, and more.

#### Asus file manager:

It's not every day we see an OEM app make an app list, but File Manager by ASUS is actually really good. It's compatible with most devices, even non-ASUS ones. You'll also get clean, simple interface with LAN and SMB support, cloud storage support, support for various types of files, archiving support, and more.[2] It's entirely free with no in-app purchases and provides a great experience for a simple file browser.

#### ES file explorer pro:

ES File Explorer has been around as long as most Android nerds can remember and comes with pretty much every feature you can ask for in a file browser. A while back, it was purchased by another company. Since then, things haven't gone well. The free version of the app, while very capable, now has a ton of added bloatware that not only doesn't add to the experience, but actively subtracts from it. Thankfully, the pro version of the app doesn't have these features.

### **File manager:**

File Manager is blandly named, but it's actually quite good. It's a newer file manager app that gives you one of the best sets of features without adding too much bloat. You'll get basic file management features along with cloud storage features, NAS support, and more. You can even browse your installed apps, music, and video with this and the player isn't half bad. Perhaps the best part is that the app is free with no in-app purchases and advertising.

### **MK explorer:**

MK Explorer is another newer file manager option. It's a simple option that doesn't have a whole lot of flair. That is extremely preferable if you really just want something simple. It features a Material Design interface, the basic file management features (copy, paste, delete, SD Card support for Lollipop), and root access.[2] There are also supports for 20 languages and it has a built-in text editor, gallery, and music player. It doesn't have anything like cloud storage or network storage support, but that's not really what it's for. It's a good, cheap option.

### **X-Plore files manager:**

X-Plore File Manager is one of the more unique options on the list. It's a forced dual pane app which means you'll be managing two windows at once pretty much all the time. This is kind of cool if you're copy/pasting between folders or need to move files quickly. It also comes with support for various types of files, cloud storage, network storage, a built-in hex editor, root support, and plenty of other features.

## CHAPTER 2

# SYSTEM ANALYSIS

### 2.1. Analysis of application

The users are provided with various alternatives which include new user and manager and Close application. This project is build at user end as well as at an administrative end,in which public can use user end and manager can use administrative end.In the user end users can interact with the system and hence enables to choose the option based on his/her choice. In the administrative end the admin can perform various operations like adding and removing of products.[1]

Supermarkets are always buzzing with activities. A busy place, with so many customers, shopping, and transactions going on. You cannot afford a wrong transaction with so much hustle and bustle. If one transaction goes wrong, it disturbs the string of activities after it.The best solution to avoid errors in day-to-day operations is a reliable POS system. It will ensure that all your supermarket activities are happening efficiently and effectively. This leads to an excellent shopping experience for customers and outstanding operations management for supermarket owners.

Supermarket Management including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirementsare included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs proposed provides a view of how the proposed system will be more efficient than the existing one.

### 2.2 Structure used to store Fields and Records

- A field is an item of stored data. A field could be a name, an address, a description etc.
- A record is the collection of fields that related to a single entity.

All details accepted during the login of a user is written onto a users.txt.

The product details are added by the admin to the products.txt file.

Each record is separated by a '\n' (next line) character and each field in the record is separated by a '|' (delimiter).

Product Details are saved as follows:

id | name | manufacture\_date | expiry\_date | amount | unit \n

## 2.3 Operations Performed On A File

The files are used to store the required information for its later uses. There are many file operations that can be performed by the computer system. Here are the lists of some common file operations:

- File Create operation
- File Delete operation
- File Open operation
- File Close operation
- File Read operation
- File Write operation
- File Search operation
- File Append operation

### **File Create operation:-**

**The file is created with no data.**

- The file create operation is the first step of the file.
- Without creating any file, there is no operation can be performed.

### **File Delete operation: -**

- File must have to be deleted when it is no longer needed just to free up the disk space.
- The file delete operation is the last step of the file.
- After deleting the file, if doesn't exist.

### **File Open operation:-**

- The process must open the file before using it.

### **File Close operation:-**

- The file must be closed to free up the internal table space, when all the accesses are finished and the attributes and the disk addresses are no longer needed.

**File Read operation:-**

- The file read operation is performed just to read the data that are stored in the required file.

**File Write operation:-**

- The file write operation is used to write the data to the file again, generally at the current position.

**File Search operation:-**

- The file search operation is used to search the data to the file based on primary key.

**File Append operation:-**

- The file append operation is used to add records on to the end of the record

## 2.4 Indexing Used

An Index is a tool for finding records in a file. It consists of

- Key field on which the index is searched.
- Reference field that tells where to find the data file record associated with particular key.

Here we have used a simple indexing:

- We choose to organize the file as variable length record with a size field preceding each record. The fields within each record are also of variable-length but are separated by delimiters.
- We form a Primary key by concatenating the record user id and the record's ID number. This should form a unique identifier.

