



**NEW HORIZON  
COLLEGE OF ENGINEERING**



**A MINI PROJECT**

**REPORT**

*for*  
**MINI PROJECT USING PYTHON (20CSE59)**

**AUTO COMPONENTS MANUFACTURING UNIT**

*submitted by*

**N KAVYA  
1NH18CS118  
5B**

*In partial fulfillment for the award of*

*the degree of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



## Certificate

*This is to certify that the mini project work titled*

### **AUTO COMPONENTS MANUFACTURING UNIT**

*Submitted in partial fulfillment of the degree of Bachelor of Engineering  
in Computer Science and Engineering*

*Submitted by*

N KAVYA

1NH18CS118

*DURING*

*ODD SEMESTER 2020-2021*

*For*

20CSE59

Signature of Reviewer

Signature of HOD

### **SEMESTER END EXAMINATION**

*Name of the Examiner*

*Signature with date*

1. \_\_\_\_\_

\_\_\_\_\_

2. \_\_\_\_\_

\_\_\_\_\_

# AUTO COMPONENTS MANUFACTURING UNIT

## ORIGINALITY REPORT

<b>6%</b>	%	<b>6%</b>	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

## PRIMARY SOURCES

- |          |  |               |
|----------|--|---------------|
| <b>1</b> | Peter Membrey, David Hows. "Learn Raspberry Pi with Linux", Springer Science and Business Media LLC, 2013  | <b>3%</b>     |
| <b>2</b> | Charles Bell. "MySQL for the Internet of Things", Springer Science and Business Media LLC, 2016  | <b>1%</b>     |
| <b>3</b> | Hua Li, Chonghua Wang, Yiming Rong. "Service-Oriented Architecture Based Dockside Container Crane CAD/CAE Integration System", Volume 2: 29th Computers and Information in Engineering Conference, Parts A and B, 2009 | <b>1%</b>     |
| <b>4</b> | Dolores Cuadra, Paloma Martínez, Elena Castro, Harith Al-Jumaily. "Guidelines for representing complex cardinality constraints in binary and ternary relationships", Software & Systems Modeling, 2012                 | <b>&lt;1%</b> |
- Publication
- Publication
- Publication
- Publication

5

Ossama Embarak. "Chapter 1 Introduction to Data Science with Python", Springer Science and Business Media LLC, 2018

<1 %

Publication

---

6

Manohar Swamynathan. "Chapter 1 Step 1 – Getting Started in Python", Springer Science and Business Media LLC, 2017

<1 %

Publication

---

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

On

## **ABSTRACT**

The automobile industry comprises a wide range of companies and organizations involved in the design, development, manufacturing, marketing, and selling of motor vehicles. The global auto component market is a highly diversified sector that involves engine and auto component manufacturers, including aftermarket parts manufacturers, suppliers, dealers, and retailers. Manufacturing unit of auto components will have many sub branches. It must require organising of raw materials supplier, various manufacturing departments like E-treatment of metal, sub-components designer, finish match grinding, assembling, testing, inspection and calibration, stock checking, finance, retailer, consultant, etc. Apart from department, machines, employees, shifts and working calendar must be maintained.

Therefore we need an application to organise the working of manufacturing industry, operations in demand and retailers and suppliers requirement. It must contain all the information and records of the day in and out working of the company. The application must give different access to visitor, buyer, workers and the owner.

Auto Components Manufacturing Unit is the project focused to organize a small scale industry that manufactures sub auto components for huge automobile industries. It involves the idea of labeling the name of the buyer on the product. The items added to cart remains saved until billed. After successful payment, the bill is written on to a file. The buyers can provide feedback to the company. It resembles the hierarchical model of the working of the organization. The manufacturing head runs the manufacturing unit which involves many departments. Each department manufactures one product. Each department is headed by Stock manager who maintains the raw input and also the productivity from the department. The Stock manager supervises the operations of the two Shift managers in his/her department. The Shift manager in turn supervises the working of the employees under his/her shift. The workers can share their views to the head of the manufacturing unit/

## **ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha**, Principal NHCE, for his constant support and encouragement.

I am grateful to **Dr.Prashanth C.S.R**, Dean Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and Head, Department of Computer Science and Engineering, for her constant support.

I express my gratitude to **Mr.Bhaskar S V**, Assistant Professor, my project reviewer, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

**N KAVYA  
1NH18CS118**

## **CONTENTS**

	<b>Page no.</b>
<b>ABSTRACT</b>	I
<b>ACKNOWLEDGEMENT</b>	II
<b>LIST OF FIGURES</b>	V
<b>1. INTRODUCTION</b>	
1.1. COURSE OBJECTIVE	1
1.2. PROBLEM DEFINITION	1
1.3. OUTCOMES OF PROJECT WORK	2
<b>2. REQUIREMENTS AND DESIGN</b>	
2.1 HARDWARE REQUIREMENTS	3
2.2 SOFTWARE REQUIREMENTS	3
<b>3. ER MODEL AND RELATIONAL SCHEMA</b>	
3.1 ENTITY AND ATTRIBUTES	4
3.2 KEYS	5
3.3 RELATIONSHIP AND PARTICIPATION	6
3.4 PROBLEM STATEMENTS	8
3.5 ER MODEL	9
3.6 SQL	10
<b>4. PYTHON FUNDAMENTALS</b>	
4.1. INTRODUCTION TO PYTHON	16
4.2. DATASTRUCTURES	19
4.3. TKINTER WIDGETS	21

<b>5. IMPLEMENTATION</b>	
5.1 LIBRARIES	<b>22</b>
5.1.1 MYSQL IN PYTHON	<b>22</b>
5.1.2 TKINTER	<b>22</b>
5.1.3 TKCALENDAR	<b>23</b>
5.1.4 REGULAR EXPRESSION	<b>23</b>
5.1.5 PILLOW -IMAGE TK	<b>24</b>
5.3 CODE IMPLEMENTATION	<b>24</b>
<b>6. RESULTS</b>	<b>27</b>
<b>7. CONCLUSION</b>	<b>43</b>
<b>REFERENCES</b>	<b>44</b>

## **LIST OF FIGURES**

<b><u>Fig. No</u></b>	<b><u>Figure Description</u></b>	<b><u>Page No</u></b>
1.	SQL CONNECTION	22
2.	TKCALENDAR	23
3.	IMAGETK	24
4.	FILE OUTPUT	25
3.1.	REGISTERED BUYER LOGIN	27
3.2.	REGISTERED BUYER CHANGE OF PASSWORD	27
3.3.	NEW USER REGISTRATION	28
3.4.	SHOPPING PAGE	28
4.	BILL	29
5.	BUYER'S FEEDBACK	29
6.	WORKER'S MESSAGE	30
7.1.	MAIN ADMIN LOGIN	30
7.2.	ADMIN PAGE-VIEW FEEDBACK	31
7.3.	ADMIN PAGE-VIEW MESSAGES	31
8.	ADMIN PAGE-ADD BONUS	32
9.	DATA BEFORE AND AFTER SALARY BONUS UPDATE	32
10.	ADMIN PAGE-VIEW SHOPPING	33
11.	ADMIN PAGE- INCOME IN THE GIVEN PERIOD	33
12.	ADMIN PAGE-VIEW SUPPLY	34
13	ADMIN PAGE-VIEW EXPENDITURE	34

<u>Fig. No</u>	<u>Figure Description</u>	<u>Page No</u>
1.	ADMIN PAGE-VIEW DEPARTMENT	35
2.	ADMIN PAGE-VIEW STAFF	35
3.	STOCK MANAGER LOGIN	36
4.	STOCK MANAGER PAGE-VIEW SHIFT MANAGERS	36
3.1.	STOCK MANAGER PAGE-VIEW SHIFT 1 MANAGERS	37
3.2.	STOCK MANAGER PAGE-VIEW SHIFT 2 MANAGERS	37
3.3.	STOCK MANAGER PAGE-VIEW SUPPLIES	38
3.4.	STOCK MANAGER PAGE-ADD PURCHASE	38
4.	STOCK MANAGER PAGE-IMPORT GOODS	29
5.	STOCK MANAGER PAGE-ADD QUANTITY	29
6.	STOCK MANAGER PAGE-	30
7.1.	MAIN ADMIN LOGIN	30
7.2.	ADMIN PAGE-VIEW FEEDBACK	31
7.3.	ADMIN PAGE-VIEW MESSAGES	31
8.	ADMIN PAGE-ADD BONUS	32
9.	DATA BEFORE AND AFTER SALARY BONUS UPDATE	32
10.	ADMIN PAGE-VIEW SHOPPING	33
11.	ADMIN PAGE- INCOME IN THE GIVEN PERIOD	33
12.	ADMIN PAGE-VIEW SUPPLY	34
13	ADMIN PAGE-EXPENDITURE	34

## CHAPTER 1

# INTRODUCTION

### 1.1 COURSE OBJECTIVE

The objective of this Mini-project is to apply the programming knowledge of Python Language, Database Management System concepts and Graphical User Interface into real-time applications to develop and provide solutions to several real-world problems in an effective and efficient manner and also to gain exposure and acquire the programming skills required for a good computer science engineer.

### 1.2 PROBLEM DEFINITION

The automobile industry comprises a wide range of companies and organizations involved in the design, development, manufacturing, marketing, and selling of motor vehicles. The global auto component market is a highly diversified sector that involves engine and auto component manufacturers, including aftermarket parts manufacturers, suppliers, dealers, and retailers. Manufacturing unit of auto components will have many sub branches. It must require organising of raw materials supplier, various manufacturing departments like E-treatment of metal, sub-components designer, finish match grinding, assembling, testing, inspection and calibration, stock-checking, finance, retailer, consultant, etc. The huge automobile industries import the auto components from smaller industries. This project is focused on one such small autocomponents manufacturing industries.

The Manufacturing head must maintain his department working along with staff and selling of the production. Therefore we need an application to organise the working of manufacturing industry, operations in demand and retailers and suppliers requirement. It must contain all the information and records of the day in and out working of the company. The application must give different access to visitor, buyer, workers and the owner.

### **1.3 OUTCOMES OF PROJECT WORK**

- Maintaining of records of employee,buyer,production factors.
- Buyers portal to search for auto components and place order.
- Productivity factors calculator.
- Workers salary management.
- Raw materials and output organisation.
- Workers division as per department responsible for production .

## CHAPTER 2

# REQUIREMENTS AND DESIGN

### 2.1 HARDWARE REQUIREMENTS

- **Processor :** Intel i3 or above
- **RAM :** 2 GB or above
- **Hard disk :** 200 GB or above
- **Monitor :** 15" color and above

### 2.2 SOFTWARE REQUIREMENTS

- **Operating system :** Windows 8 and above
- **Language and Skills :** Python ,SQL.
- **IDE :** Pycharm
- **Libraries:**Tkinter,Pillow,Python-SQL-Connector,Regular Expression
- **Prerequisites:**Knowledge of implementation of Python Programming Concepts and its libraries, Tkinter,Knowledge of DBMS Concepts and Queries in MySQL .

## CHAPTER 3

# ER MODEL AND RELATIONAL SCHEMA

### 3.1 ENTITY AND ATTRIBUTES

The basic concept that the ER model represents is entity with attributes. Entity is a thing or object in the real world with an independent existence. An entity may be an object with a physical existence or a unit. Each entity has attributes, which are the particular properties that describe it.

