INDUSTRIAL DATABASE MANAGEMENT SYSTEM
(DAIRY INDUSTRY)

Project Report

Made by

Ramachandran R

Prepared For

DATABASE MANAGEMENT SYSTEMS

TITLE OF THE PROJECT – INDUSTRIAL DATABASE MANAGEMENT SYSTEM (DAIRY INDUSTRY)

ABSTRACT :

The Dairy Management System is a database management system which can be used to maintain the records of seller, sales, staff, etc.. Even though technologies are evolving day to day, still rural people are not benefitted with most of the technologies. Due to this fact, rural people face difficulties in doing jobs related to record maintenance, calculation of profit or loss, etc.. Therefore, my aim is to develop a Graphical User Interface(GUI) application for Dairy Management System which can help to have efficient communication between rural people and Dairy industry.

PROBLEM STATEMENT :

To promote technologies in rural and to develop GUI application for Dairy management System to have efficient communication between rural people (sellers) and Dairy management.

INTRODUCTION :

Before a many decades, Dairy Industries were established. Due to this demand of milk started rising. But only some farmers are benefited from both society and dairy industries. One the other hand, some farmers starved to death due to lack of money (i.e., Poverty). This is because of lack of communication between industry's management and farmers. The major issues revealed were related to lack of fodder and concentrates, scarcity of veterinary and diagnostic services, lack of information and technological awareness. So, this can be overcome with existing technologies. So, I have planned to create a Graphical User Interface (GUI) which helps to establish healthy communication (billing system) between farmers and industry's management .

LITERATURE SURVEY :

- The authors analysed the value chains of milk and milk products in the co-operative and private dairy plants in the Salem district of Tamil Nadu, based on primary data that they have collected form one cooperative and one private dairy plant, five transport routes and six chilling plants. The authors have observed that the procurement cost per litre of milk was higher for the co-operative dairy plant than the private plant due to increase in the transportation, chilling and reception costs. The value chain analysis revealed that the products such as peda, khoa and SMP could earn a

higher value after passing through the value chain in the co-operative plant while ice- cream, Mysore and ghee in the private plant. The marketing margins and marketing efficiency was found higher in toned milk, standardized milk and butter for the private plant and in full cream milk, ghee and SMP for the co-operative plant [1].

- The study was conducted to analyse the marketing efficiency of cooperative and privet dairy plants in Tamil Nadu. To evaluate the marketing efficiency, primary data was collected from 20 milk producers' cooperative societies, 20 milk collection centres, 20 transportation routes (from cooperative and private each). The marketing efficiency of cooperative dairy plant for all dairy products has been observed relatively less than that of private dairy plant, except toned milk [2].
- The study was intended to develop a price determination model for milk. In developing the price determination model different factors like input prices, non-price factors like technology were considered. The authors through the study developed a price model based on cost of production. According to them the model can be used to project the future price of milk. The authors revealed that the elasticity of cost of production with respect to prices of variable inputs was positive and less than one. The prices of dry fodder and concentrate had a major impact in raising the cost of milk production for buffalo milk. Based on the primary data, collected from 160 households in financial year 2002-03 the study explained that the milk price should be adjusted within that range where net income elasticity floats between zero and one [3].
- The study was aimed to find the impact of performance of dairy cooperatives on milk production, income and employment. The primary data was collected from four milk cooperatives from Kolar District of Karnataka. A trend analysis of physical indicators like total membership, total employees and total milk production showed an increasing trend year on year for the period 1995-96 to 2004-05. And for the same period financial performance was analysed based on financial indicators like share capital, sales value, net profit net worth etc. It was observed that all financial indicators showed a positive trend during the period. The employment generation and income earned by the members of dairy cooperatives were higher compared to the non-members because the members of cooperative societies received different services from the societies at low cost or free of cost [4].
- The paper assessed the impact of Karnataka Dairy development project on dairy development in Karnataka. Primary data was collected through survey of 21 villages with cooperatives And 10 villages with non-cooperative dairy units. The author found a positive impact of the project on milk production, as the average production in villages with milk cooperatives was twice the

production in villages without co-operatives. The increase in milk production was achieved through a shift in herd composition. The indigenous cows were replaced by cross bred cows or buffalos. Project led to increase the herd size and investment in cattle. The project had no impact on wage earnings and changes in labour pattern however it had an impact on milk prices in cooperative villages [5].