In this project primary indexing as well as secondary indexing is implemented as follows:

- This project allows to organize the file as variable length record with a size field preceding each record. The fields within each record are also of variable – length but are separated by delimiters.
- Here, it searches a particular record based on the **primary key**.
- In this program the concept of the **secondary index** is also used as the search are based on this concept only like if we ask for the modification or deletion it is done on the basis of the secondary index.

## CHAPTER 3

# SYSTEM DESIGN

### 3.1 Design of the Fields and Records

Supermarket Management is a menu driven application in which menu allows a administer to interpolate, modify, expunge various details of products in order to make the website more user friendly.

After choosing the options on menu the admin can enter the product name which they want to buy. The administrator has to login using the given password and the further procedure of the administrator will takes place.

## SUPERMARKET

### 1. Admin Dashboard:

Admin part has the following functionalities:

1. Can view the number of products available and as well as the product purchased in the record.
2. Add new products to the record.
3. Delete the products from the record.
4. Update the product list from the record.
5. Can view the products purchased by the customer.

### 3.2 User Interface:

- The user interface (UI), in the industrial design field of human-computer interaction, is the space where interactions between humans and machines occur.
- The goal of this interaction is to allow effective operation and control of the machine from the human end, whilst the machine simultaneously feeds back information that aids the operators' decision-making process.
- Examples of this broad concept of user interfaces include the interactive aspects of computer

operating systems, hand tools, machinery operator controls, and process controls. The design considerations applicable when creating user interfaces are related to or involve such disciplines as ergonomics and psychology.

- The junction between a user and a computer program. An interface is a set of commands or menus through which a user communicates with a program.

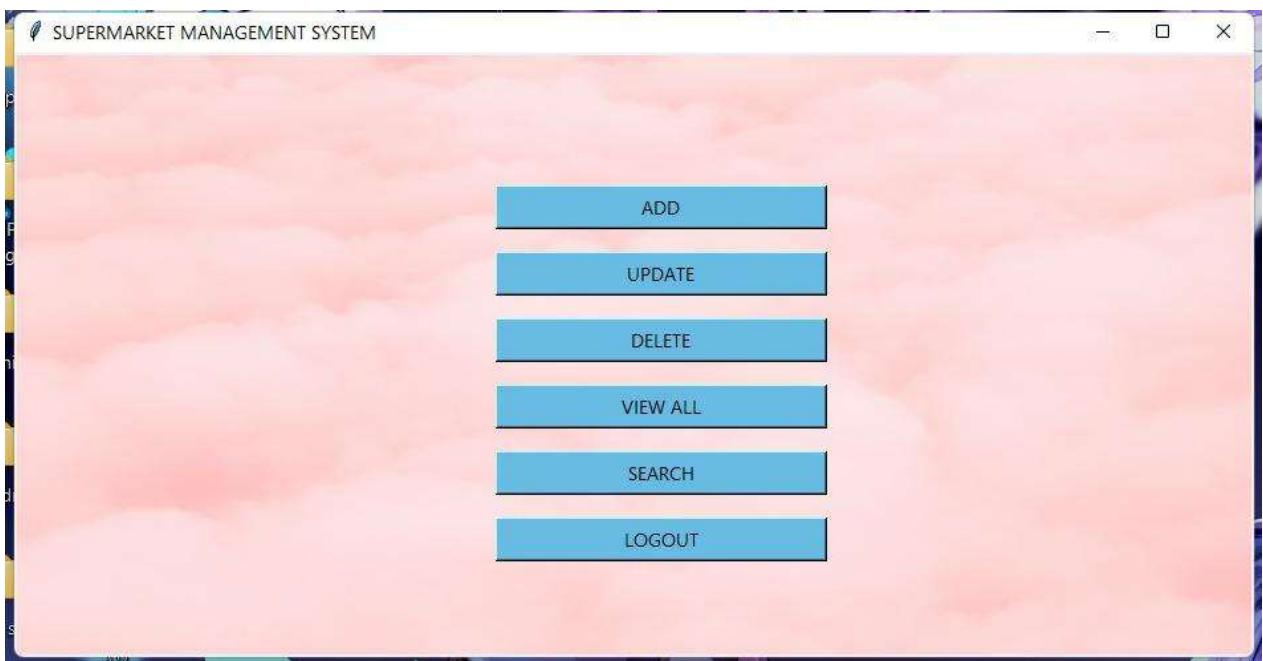
### 3.2.1 Admin login

- A new record is being inserted into the record.txt file when admin logins with his/her credentials.



**Fig 3.2.1.1 Login Page**

- A admin logged in to choose further options.