An entity set is a collection of similar types of entities, this set may contain entities with attribute sharing similar values. For example, a Students set contains all the students of a school, a Teachers set contains all the teachers of a school from all faculties. Entity sets need not necessarily be disjoint.

Entities have properties called attributes and all attributes have values. For example, a student entity has name, class, and age as attributes. There exists a domain or range of values for attribute values. For example, a student's name cannot be a numeric value, it has to be only alphabetic. A student's age cannot be negative or a word, etc.

#### Types of Attributes

1. Simple attribute – Simple attributes are atomic or further non divisible values. For example, a student's phone number is an atomic value of 10 digits, which can't be divided into parts.

2. Composite attribute – Composite attributes are made of more than one simple attribute, therefore can be further divided. For example, a student's complete name can be considered as firstname and lastname.

3. Derived attribute – Derived attributes are the attributes that do not exist in the physical database or not directly entered into database, but their values are derived by some means from other attributes present in the database. For example, AverageSalary

in a department should not be saved directly in the database, instead it can be derived from the salaries of the staff. For example, age can be derived from DOB.

4. Single-value attribute – Single-value attributes contain single value for the attribute.

For example – SocialSecurityNumber.

5. Multi-value attribute – Multi-value attributes contains more than one values. For example, a person can have more than one phone number, email address, preferences,etc.

These attribute types may be in combination like simple single-valued attributes,simple multi-valued attributes,composite single-valued attributes,composite multi-valued attributes

## 3.2 KEYS

Keys in DBMS refers an attribute or set of attributes which helps you to identify a row(tuple) in a relation(table). They help to find the relation between two tables. They help you uniquely identify a row in a table by a combination of one or more columns in that table and also also helpful for finding unique record or row from the tableThey are also helpful for finding unique record or row from the table.

**Types of Keys in Database Management System are:**

(1) Super Key - It is a group of single or multiple keys which identifies rows in a table.

(2) Primary Key - It is a column or group of columns in a table that uniquely identify every row in that table.

(3) Candidate Key - It is a set of attributes that uniquely identify tuples in a table.This is a super key with no repeated attributes.

(4) Alternate Key - It is a column or group of columns in a table that uniquely identify every row in that table.

(5) Foreign Key - It is a column that creates a relationship between two tables. This maintains data integrity and helps in navigation between two different instances of an entity.

(6) Compound Key - It is two or more attributes that allow you to uniquely recognize a specific record. It is possible that each column may not be unique by itself within the database records.

(7) Composite Key - It is an artificial key which aims to uniquely identify each record is called a surrogate key. These kind of key are said to be unique because they are created when you don't have any natural primary key in the database.

(8) Surrogate Key - It is an artificial key which is used to uniquely identify each record is called a surrogate key. These kind of key are said to be unique because they are created when you don't have any natural primary key in the database.

### **3.3 RELATIONSHIP AND PARTICIPATION**

**Relationship :** A relationship represents the association between any two related entity types. For example, 'Enrolled in' is a relationship that exists between Student entity and Course entity . In ER diagram, relationship type is represented by a diamond shape and connecting the entities with lines.A set of relationships of same type is termed as relationship set. The number of different entity sets participating in a relationship set is termed as degree of a relationship.

**Types:**

(1) Unary Relationship –When there is only one entity set participating in a relation, the relationship is termed as unary relationship.

(2) Binary Relationship –When there are TWO entities set participating in a relation, the relationship is termed as binary relationship.

(3) n-ary Relationship – When there are n entities set participating in a relation, the relationship is termed as n-ary relationship.

**Cardinality:** The number of times an entity of an entity set participates in a relationship set is termed as cardinality. Cardinality can be of following types:

(1) One to one – When each entity in each entity set can take part only once in the relationship, the cardinality is said to be one to one. Lets say a male can marry to one female and vice versa. Thus the relationship will be one to one.

(2) Many to one – When entities in one entity set can take part only compulsorily once in the relationship set and entities in other entity set can take part more than once in the relationship set, cardinality is said to be many to one. Lets say a student can take only one course but one course can be taken by many students. Thus the cardinality will be n to 1, which means that for one course there can be n students but for one student, there will be only one course. One to many is vice versa

(3) Many to many – When entities in all the entity sets can take part more than once in the relationship cardinality is said to be many to many. Lets say a student can take more than one course and one course can be taken by many students. Thus the relationship will be many to many.

**Participation Constraint:** A Constraint is applied on the entity participating in the relationship set.

**Types:**

(1) Total Participation – Each entity in the entity set must necessarily participate in the relationship. If each student must enroll in a course, the participation of student will be total(TP). In ER diagram, total participation (TP) is shown by double line .

(2) Partial Participation – The entity in the entity set may or may not necessarily participate in the relationship. If some courses are not enrolled by any of the student, the participation of course will be partial(PP).

### 3.4 PROBLEM STATEMENTS

Assumptions made are:

- 1)There is one manufacturing unit.
- 2)One manufacturing head maintains n number of departments.
- 3)Each department manufactures one product.
- 4) Each department has one stock manager,two shift managers and employees.  
Stock manager work under the main admin who is the head of the manufacturing head.Shift1 and Shift2 manager works under stock manager.Employees are divided into shift 1 and shift 2 ,and they work under respective shift manager.
- 5) Buyers can register to purchase the products.
- 6)Buyers can add products to the cart and buy anytime.
- 7)Buyers who wish to label the products in their name can do so by agreeing to pay additional charges.
- 8)Buyers can provide feedback to the company.
- 9)Buyers must get their billing details.
- 10) Workers can provide message to main admin.Main admin can provide salary bonus.
- 11) Stock manager manage the production quantity of his/her department.
- 12)Raw Input suppliers can tie up with the company.
- 13)Stock manager purchases the raw inputs from available suppliers.

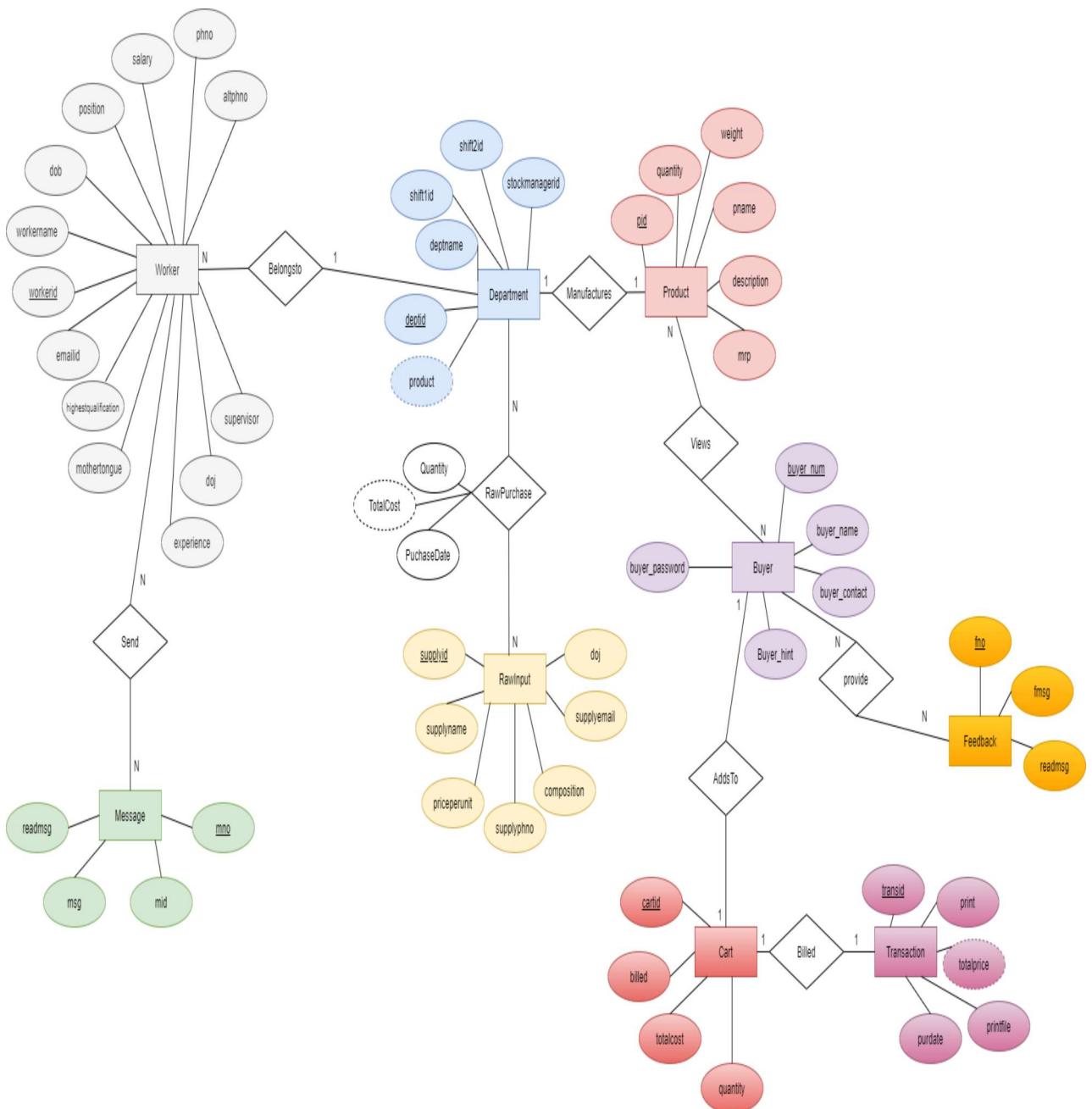
### 3.5 ER MODEL

Square-Entity

Oval-Attribute

Rhombus-Relationship

Underlined attribute-Primary key.



### 3.6 SQL

**1)desc worker;**

Field	Type	Null	Key	Default	Extra
workerid	varchar(10)	NO	PRI	NULL	
workername	char(30)	YES		NULL	
position	varchar(30)	YES		NULL	
salary	int(11)	YES		NULL	
dob	date	YES		NULL	
emailld	varchar(30)	YES		NULL	
phno	bigint(15)	YES		NULL	
altpphno	bigint(15)	YES		NULL	
highestqualification	varchar(40)	YES		NULL	
mothertongue	varchar(20)	YES		NULL	
experience	varchar(50)	YES		NULL	
doj	date	YES		NULL	
deptid	varchar(10)	YES	MUL	NULL	
supervisor	char(20)	YES		NULL	

**2)desc department;**

Field	Type	Null	Key	Default	Extra

## AUTO COMPONENTS MANUFACTURING UNIT

---

deptid	varchar(10)	NO	PRI	NULL	
deptname	varchar(30)	YES		NULL	
product	varchar(30)	YES		NULL	
prodid	varchar(10)	YES		NULL	
quantity	int(11)	YES		NULL	
stockmanagerid	varchar(10)	YES		NULL	
shift1id	varchar(10)	YES		NULL	
shift2id	varchar(10)	YES		NULL	
+.....	+.....	+.....	+.....	+.....	+.....

### 3)desc product;

+.....	+.....	+.....	+.....	+.....	+.....
Field	Type	Null	Key	Default	Extra
+.....	+.....	+.....	+.....	+.....	+.....
deptid	varchar(10)	NO	PRI	NULL	
deptname	varchar(30)	YES		NULL	
product	varchar(30)	YES		NULL	
prodid	varchar(10)	YES		NULL	
quantity	int(11)	YES		NULL	
stockmanagerid	varchar(10)	YES		NULL	
shift1id	varchar(10)	YES		NULL	
shift2id	varchar(10)	YES		NULL	
+.....	+.....	+.....	+.....	+.....	+.....