- ❖ The study dealt with the concept of profitability, measurement of profitability in relation to total investment, sales and shareholders' funds in Dairy Industry in Andhra Pradesh during 2001 to 2011. It also dealt with the evaluation of earning power, analysis of operating efficiency, Analysis of financial efficiency and measurement of financial health of Dairy Industry in Andhra Pradesh, using Z score analysis. The data was collected from 5 dairy enterprises of Andhra Pradesh for a period of 10 years (2000-01 to 2010-11). Four out of five dairy units were found financially sound. While one was found in bankruptcy zone [6].
- ❖ It was a case study of "Gokul" cooperative union, western Maharashtra. A SWOT analysis was carried out for the cooperative union. Through a pretested interview scheduled, data was collected from 150 dairy farmers. The study explained about the 46 livestock services delivered by "Gokul" and the feedback of 150 respondents about the services rendered. The author expressed his opinion that the union had successfully strengthen the dairy production and marketing by providing the livestock services. However the author found that the union had to improve upon the quality of the services rendered and had to reduce the cost attached to these services [7].
- ❖ The study was carried out to understand the issues faced by the dairy farmers and the staff members of the cooperative union. With this purpose the data was collected through interview method, from 150 farmers and the staff members. The constraints were divided into 4 categories as Human Resource, Financial, Policy-Related and Administrative Constraints. Lack of veterans, lack of medical facilities, high cost of concentrates, complex insurance procedure were the major constraints expressed by the farmers. Whereas lack of job satisfaction, low payment, poor coordination among various agencies, lack of proper diagnostic and cold storage facilities were the major issues related to cooperative staff [8].

PROPOSED SYSTEM :

○ **PROJECT PLANNING :**

Developing GUI consist of following Functional Modules :

- To facilitate easily maintenance of records using Database concepts.

- Maintain Daily inventory reports and record of Members.
- Quick access of all records.
- Generate Regular basis Salary.
- Reduce manual work.
- Prevent and reduce human error.

○ REQUIRED TOOLS FOR PROJECT IMPLEMENTATION :