**Fig 3.2.1.2 Admin login**

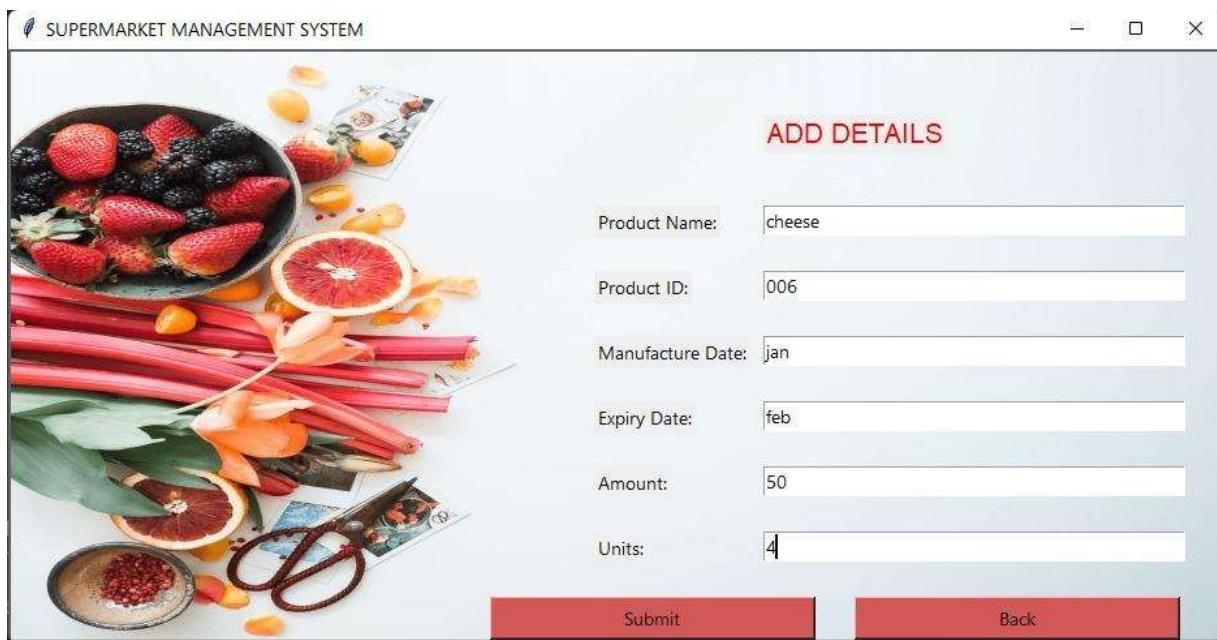
### 3.2.2 Insertion of a Product

- A new product is being added to product.txt file.

A screenshot of a Windows application window titled "SUPERMARKET MANAGEMENT SYSTEM". The window features a decorative background image of various fruits like strawberries, blackberries, and citrus slices. On the right, there is a form titled "ADD DETAILS" with six input fields: "Product Name", "Product ID", "Manufacture Date", "Expiry Date", "Amount", and "Units", each with a corresponding text input field. At the bottom of the form are two red buttons labeled "Submit" and "Back".

**Fig 3.2.2.1 Insertion of a Product**

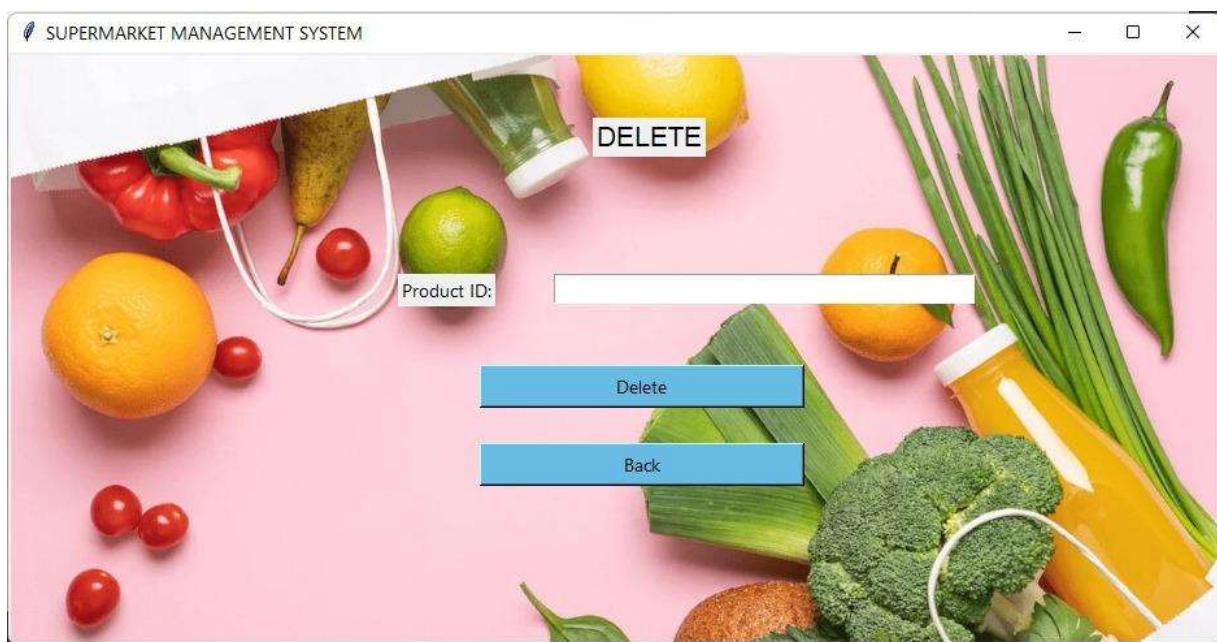
- Confirmation that the product is added.