## AUTO COMPONENTS MANUFACTURING UNIT

---

4)desc cart;

Field	Type	Null	Key	Default	Extra
cartid	varchar(10)	NO	PRI	NULL	
buyerid	varchar(10)	YES		NULL	
prodid	varchar(10)	YES		NULL	
prodname	char(30)	YES		NULL	
mrp	int(11)	YES		NULL	
quantity	int(11)	YES		NULL	
totalcost	int(11)	YES		NULL	
billed	char(4)	YES		NULL	

5)desc transaction;

Field	Type	Null	Key	Default	Extra
transid	varchar(10)	NO	PRI	NULL	
sum	bigint(10)	YES		NULL	
print	int(11)	YES		NULL	
buyerid	varchar(10)	YES		NULL	
printfile	varchar(50)	YES		NULL	
totalprice	int(11)	YES		NULL	
purdate	date	YES		NULL	

## AUTO COMPONENTS MANUFACTURING UNIT

---

6)desc buyer;

+-----+-----+-----+-----+-----+	Field   Type   Null   Key   Default   Extra	+-----+-----+-----+-----+-----+
BUYER_NUM   int(5) unsigned zerofill   NO   UNI   NULL   auto_increment		
BUYER_NAME   char(20)   NO     NULL		
BUYER_CONTACT   bigint(10)   NO   PRI   NULL		
BUYER_PASSWORD   varchar(10)   NO     NULL		
BUYER_HINT   varchar(20)   NO     NULL		
+-----+-----+-----+-----+-----+		

7)desc rawinput;

+-----+-----+-----+-----+-----+	Field   Type   Null   Key   Default   Extra	+-----+-----+-----+-----+-----+
BUYER_NUM   int(5) unsigned zerofill   NO   UNI   NULL   auto_increment		
BUYER_NAME   char(20)   NO     NULL		
BUYER_CONTACT   bigint(10)   NO   PRI   NULL		
BUYER_PASSWORD   varchar(10)   NO     NULL		
BUYER_HINT   varchar(20)   NO     NULL		
+-----+-----+-----+-----+-----+		

## AUTO COMPONENTS MANUFACTURING UNIT

---

8)desc rawpurchase;

Field	Type	Null	Key	Default	Extra
BUYER_NUM	int(5) unsigned zerofill	NO	UNI	NULL	auto_increment
BUYER_NAME	char(20)	NO		NULL	
BUYER_CONTACT	bigint(10)	NO	PRI	NULL	
BUYER_PASSWORD	varchar(10)	NO		NULL	
BUYER_HINT	varchar(20)	NO		NULL	

9)desc message;

Field	Type	Null	Key	Default	Extra
mno	int(11)	NO	PRI	NULL	auto_increment
mid	varchar(10)	YES		NULL	
msg	varchar(100)	YES		NULL	
readmsg	char(4)	YES		no	

**10) desc feedback;**

Field	Type	Null	Key	Default	Extra
fno	int(11)	NO	PRI	NULL	auto_increment
fmsg	varchar(100)	YES		NULL	
readmsg	char(4)	YES		no	

## CHAPTER 4

# PYTHON FUNDAMENTALS

### 4.1 INTRODUCTION TO PYTHON.

#### Representations:

The Python fundamental consists of the basic building blocks of Python programming language. And it is basically divided into the following categories as follows.

- Statements
- Indentations
- Comments
- Variables
- Constants
- Tokens

#### (1) Statements:

They are logical instructions that interpreter can execute and read them, it can also be both single and multiline. The two categories of the Python Statements are:

**(a) Expression Statement:** By the help of expression statements, we can perform the operations like addition, subtraction, concentration and many more. In the short, the statement has return value. It is an expression that appears on the right side of the assignment, as a parameter to method call.

**(b) Assignment Statement:** By the help of the assignment statements we can create new variables, assign values and also we can change values. Assignment statements are categorized into three:

- Value-Based Expressions on Right hand side
- Current Variables on Right hand side

- Operation on Right hand side

## **(2) Indentation:**

The programming languages python uses indentation to mark a block of the code. Most of the Programming languages provide indentation for better code formatting and doesn't enforce to have it. But mainly in Python it is mandatory. That's why indentation is crucial in Python.

## **(3) Comments:**

Comments are basically nothing but tagged lines of in codes which increases the readability of the code and make the code self-explanatory. There are two categories of Comments:

- Single line Comments: '#' by the help of these we begin a single-line comment.
- Multi-line comments: "..." by the help of these we write multiline comments in python.
- Doctstring comments: The documentation string in Python gives programmers an easy way of adding quick notes with every Python module, functions, class and method. Multiline comments are using triple quotation in strings.

## **(4) Variables:**

In Python variable is a memory address that can change, when a memory address cannot change then it is known as constant. Variable is the name of the memory location where the data is stored. Once the variable is stored then space is allocated in memory. It also defines the variable using a combination of numbers, letters, and the underscore character.

## **(5) Constants:**

In Python constants is a type of variable that holds values, whose value cannot be changed. We rarely use constants in Python.

**(6) Token:**

In Python tokens are the smallest unit of the program. Python contains the following tokens:

**(a) Reserved words:**

Reserved words are nothing but a set of special words, which are reserved by python and also have a specific meaning. Here, in Python we are not allowed to use keywords as variables. Reserved words are case sensitive in Python. For example: False, if, none, import, True, in, and, def, return, elif, try, else, while, except, with, finally, yield, is, as, break, class, etc Identifiers: In Python identifiers are nothing but user-defined names to represent programmable entity like variables, functions, modules, classes. There are few rules that we need to follow while defining an identifier. They are:

- i) We can use a sequence of letters lowercase or uppercase. We can also mix up digits or an underscore while defining an identifier.
- ii) We cannot use digit to begin an identifier name.
- iii) We should not use reserved keywords to define an identifier.
- iv) You are not allowed to use any other special characters other than underscore.
- v) Even though python doc says that you can name an identifier with unlimited length.

**Literals:**

The Literals can be defined as data that is given in a variable or constant. The following literals are as following:

- (i) String Literals: String literals are a sequence of characters surrounded by quotes. Single, double or triple quotes can be used for a string.
- (ii) Boolean literals: Boolean literal can have any of two values i.e. true or false.
- (iii) Numeric literals: They are immutable. Numeric literals can belong to three different numerical types Integer, Float, Complex.

(iv) Collection literals: The four types of collection literals are List literals, Tuple literals, Dict literals, and Set literals.

(v) Special literals: Python basically contains one special character that is none.

**Operators:**

In python operators are the symbols which perform the operation on some values. The following are the known operators in the Python.

- Arithmetic Operators
- Relational Operators
- Assignment Operators
- Logical Operators
- Membership Operators
- Identity Operators
- Bitwise Operators

## 4.2 DATASTRUCTURES

**(A) Python Lists:**

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets([]). To some extent, lists are similar to arrays in C. One difference between list is that all the items belonging to a 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 –

```
lst = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
```

```
print(lst) # Prints complete list
```

**(B) Python Tuples:**

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses. The main differences between lists and tuples are: Lists are enclosed in brackets [ ] and their elements and size can be changed, while tuples are enclosed in parentheses ( ) and cannot be updated. Tuples can be thought of as read-only lists.

For example –

```
tple = ( 'abcd', 786 , 2.23, 'john', 70.2 )
```

```
print(tple) # Prints the complete tuple
```

**(C) Python Dictionary :**

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object. Dictionaries are enclosed by curly braces({}) and values can be assigned and accessed using square braces ([]).

For example–

```
dct = {}
```

```
dct ['one'] = "This is one"
```

```
dct[2] = "This is two"
```

```
print(dct)
```

## 4.3 TKINTER WIDGETS

The tkinter package (“Tk interface”) or library is the standard Python interface to the Tk GUI toolkit.

### LABEL WIDGET:

A label widget Shows any text to the user. A text can be added programmatically or by default to label to display a text on the screen for the user. A label can be placed on the window or a fame or a canvas.

### BUTTON WIDGET:

A Button can be on and off. When a user clicks it or selects it, the button emits an event. Images can be displayed on buttons too. A button has a command and a function associated with it to perform an action when it is passed.

### ENTRY WIDGET:

An Entry widget is used to accept a text input from the user. The obtained input text can be stored in avariable and used for several operations . The entry box can be placed on the window or frame and can accept values of several types such as StringVar(), IntVar() and DoubleVar().

### SCALE WIDGET: HORIZONTAL

Scale widget is used to have a slider that goes from one value to another. The starting, ending and step values can be set. The current value of the slider can be accessed by its Get method and the current value can be set by its set method.

### LISTBOX WIDGET:

Listbox displays a list of items and lets the user choose from one set of options. The list can be of various types and the current value of the list box which is selected can be bound to an event and is represented by <<ListSelect>>.

## CHAPTER 5

# IMPLEMENTATION

## 5.1 LIBRARIES

### 5.1.1 MYSQL IN PYTHON

“python-mysql-connector” was installed and “mysql.connector” was imported.

```
try:  
    mydb=mysql.connector.connect(user='root', password='kavya',host='127.0.0.1',database='ManufacturingUnit',autocommit=True)  
    mycursor=mydb.cursor(buffered=True)  
  
    mycursor.execute("USE ManufacturingUnit")  
    mycursor.fetchone()  
except:  
    messagebox.showwarning("No Connection.", "Error in connecting to database.")
```

Fig.no.1 SQL CONNECTION

- To execute: mycursor.execute(sqlstatement)
- To get row count or number of tuples present in given table: mycursor.rowcount
- To receive column names: mycursor.description
- To get tuples: mycursor.fetchall()

### 5.1.2 Tkinter

Label, Button, Balloon, MessageBox, Entry and Textbox Widgets are used.

### 5.1.3 tkcalendar

To select date from calendar.

```
def setDate():
    pw.delete(0, 'end')
    d=myCal.get_date()
    pw.insert(0,d)

choose.destroy()
myCal.destroy()

def getDate():
    global myCal
    myCal = Calendar(L_Frame, setmode='day', date_pattern='yyyy-mm-dd')
    myCal.place(x=200,y=270)
    global choose
    choose=Button(L_Frame, text="Choose", command=setDate, font=("arial", 10, "bold"), bd=5, relief=GROOVE)
    choose.place(x=450, y=230)
```

Fig.no.2 tkcalendar

### 5.1.4 re(Regular Expression)

To verify the contact number while registering of buyers in the application.

```
def register():
    bname=name.get()
    print(bname)
    bcontact=int(contact.get())
    pattern = re.compile("[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]")

    bpwd=passwd.get()
    bhint=hint.get()
    if bname=="" or bpwd=="" or bhint=="":
        messagebox.showwarning("PLEASE CHECK", "Enter ALL DETAILS")
        New_user(root)
    else:
        if not pattern.match(str(bcontact)):
            messagebox.showwarning("NOT VALID", "Enter proper mobile number")
            return
```

Fig.no.3 Regular Expression

### 5.1.5 PILLOW-ImageTk

Library used to read and display background image and product image.

```
self.bg_icon = ITk.PhotoImage(file="background.jpg")
bg_lbl = Label(self.root, image=self.bg_icon).pack()
title = Label(self.root, text="AUTO COMPONENTS MANUFACTURING UNIT", font=("times new roman", 40, "bold"),
              fg="red", bg="white", relief=GROOVE)
title.place(x=0, y=30, relwidth=1)
self.ul_icon = ITk.PhotoImage(file="userlogin.jpg")
```

Fig.no.4 ImageTk

### 5.2 CODE IMPLEMENTATION

The concept of inheritance is used here: class Base is the parent class for all the other classes. To inherit the data of parent class, in the `_init` function constructor, "super" keyword is used. Appropriate error messages are displayed using message box according to operation. Database connectivity is included with exception handling concepts of Python.