Host Language **:** Python with PyQT5 framework

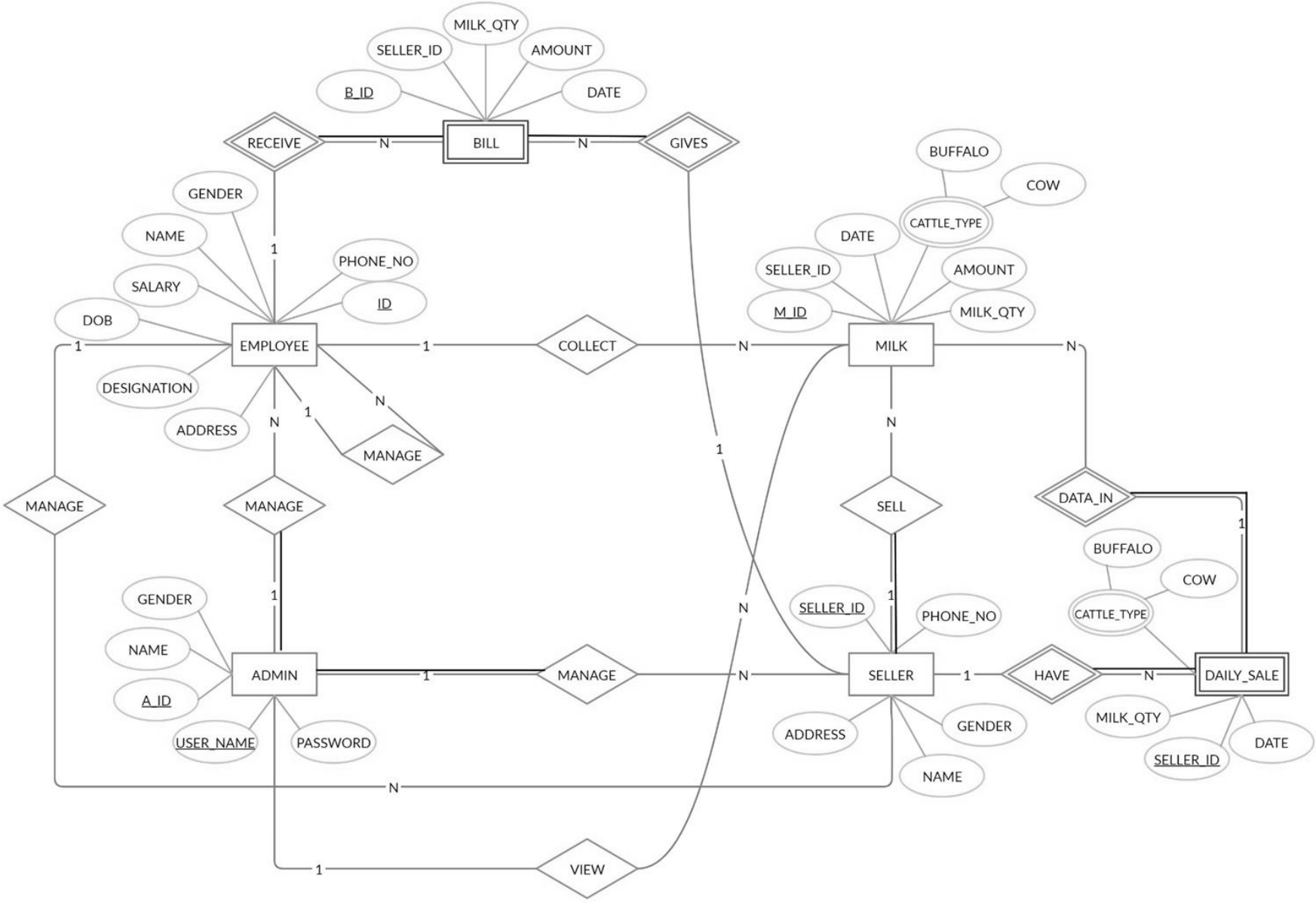
Database Language : MySQL

PyQT5 framework is used to develop GUI in Python environment. MySQL is used for database management.

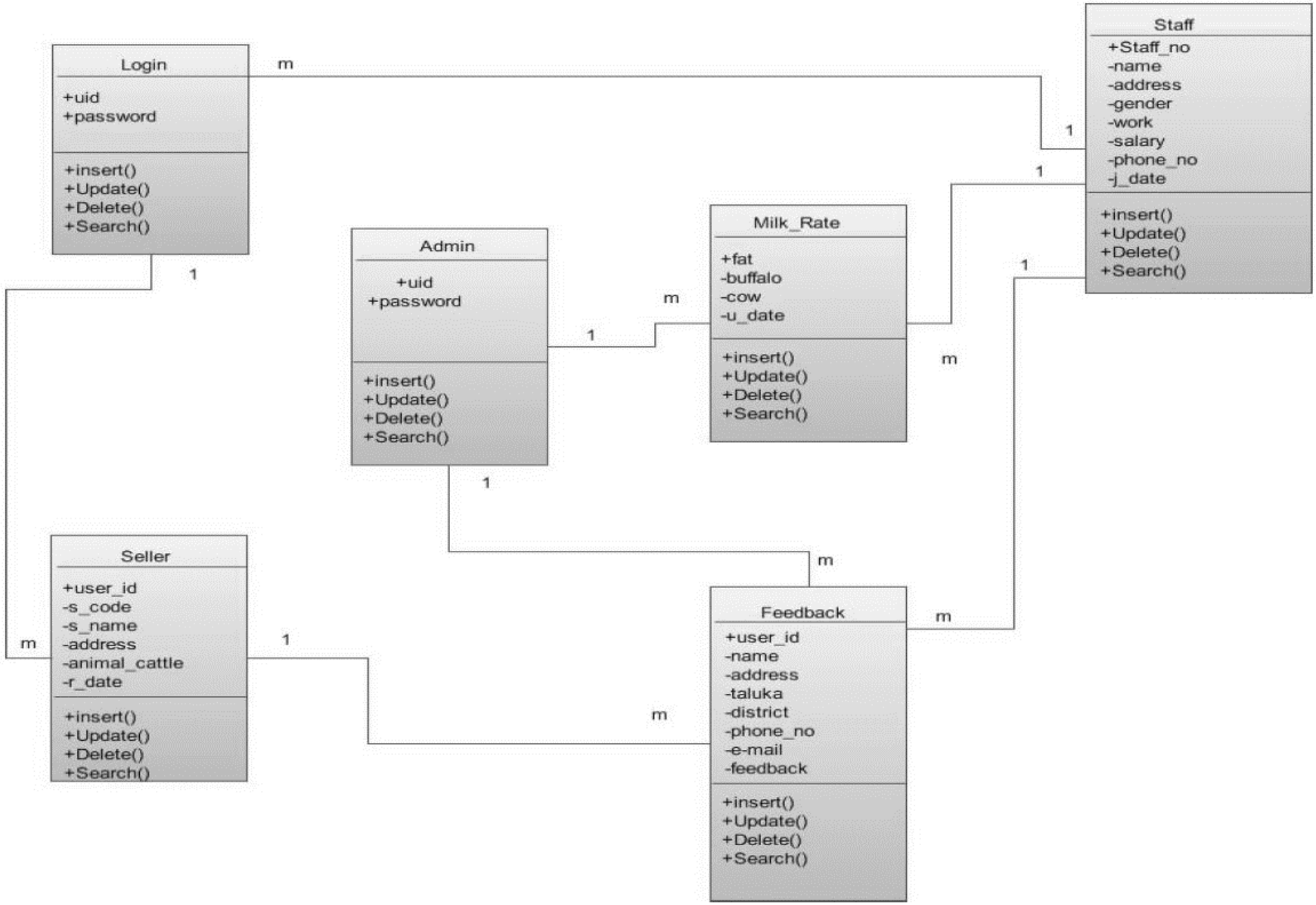
Modules used :