**Fig 3.2.2.2 After insertion of a product**

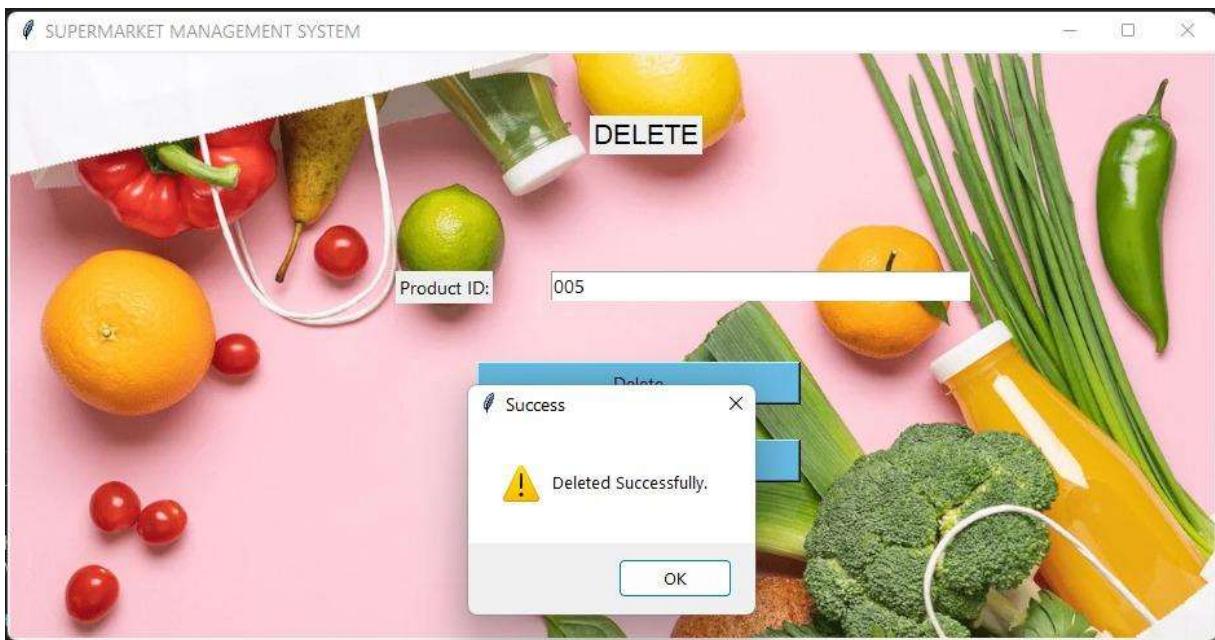
### 3.2.3 Deletion of a Product

- A product is deleted from the file product.txt file.



**Fig 3.2.3.1 Deletion of a product**

- Confirmation that the product is deleted.



**Fig 3.2.3.2 Conformation of a deleted product**

### 3.2.4 Update a product

- A product is updated and it is stored in product.txt file

A screenshot of a Windows application window titled "SUPERMARKET MANAGEMENT SYSTEM". The title bar has a logo on the left. The main area is titled "UPDATE DETAILS" in red. It contains six text input fields with labels: "Product name:", "Product ID:", "manufacture\_date:", "Expiry Date:", "Amount:", and "Units:". At the bottom are two blue buttons: "Submit" on the left and "Back" on the right. The background features a close-up image of green leaves.

**Fig 3.2.4.1 Update a product**

- The product details are entered which is to be updated.

SUPERMARKET MANAGEMENT SYSTEM

UPDATE DETAILS

Product name: cheesecake

Product ID: 006

manufacture\_date: jan

Expiry Date: feb

Amount: 50

Units: 4

Submit Back

**Fig 3.2.4.2 Product to be updated**

### 3.2.5 View the Product

- The products which is added by the admin will be visible.