Lets us see the implementation according to the classes:

(i) class Base: It initializes the basic background ,the title, the dimensions and the common widgets of the application. It has the frame "DROP HERE", where buyers can provide their feedback and the workers in the manufacturing unit can give their message to the manufacturing head.

(ii) class Login: The application starts with this page . This allows verification of user login and leading to his/her shopping cart page. It has options for new buyer registration, changing of password and separate login for people belonging to the the manufacturing unit. Username and password verification is done.

(iii) class Newuser: It allows registration of new customers. It takes information such as valid name, phone number, password and password hint(First school name). Checking for valid phone number and entering of all details is done.

## AUTO COMPONENTS MANUFACTURING UNIT

---

(iv) class passwd:It allows to customer to change the password,only if he/she is verified.The username and the answer to hint must be answered correctly.

(v) class memlogin:It allows people working at organization to login using their worker id,date of joining and role played in the organization(Main admin,Stock Manager,Shift manager,Employee).

(vi) class view:Once the user signs in,he is lead to the shopping page.Here he views the products(image,code,name,description,weight,mrp).If he enters the quantity and adds to the cart,this item will be added to the cart.The items in the cart remain as it is,till it is billed.If user wants to bill,he will be confirmed with the payment .If customer want their products to be labelled in their name,additional charges are applicable.The bill is written on to a text file using file I/O operations.

```
outF = open(filestr, "w")
outF.write("AUTO COMPONENTS MANUFACTURING UNIT")
outF.write("\n")
outF.write("-----\n")
outF.write("Transaction ID:"+billstr)
outF.write("\n")
outF.write("Name:"+str(f1[0][0]))
outF.write("\n")
outF.write("Contact:"+str(f1[0][1]))
outF.write("-----\n")
outF.write("|%6s|%15s|%7s|%5s|%8s|\n%("CODE", "NAME", "MRP(Rs)", "Quantity", "TotalCost(Rs)")
outF.write("-----\n")
for i in range(r_rows):
    outF.write("|%6s|%15s|%7d|%5d|%8d|\n%("r_set[i][0],r_set[i][1],r_set[i][2],r_set[i][3],r_set[i][4]))
outF.write("-----\n")
outF.write("Total:Rs."+str(sum))
outF.write("\n")
outF.write("Printing Price:Rs."+str(pri))
outF.write("\n")
outF.write("Printing name:"+str(pr.get()))
outF.write("\n")
outF.write("-----\n")
outF.write("Total amount paid:Rs."+str(total))
outF.close()
```

Fig.no.5 File Output

(vii) class admin:This is the main admin or the manufacturing head page which enables the operations and complete control of the application.It involves reading unread feedbacks from customers and the unread messages from the workers.Calculating the expenditure and the income of the manufacturing unit.Viewing of all workers and making changes in the salary.

(viii) class stock:This is the Stock Manager page which enables him/her to manage the shift managers and employees.He/She can view the available raw input suppliers.After purchase,he/she can update the information.He/She can add the quantity that is produced by the department. View his/her salary and other details.

(viii) class shift: This is the Shift Manager page which enables him/her to manage the employees working in his/her shift. View his/her salary and other details.

(ix) class Employee:View his/her salary and other details.

## CHAPTER 6

## RESULTS

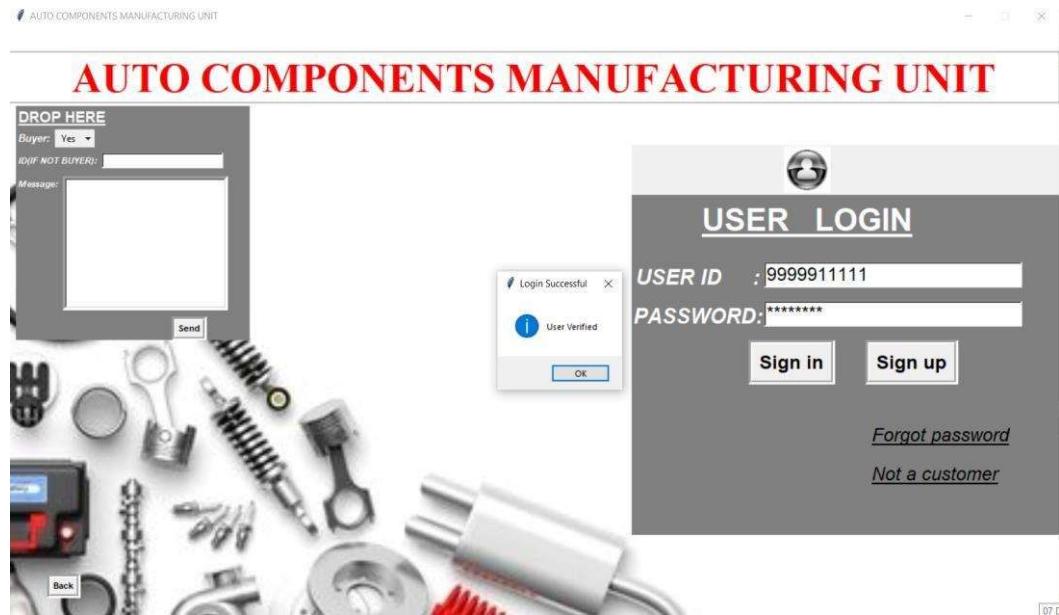


Fig.no.6 Registered Buyer Login

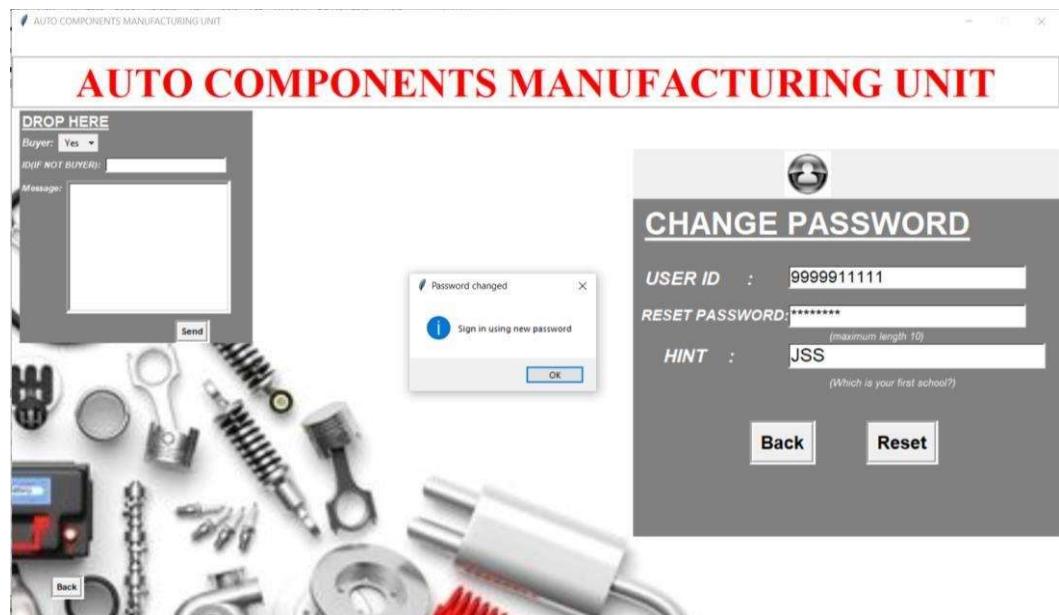


Fig.no.7 Registered Buyer Change Of Password

## AUTO COMPONENTS MANUFACTURING UNIT

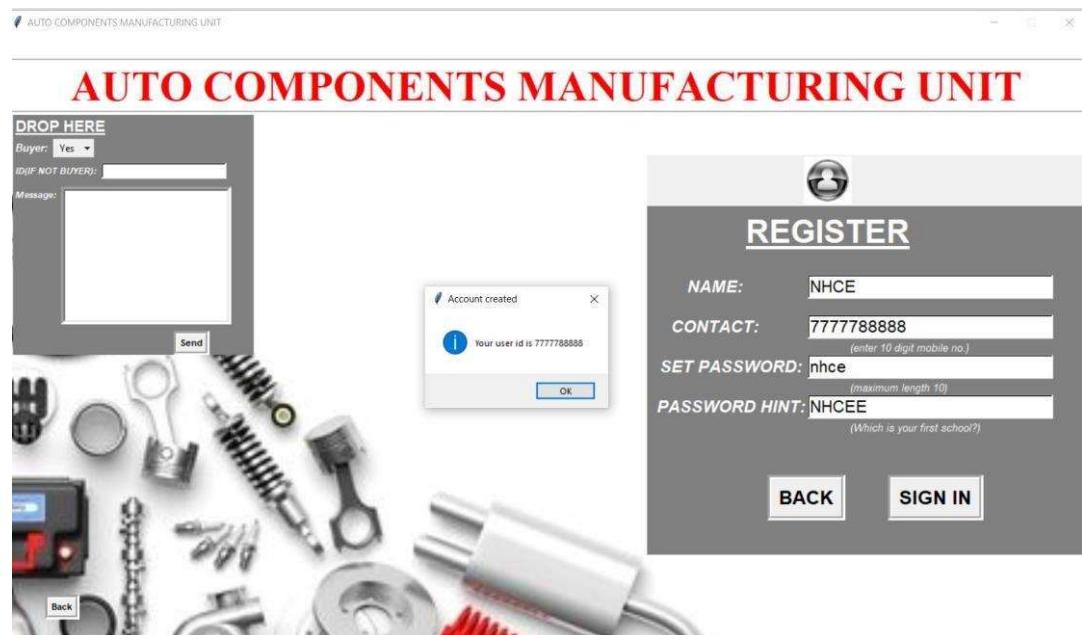


Fig.no.8 New User Registration

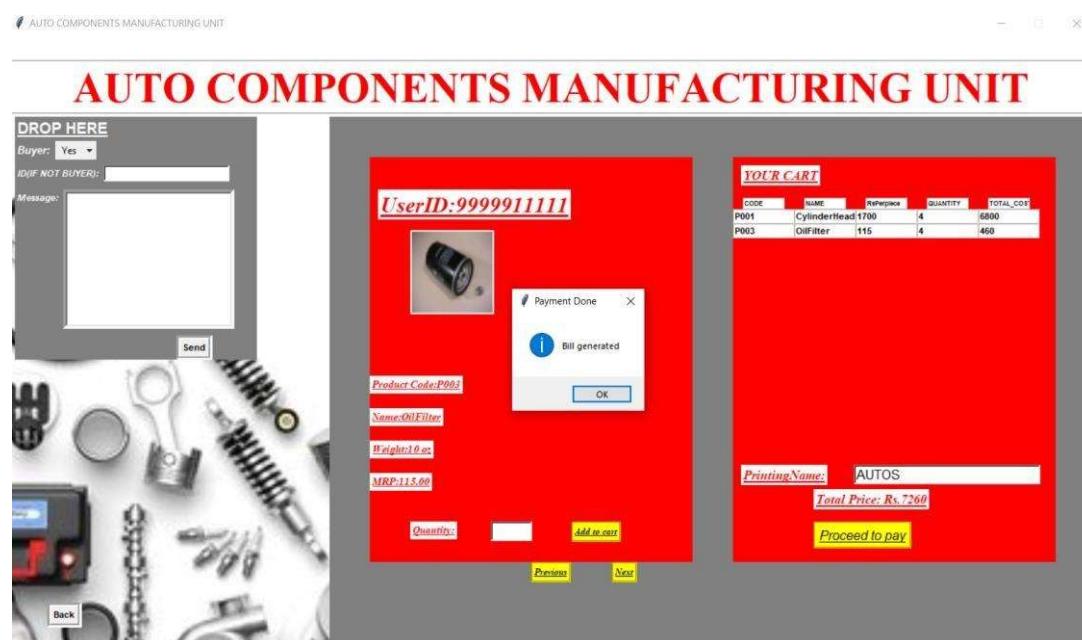


Fig.no.9 Shopping Page

## AUTO COMPONENTS MANUFACTURING UNIT

```
tpy x T19.txt x T17.txt x T14.txt x T10.txt x
supporting *.txt files found.
AUTO COMPONENTS MANUFACTURING UNIT
-----
Transaction ID:T10
Name:GAGAN
Contact:9919919911-----
| CODE| NAME|MRP(Rs)|Quantity|TotalCost(Rs)|
-----
| P001| CylinderHead| 1700| 4| 6800|
| P004| PetrolEngine| 15900| 4| 63600|
| P003| OilFilter| 115| 6| 690|
| P002| Balanceshaft| 1300| 8| 10400|
-----
Total:Rs.81490
Printing Price:Rs.44
Printing name:bosch
-----
Total amount paid:Rs.81534
```

Fig.no.10. Bill



Fig.no.11.Buyer's Feedback



Fig.no.12.Worker's Message



Fig.no.13. Main Admin Login



Fig no.14. ADMIN PAGE-View Feedback



Fig.no.15. ADMIN PAGE-View Messages

## AUTO COMPONENTS MANUFACTURING UNIT

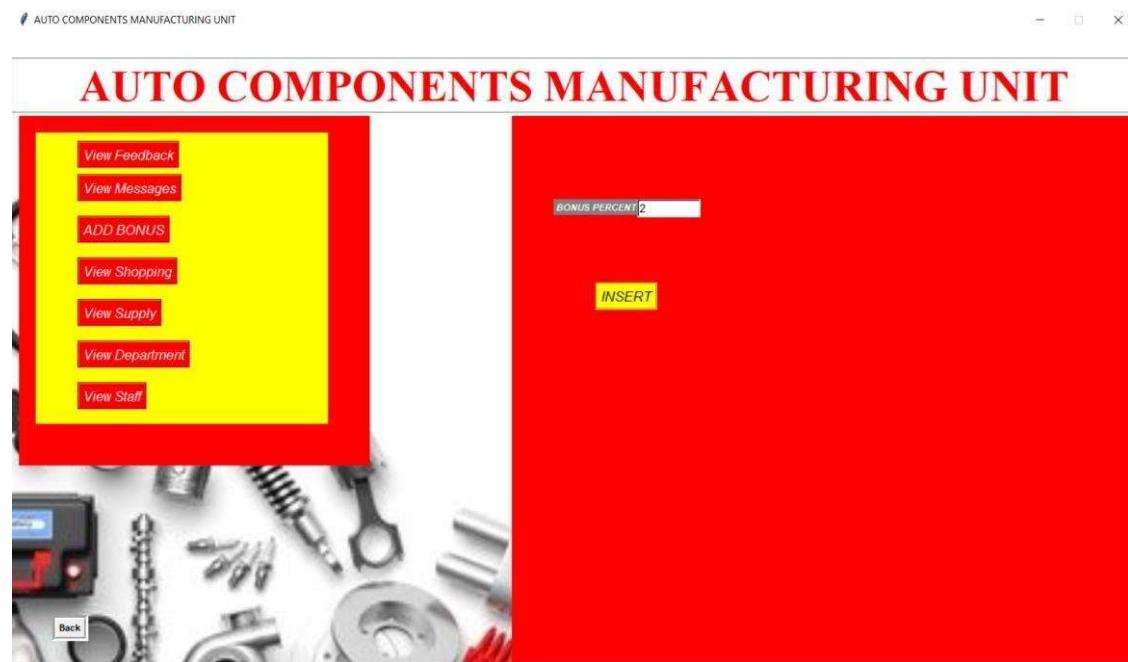


Fig.no.16 ADMIN PAGE-Add Bonus

```
mysql> select distinct(position),salary from WORKER;
+-----+-----+
| position      | salary |
+-----+-----+
| STOCK MANAGER | 75748 |
| SHIFT 1 MANAGER | 54106 |
| SHIFT 2 MANAGER | 54106 |
| EMPLOYEE       | 43285 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select distinct(position),salary from WORKER;
+-----+-----+
| position      | salary |
+-----+-----+
| STOCK MANAGER | 77263 |
| SHIFT 1 MANAGER | 55188 |
| SHIFT 2 MANAGER | 55188 |
| EMPLOYEE       | 44151 |
+-----+-----+
4 rows in set (0.00 sec)
```