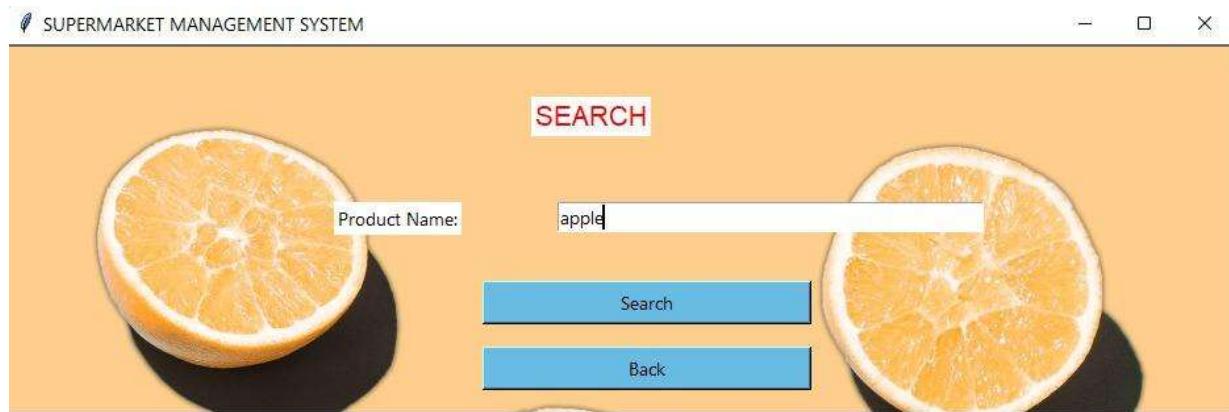
Product Name	Product ID	Manufacture_date	Expiry Date	Amount	Units
chesse	006	jan	feb	50	4
tea	004	jan	feb	40	2
grape	003	jan	feb	50	1
almond	002	jan	feb	200	1
apple	001	jan	feb	20	3

Back

**Fig 3.2.5.1 View all the products**

### 3.2.6 Search a Product

- Search a product by giving the product name.



apple|001|jan|feb|20|3|2022:07:14-14:24:36

**Fig 3.2.6.1 Search a product**

## Chapter 4

# IMPLEMENTATION

### 4.1 About Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective

## 4.2 Pseudo Code

- Pseudo code is a simple way of writing programming code in English.  
Pseudo code is not actual programming languages.
- It uses short phrases to write code for programs before you actually create it in a specific language.
- Once the functionality of the program is known, then one can use pseudo code to create statements to achieve the required results for your program.

### 4.2.1 Insertion module pseudo code

**Step1:** Input the variables Product's name, Product's ID, Manufactured date, Expiry date, Product's amount, Quantity into the add menu.