Fig.no.17. Data Before and After Salary Bonus Update

## AUTO COMPONENTS MANUFACTURING UNIT

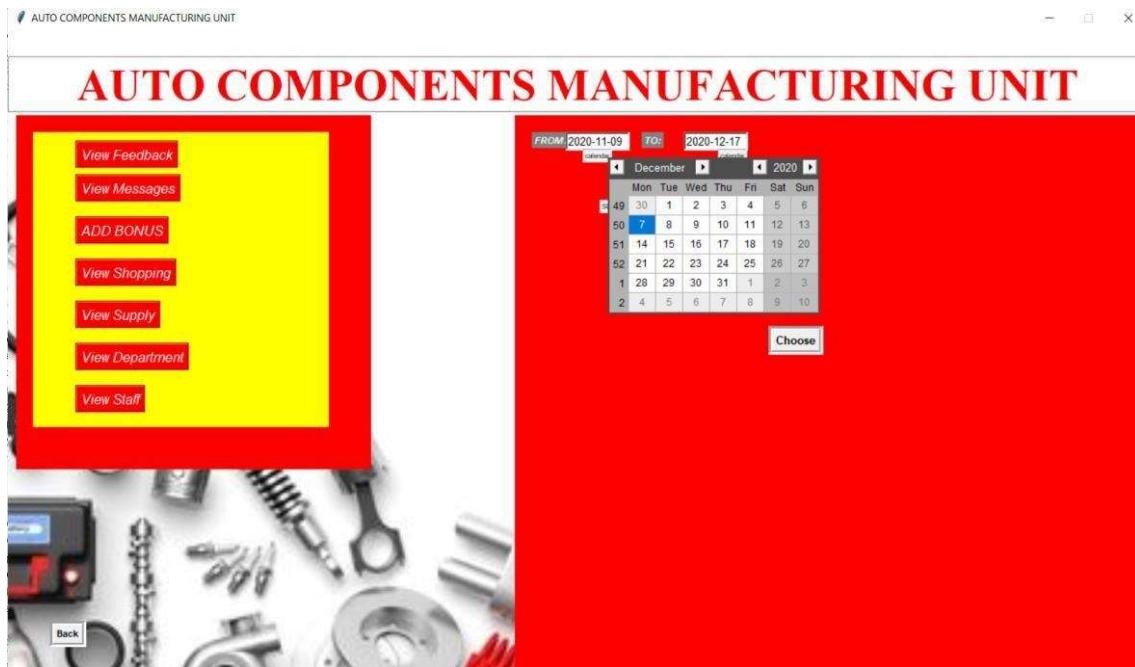


Fig.no.18. ADMIN PAGE-View Shopping

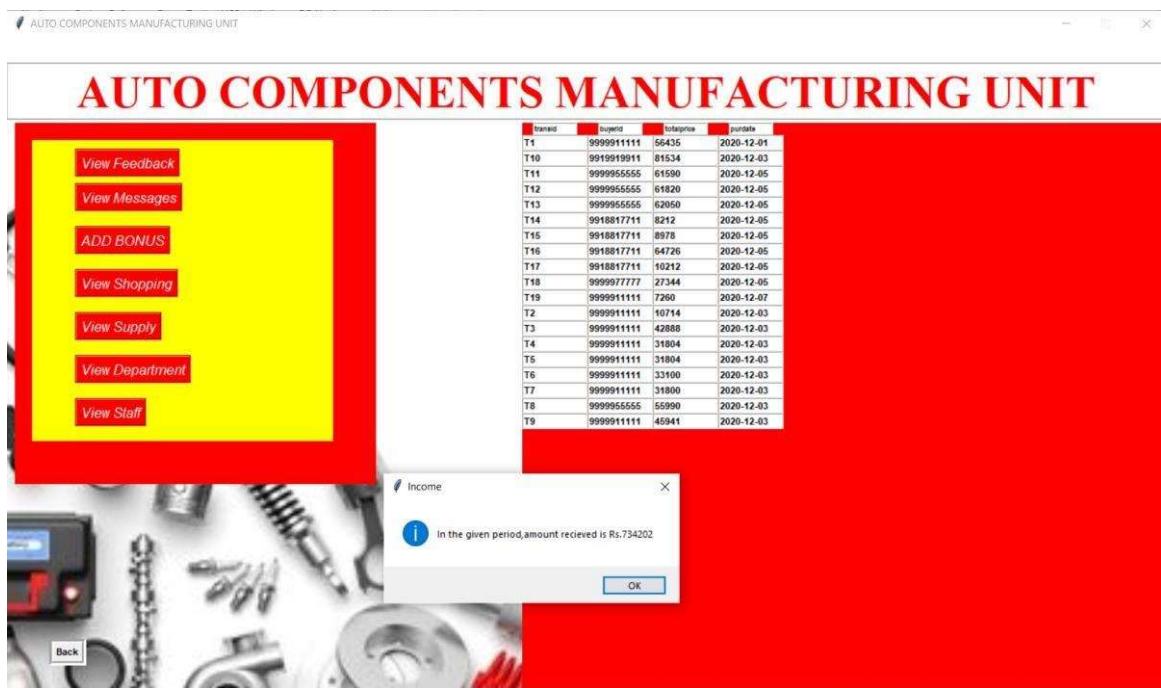


Fig.no.19. ADMIN-PAGE Income in the given period

## AUTO COMPONENTS MANUFACTURING UNIT

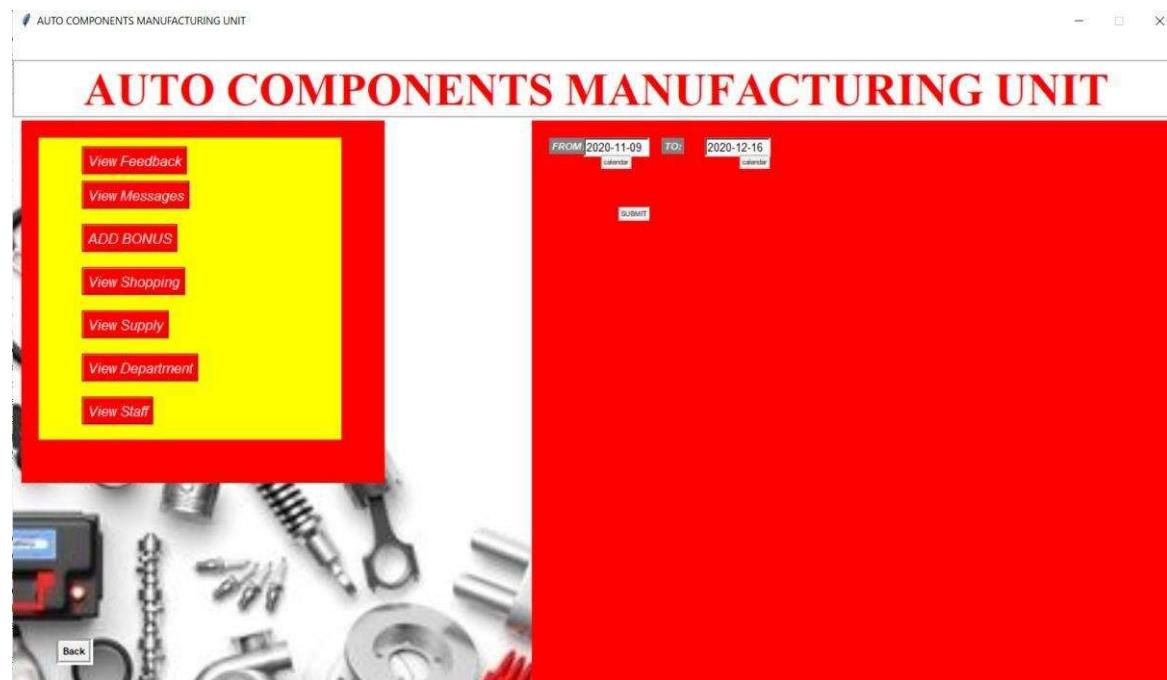


Fig.No.20. ADMIN PAGE-View Supply

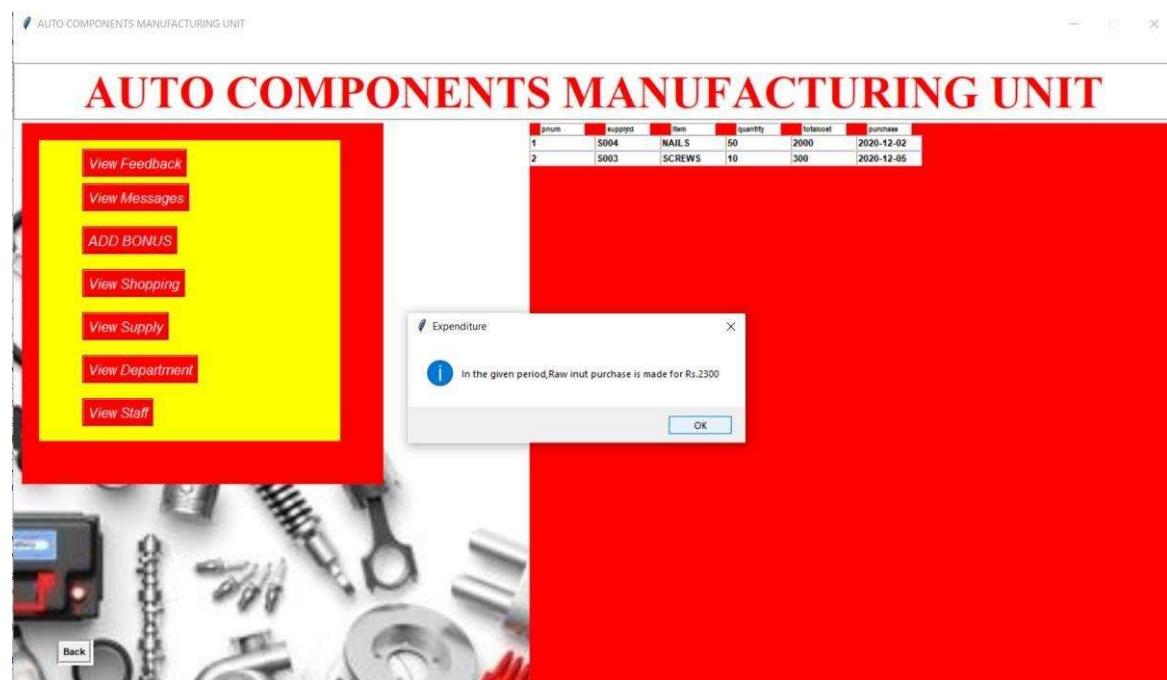


Fig.No.21.ADMIN PAGE- Expenditure

## AUTO COMPONENTS MANUFACTURING UNIT

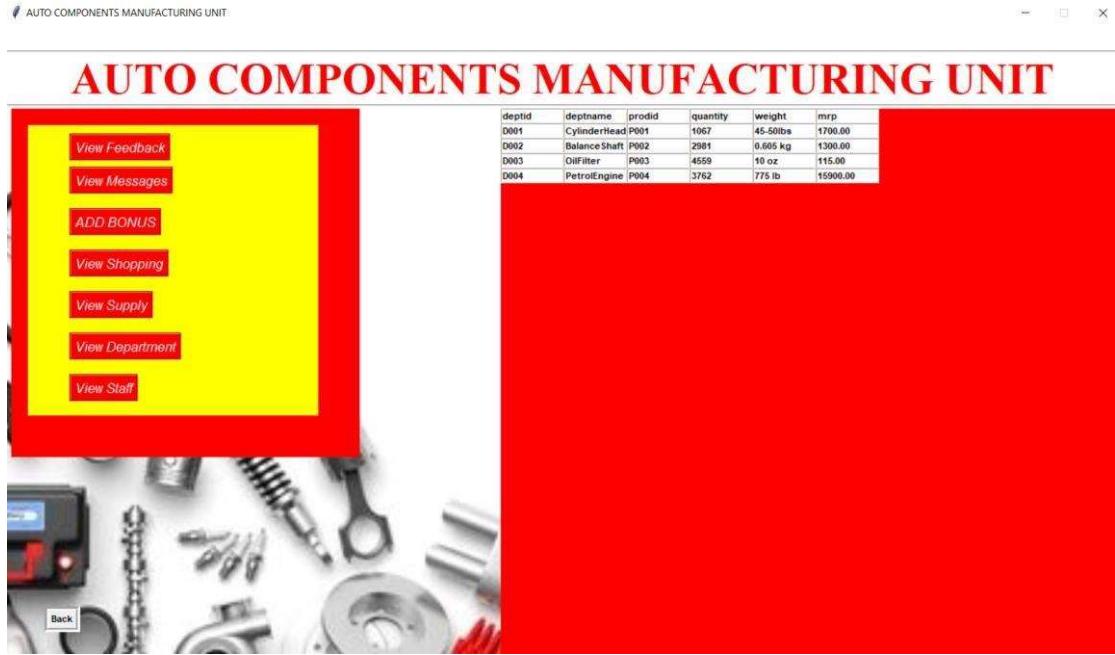


Fig.no.22. ADMIN PAGE-View Department



Fig.no.23. ADMIN PAGE-View Staff

## AUTO COMPONENTS MANUFACTURING UNIT

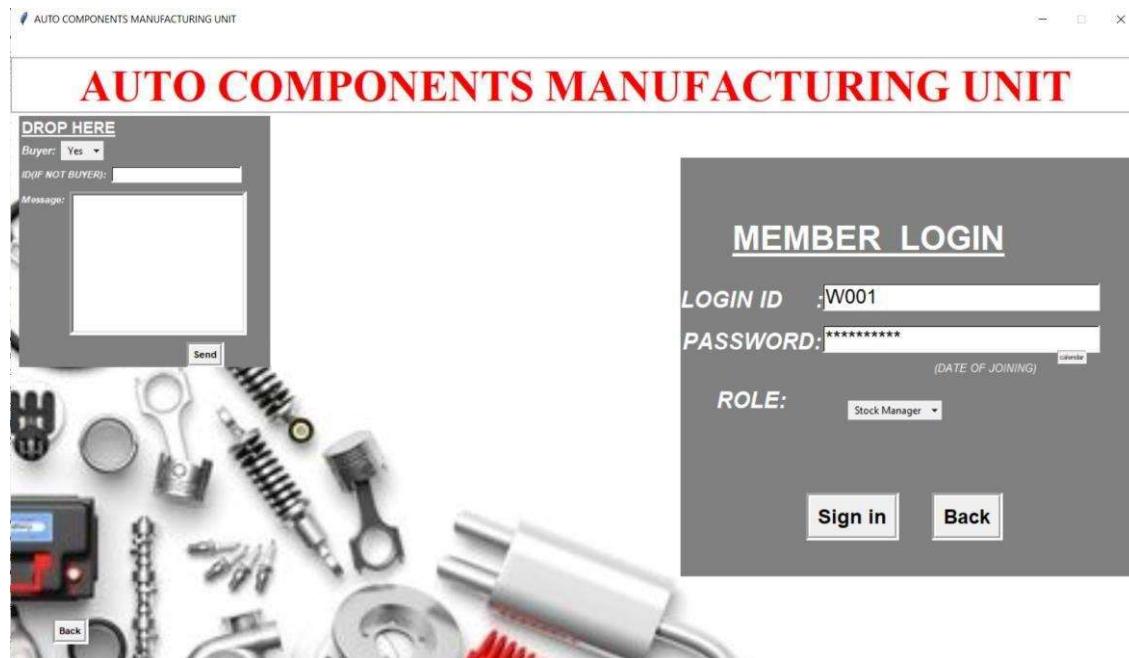


Fig.no.24 Stock Manger Login

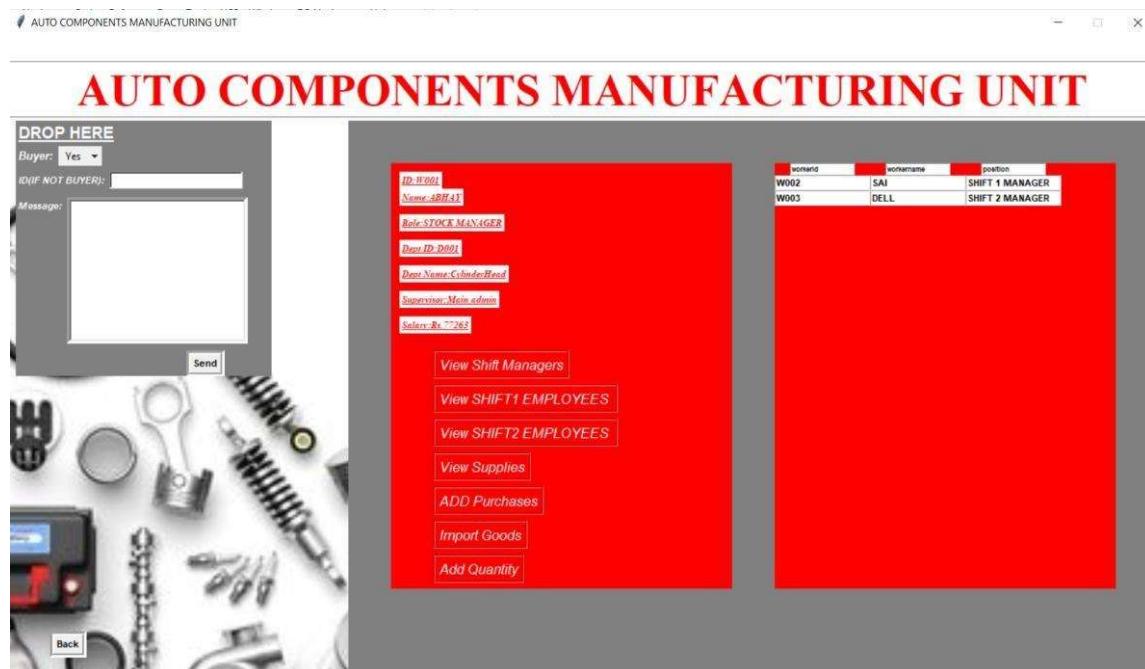


Fig.no.25. Stock Manger Page-View Shift Managers

## AUTO COMPONENTS MANUFACTURING UNIT

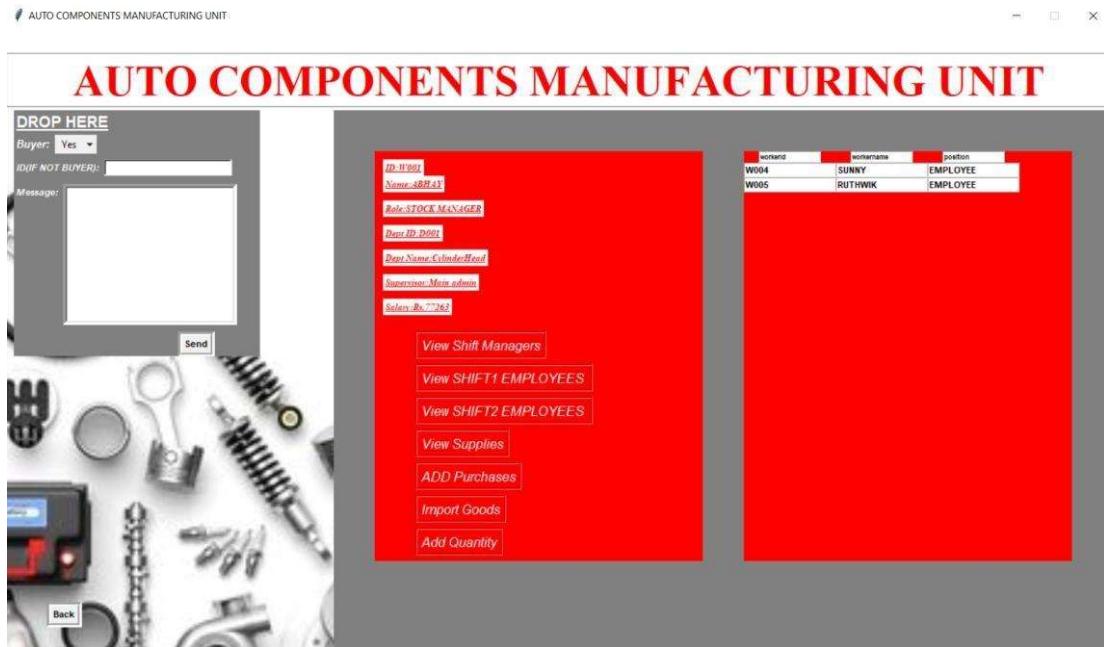


Fig.no.26. Stock Manger Page-View Shift 1 Employees

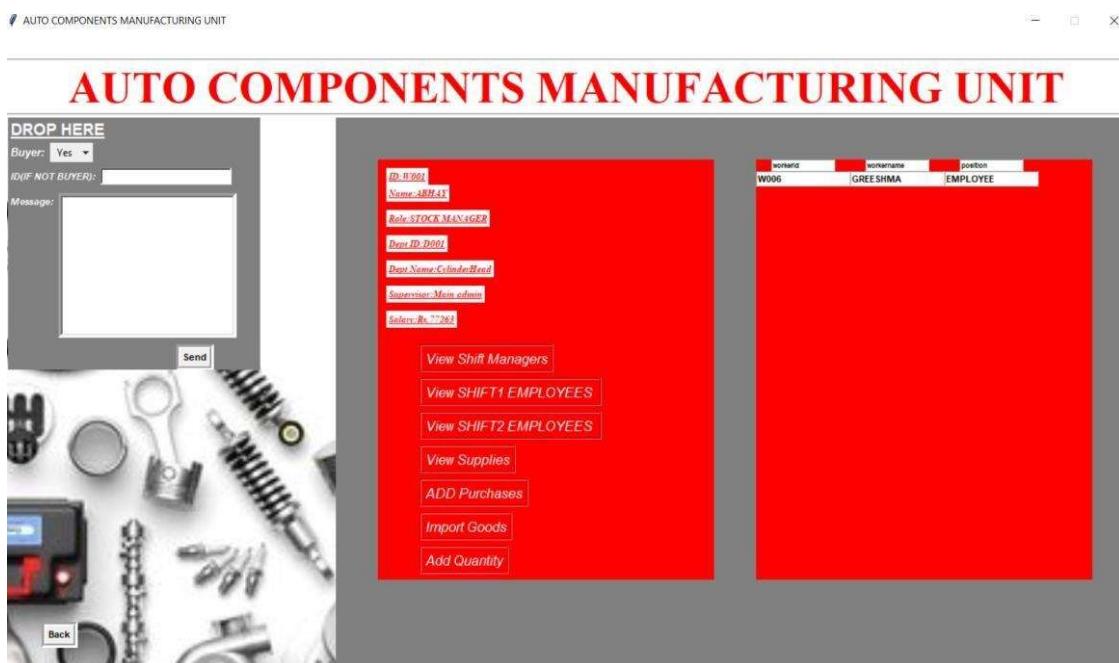


Fig.no.27. Stock Manger Page-View Shift 2 Employees

## AUTO COMPONENTS MANUFACTURING UNIT

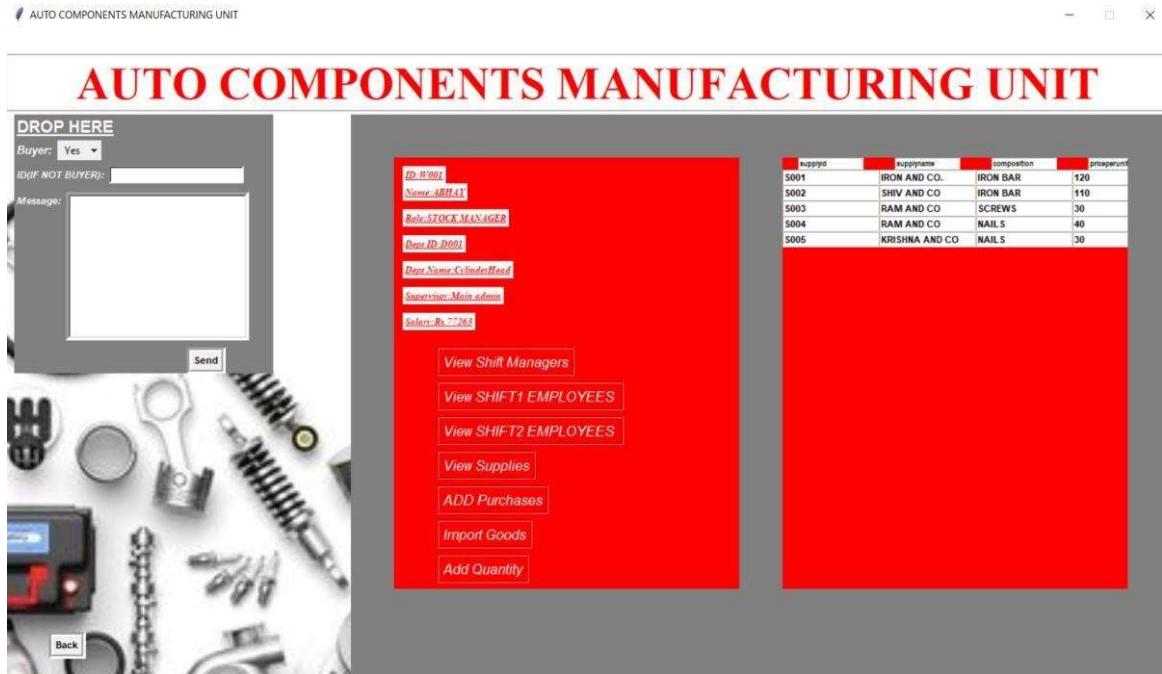


Fig.no.28. Stock Manger Page-View Supplies



Fig.no.29. Stock Manger Page-Add Purchase

## AUTO COMPONENTS MANUFACTURING UNIT

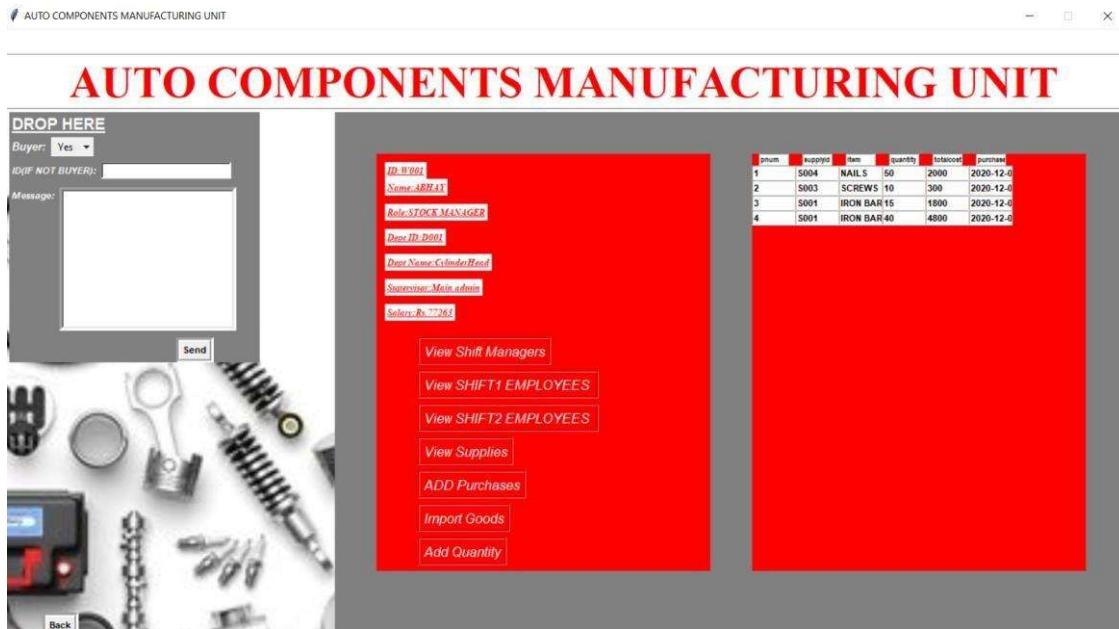


Fig.no.30. Stock Manger Page-Import Goods