**Step2:** Check whether the entered inputs are correct.

**Step3:** Read product.txt and check whether the product is added.

### 4.2.2 Deletion module pseudo code

**Step1:** Click the delete button.

**Step2:** Enter the product's id which is to be deleted.

**Step3:** Confirmation message appears if the product is deleted.

**Step4:** Open product.txt to check if the product is deleted or not.

### 4.2.3 Update module pseudo code

**Step1:** Click the update button.

**Step2:** Enter the details of the product which is to be updated.

**Step3:** Open the file product.txt and check whether the product is updated or not.

### 4.2.4 View module pseudo code

**Step1:** Click on the view button.

**Step2:**The products which are modified by the admin will appears.

#### **4.2.5 Search module pseudo code**

**Step1:**Click on the search button.

**Step2:**Enter the product's name you need to search.

**Step3:**The entered product will appear on the bottom.

#### **4.2.6 Sorting module pseudo code**

**Step1:**Add the products which are brought by the customer.

**Step2:**The products details according to their name will be stored.

**Step3:**The sorted list of product's will be present at sorted.txt file.

#### **4.2.7 Buffer module pseudo code**

**Step1:**Run the buffer module inside the source file.

**Step2:**The file buffered.txt will be created.

**Step3:**And the buffered data of admin.txt file will be stored.

**Step4:**The size of the file is compressed.

#### **4.2.8 Compression module pseudo code**

**Step1:**Run the compression which is inside the source file.

**Step2:**The comp.txt file is created to store the compression data.

**Step3:**The data of the compressed file is seen.

### **4.3 Testing**

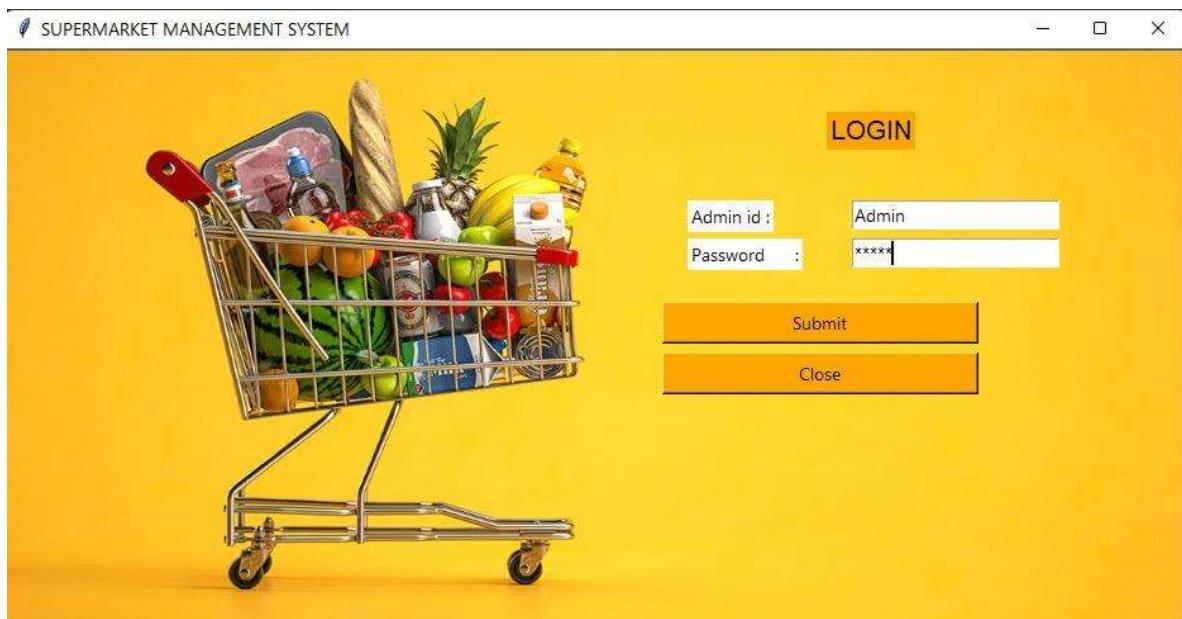
#### **4.3.1 Unit Testing**

Unit testing is software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit

for use.

We are going to test the components of admin login.

- When the admin logs in with the credentials like Admin username, password it should reflect in the record.txt file.



**Fig 4.3.1.1 Admin login**

A screenshot of a terminal window titled "record.txt". The window shows a list of 11 entries, each consisting of three columns separated by vertical bars. The first column contains numbers from 1 to 11. The second column contains the word "Admin". The third column contains dates and times ranging from "2022:07:14 | 11:40:16" to "2022:07:14 | 14:02:14".

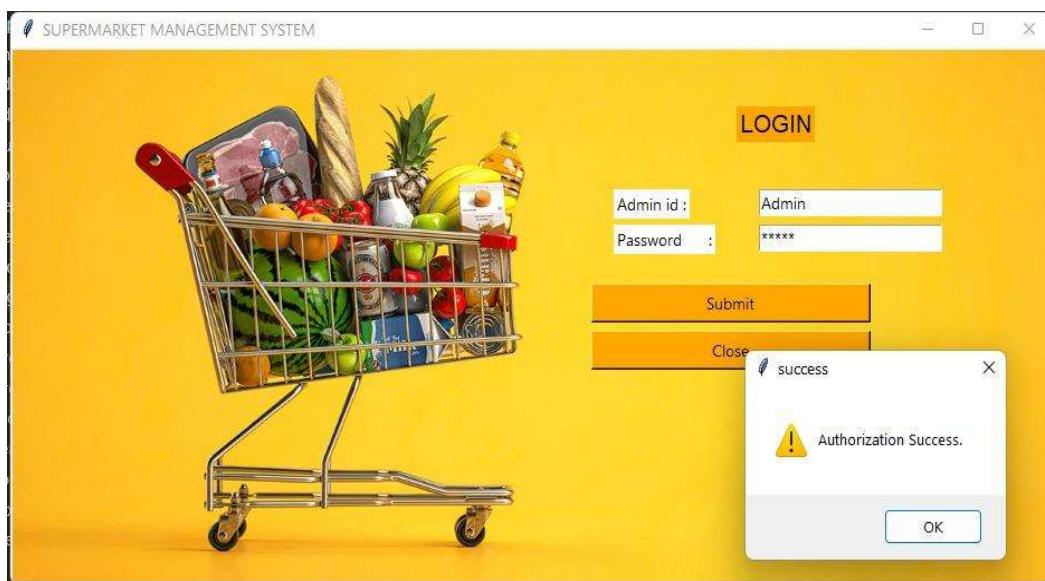
1	Admin	2022:07:14   11:40:16
2	Admin	2022:07:14   11:40:32
3	Admin	2022:07:14   11:40:45
4	Admin	2022:07:14   11:40:59
5	Admin	2022:07:14   11:41:13
6	Admin	2022:07:14   11:41:28
7	Admin	2022:07:14   11:41:42
8	Admin	2022:07:14   13:11:57
9	Admin	2022:07:14   14:00:16
10	Admin	2022:07:14   14:02:14
11		

**Fig 4.3.1.2 record.txt**

### 4.3.2 Integration Testing

Integration testing is a phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing.

Integration testing follows two approaches known as ‘Top down’ approach and ‘Bottom Down’ approach. Admin and user have relation between them, here on the admin side i.e., in managing the centers we update, delete and add new centers into the centers list.[2] As soon we do any changes in the admin side, user side also gets updated. In this way we have developed small units and then tested by integrating them using integration testing.



**Fig 4.3.2.1 Authorization success**

### 4.3.3 System Testing

System testing is a level of software testing where complete and integrated software is tested. The process of testing an integrated system to verify that it needs specified requirements.[4]

The process of testing an integrated system to verify that it needs specified requirements. This program works well on the Windows operating system and as well as on the Linux operating system. The IDE used for coding is Visual Studio Code. It is

recommended to run the project in an IDE that supports the latest version of the programming language.