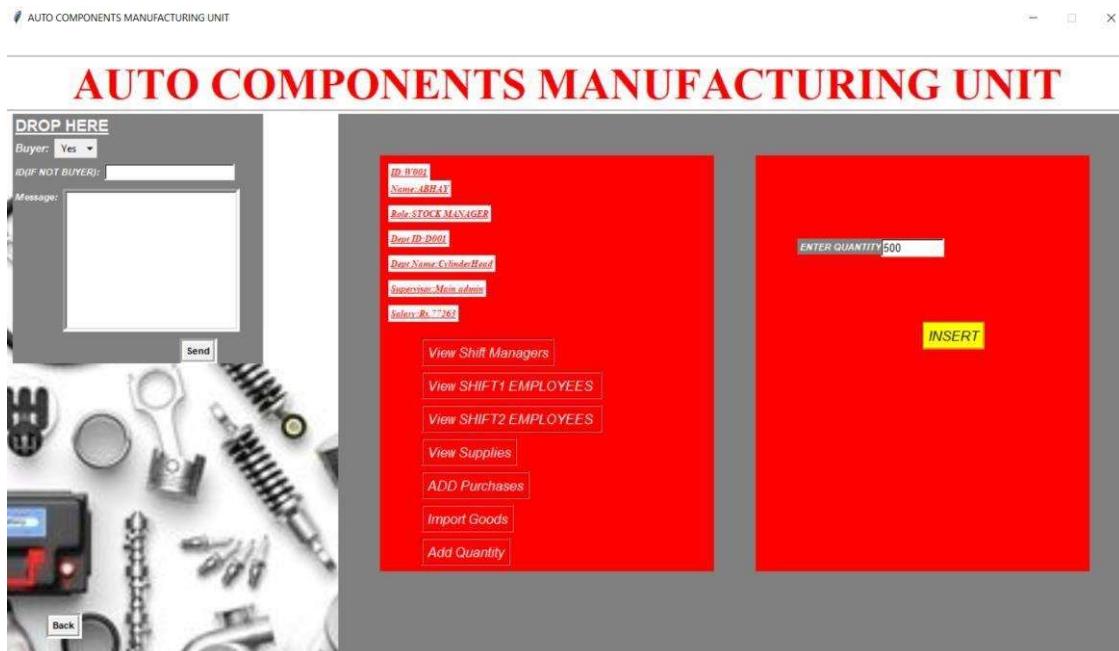


Fig.no.31. Stock Manger Page-Add Quantity

```
mysql> SELECT quantity from department where deptid="D001";
+-----+
| quantity |
+-----+
|    1067 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT quantity from department where deptid="D001";
+-----+
| quantity |
+-----+
|    1567 |
+-----+
1 row in set (0.00 sec)
```

Fig.no.32. Data before and after -Adding Quantity



Fig.no.33 Shift Manager Login

## AUTO COMPONENTS MANUFACTURING UNIT

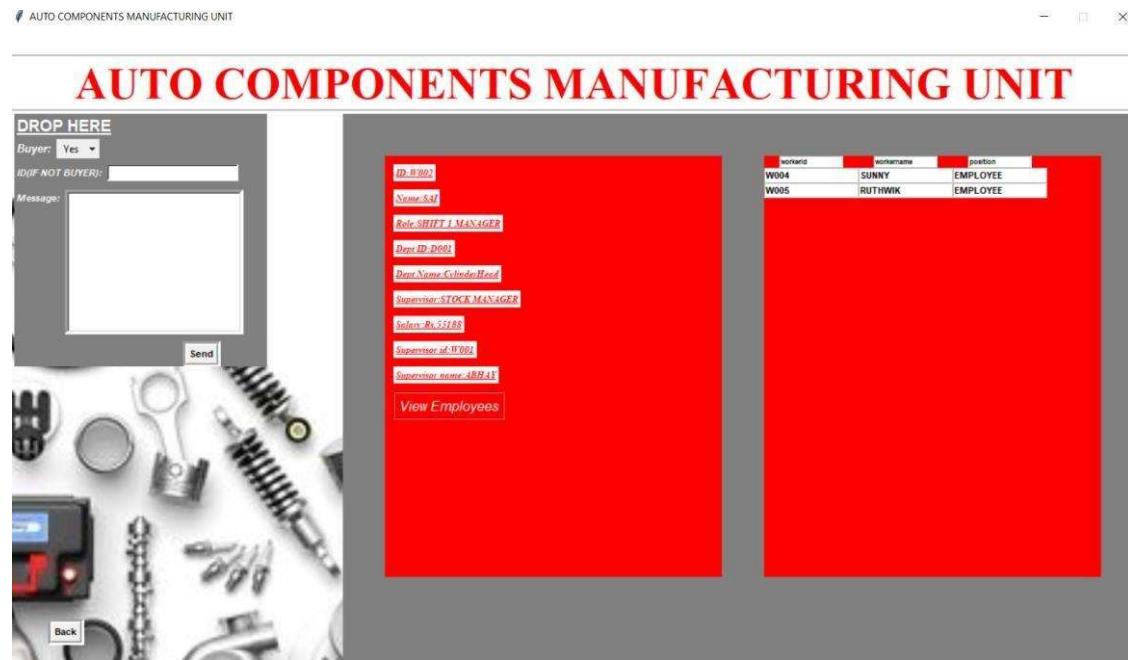


Fig.no.34. Shift Manager Page-View Employees

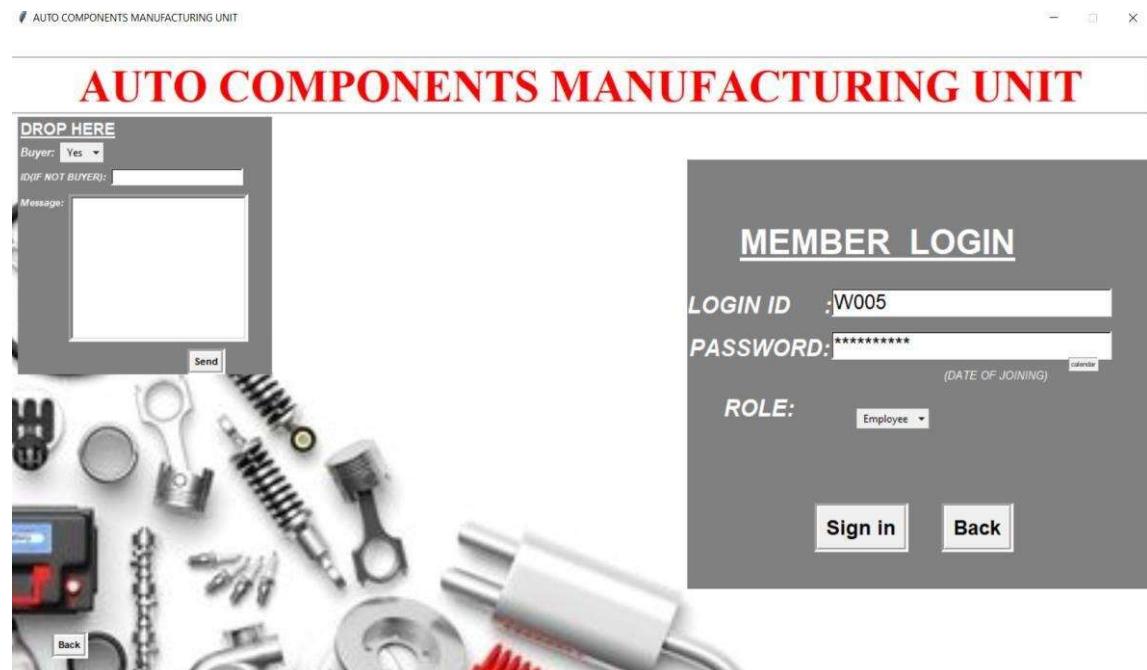


Fig.no.35. Employee Login

## AUTO COMPONENTS MANUFACTURING UNIT

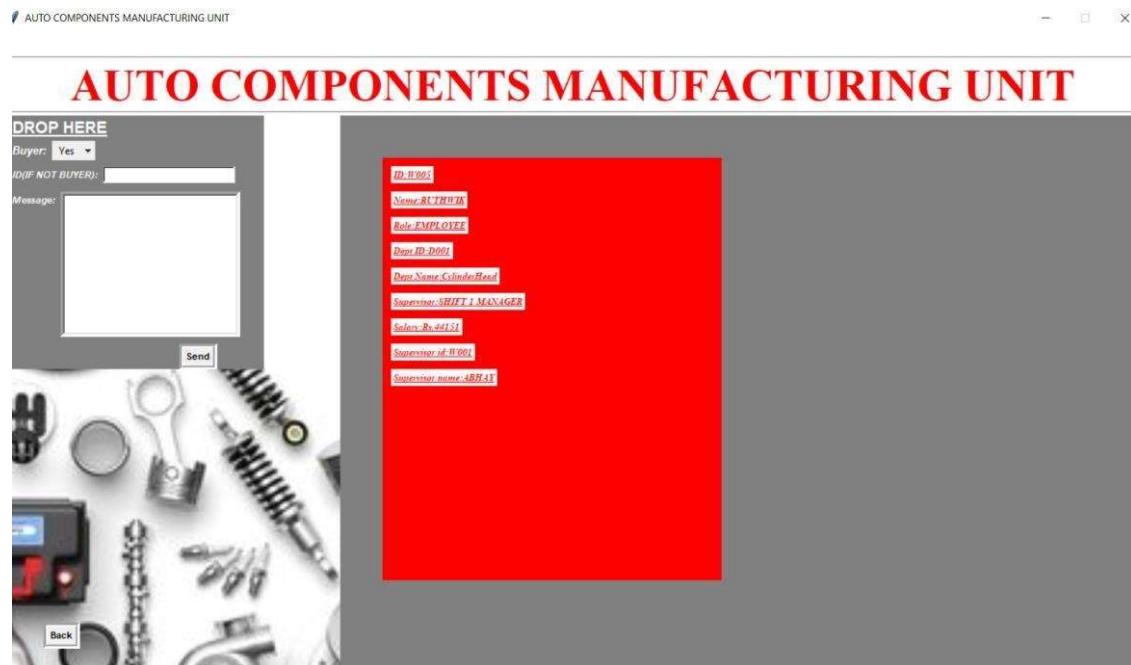


Fig.no.35. Employee Page

## CHAPTER 8

### CONCLUSION

Auto Components Manufacturing Unit is the project focused to organize a small scale industry that manufactures sub auto components for huge automobile industries. It involves the idea of labeling the name of the buyer on the product. The items added to cart remains saved until billed. After successful payment, the bill is written on to a file. The buyers can provide feedback to the company. It resembles the hierarchical model of the working of the organization. The manufacturing head runs the manufacturing unit which involves many departments. Each