## 4.4 Discussion of Results

### 4.4.1 Menu

At first the admin logs in to enter the product details using his/her credentials.

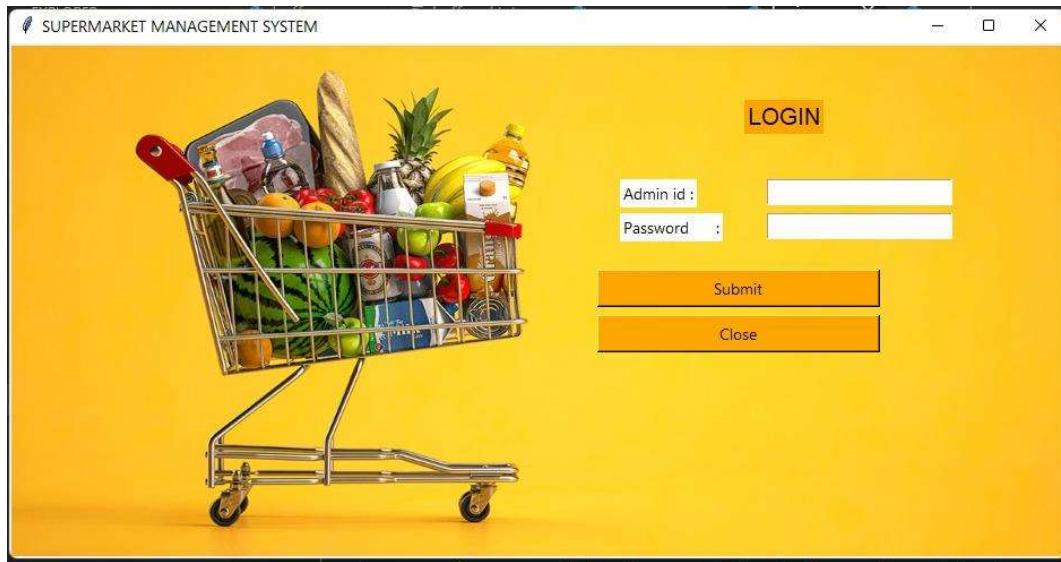


Fig 4.4.1.1 Menu

### 4.4.2 Insertion

The product.txt file after insertion of a product.

```
product.txt
1 apple|001|jan|feb|20|3|2022:07:14-14:24:36
2 almond|002|jan|feb|200|1|2022:07:14-14:24:54
3 grape|003|jan|feb|50|1|2022:07:14-14:25:12
4 tea|004|jan|feb|40|2|2022:07:14-14:25:28
5 chesse|006|jan|feb|50|4|2022:07:14 - 15:04:05
6
```

Fig 4.4.2.1 Insertion

#### 4.4.3 Deletion

The product.txt file before deletion of a product.

```
product.txt
1 apple|001|jan|feb|20|3|2022:07:14-14:24:36
2 almond|002|jan|feb|200|1|2022:07:14-14:24:54
3 grape|003|jan|feb|50|1|2022:07:14-14:25:12
4 tea|004|jan|feb|40|2|2022:07:14-14:25:28
5 chesse|006|jan|feb|50|4|2022:07:14 - 15:04:05
6
```

Fig 4.4.3.1 Before Deletion

```
product.txt
1 apple|001|jan|feb|20|3|2022:07:14-14:24:36
2 almond|002|jan|feb|200|1|2022:07:14-14:24:54
3 grape|003|jan|feb|50|1|2022:07:14-14:25:12
4 tea|004|jan|feb|40|2|2022:07:14-14:25:28
5
```

Fig 4.4.3.2 After Deletion

#### 4.4.4 Modification

The product.txt file before and after modification.

```
product.txt
1 apple|001|jan|feb|20|3|2022:07:14-14:24:36
2 almond|002|jan|feb|200|1|2022:07:14-14:24:54
3 grape|003|jan|feb|50|1|2022:07:14-14:25:12
4 tea|004|jan|feb|40|2|2022:07:14-14:25:28
5 chesse|006|jan|feb|50|4|2022:07:14 - 15:04:05
6
```

**Fig 4.4.4.1 Before Modification**

```
product.txt
1 apple|001|jan|feb|20|3|2022:07:14-14:24:36
2 almond|002|jan|feb|200|1|2022:07:14-14:24:54
3 grape|003|jan|feb|50|1|2022:07:14-14:25:12
4 tea|004|jan|feb|40|2|2022:07:14-14:25:28
5 cheesecake|006|jan|feb|50|4|2022:07:17 - 11:10:19
6
```