department manufactures one product. Each department is headed by Stock manager who maintains the raw input and also the productivity from the department. The Stock manager supervises the operations of the two Shift managers in his/her department. The Shift manager in turn supervises the working of the employees under his/her shift. The workers can share their views to the head of the manufacturing unit.

Thus this is an application that resembles an auto components manufacturing unit and its operations. The frontend is framed using python, tkinter and other libraries. The database support is by MySQL, which is connected using python-mysql-connector.

## REFERENCES

1. <https://www.geeksforgeeks.org/python-programming-language/>
2. <https://www.w3schools.com/python/>
3. <https://docs.python.org/>
4. <https://www.programiz.com/python-programming/methods>
5. [https://www.w3schools.com/sql/sql\\_quickref.asp](https://www.w3schools.com/sql/sql_quickref.asp)
6. <https://dev.mysql.com/doc/refman/en/>
7. [https://en.wikipedia.org/wiki/Crankshaft\\_position\\_sensor](https://en.wikipedia.org/wiki/Crankshaft_position_sensor)
8. Ramez Elmasri and Shamkant B. Navathe: Fundamentals of Database Systems, 7th Edition, Pearson ,2016.
9. Johannes Gehrke, Raghu Ramakrishnan, Database Management Systems 3rd Edition, McGraw Hill Education,2014.