**Fig 4.4.4.2 After Modification**

#### 4.4.5 Sorted File Content

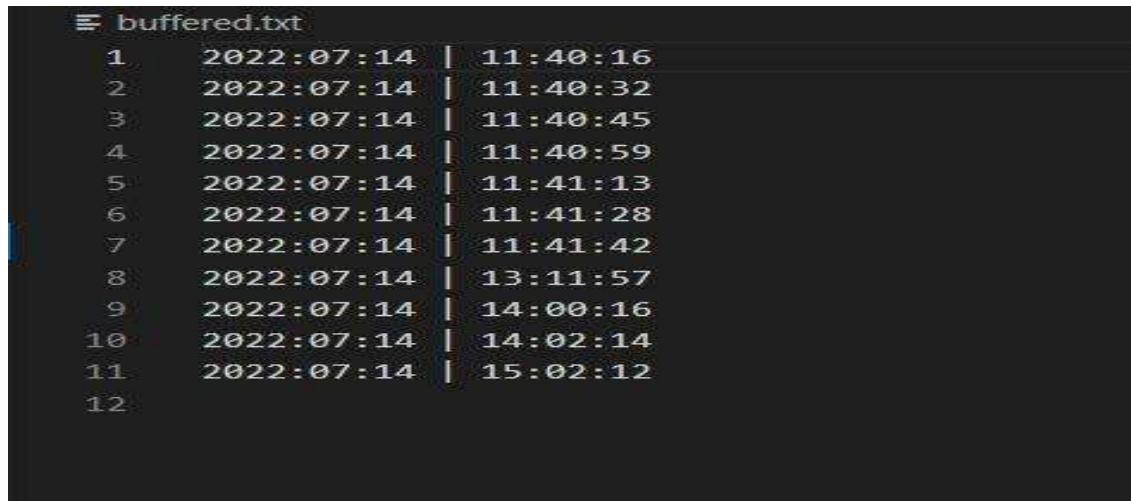
The sorted contents are added to sorted.txt file.

```
sorted.txt
1 almond|002|jan|feb|200|1|#
2 apple|001|jan|feb|20|3|#
3 cake|006|jan|feb|50|3|#
4 grape|003|jan|feb|50|1|#
5 tea|004|jan|feb|40|2|#
6
```

**Fig 4.4.5.1 Sorted file**

#### 4.4.6 Buffered File Content

The buffered data are added to buffered.txt file.



The screenshot shows a terminal window with the command 'cat buffered.txt' at the prompt. The output displays 12 lines of data, each consisting of a number from 1 to 12 followed by a timestamp in the format '2022:07:14 | 11:40:16' through '15:02:12'. The terminal has a dark background with white text.

```
█ buffered.txt
 1 2022:07:14 | 11:40:16
 2 2022:07:14 | 11:40:32
 3 2022:07:14 | 11:40:45
 4 2022:07:14 | 11:40:59
 5 2022:07:14 | 11:41:13
 6 2022:07:14 | 11:41:28
 7 2022:07:14 | 11:41:42
 8 2022:07:14 | 13:11:57
 9 2022:07:14 | 14:00:16
10 2022:07:14 | 14:02:14
11 2022:07:14 | 15:02:12
12
```

**Fig 4.4.6.1 Buffered File**

#### 4.4.7 Compressed File Content

The compressed data is stored at comp.txt file.



The screenshot shows a terminal window with the command 'cat comp.txt' at the prompt. The output consists of a single line of highly compressed binary data, appearing as a long string of characters. The terminal has a dark background with white text.

```
█ comp.txt
1 e1fptzkE0vjAWDMC7X9EPVH1cu4B/40IAoRiyytGPb3opkVqtTKfRaq2UKT1i83d1v6fRCT3WII8aeA+sDjriWMCm9yffqqXGoooc9FobW18q6GzrP1QEPySVcgbZHTaQzrDaLmmW1b0bLxAtdnmuF1wVAe4M22:
```

**Fig 4.4.7.1 Compressed File**

## CHAPTER 5

### CONCLUSION AND FUTURE ENHANCEMENT

This website provides a computerized version of Supermarket management which will benefit the customer who visits the supermarket . It makes entire process online where customer can search and buy the product of their choice.[1] It also has a facility for admin login where admin can login and can see status of products issued . In, the conclusion I would like to say that there are several supermarkets that are coming up in the city. More or less every super markets contain similar items. Due to this there is a high tendency of competitive market. People have a high tendency to go to the super market. But some times we have seen that quality of the product has gone below average and people have shifted toward the other super markets.

In future the Supermarket Management System will be more keen to serene to use, more user interactive and more dependable

## CHAPTER 6

## REFERENCES

- [1] [www.google.com](http://www.google.com)
- [2] [www.stackoverflow.com](http://www.stackoverflow.com)
- [3] [www.youtube.com](http://www.youtube.com)
- [4] [www.quora.com](http://www.quora.com)
- [6] [www.geeksforgorgeeks.org](http://www.geeksforgorgeeks.org)
- [7] File Structures-An Object Oriented Approach with C++ by Michael J.Folk,Bill Zoellick, Greg Riccardi.