I. Member Account Registration
II. Login/Password module
III. Setting Module
IV. Manage Daily Inventor
V. Generate Salary Regularly
VI. Search/Selection Modules

- **ER – DIAGRAM :**


MILK_QTY
SELLER_ID
AMOUNT
B_ID
DATE
RECEIVE
N
BILL
N
GIVES
GENDER
NAME
SALARY
DOB
PHONE_NO
ID
1
EMPLOYEE
1
COLLECT
N
MILK
DESIGNATION
ADDRESS
N
1
N
MANAGE
MANAGE
MANAGE
1
N
SELL
1
BUFFALO
COW
CATTLE_TYPE
DATE
SELLER_ID
AMOUNT
M_ID
MILK_QTY
N
DATA_IN
1
BUFFALO
COW
CATTLE_TYPE
GENDER
NAME
A_ID
USER_NAME
PASSWORD
ADMIN
1
MANAGE
N
SELLER_ID
PHONE_NO
1
SELLER
1
HAVE
N
DAILY_SALE
ADDRESS
GENDER
NAME
MILK_QTY
SELLER_ID
DATE
N
1
VIEW

- **RELATIONAL SCHEMA :**



- **INNOVATIVE IDEA :**

To keep track of purchase from seller and also automatic syncing of bills in Admin, Employee and Seller accounts. My project also automatically update changes in the respective tables and in the respective row and column. User can also Edit their profile and can also change their profile photo.

Admin can edit Salary, Job Title and Department of employees who belong to respective admin's department. Admin can also see the details of Employees who belong to different departments. If Admin is in Purchase department, Admin can Add New Employee, Delete Existing Employee who doesn't perform well in their job. He/ She can see the purchase report along with seller's ID and under which Employee that seller works. Also, Admin can keep track of job given to Employee by searching in the search bar.

Employee can edit Salary of sellers who belong to respective employee's department. If Employee is in Purchase department, Employee can Add New Seller, Delete Existing Seller who doesn't perform well in their job. He/ She can see the purchase report of sellers working under them. Also, Employee can keep track of job given to Seller by searching in the search bar.

Seller can edit their Sales report, Delete the particular sale and Add the new sale to their sales report.

IMPLEMENTATION :

- **FRONTEND PART (Python)** :

Sample Code :

LoginPage.py

```
# -*- coding: utf-8 -*-

from PyQt5 import QtCore, QtGui, QtWidgets
import MySQLdb
import re

class Ui_LoginPage(object):

    def login(self):
        from AdminPage1 import Ui_AdminPage1
        from EmployeePage1 import Ui_EmployeePage1
        from SellerPage1 import Ui_SellerPage1
        db = MySQLdb.connect(host="localhost",user="root",passwd="")
        c=db.cursor()
        c.execute("use project")
        username=self.lineEdit.text()
        password=self.lineEdit_2.text()
        c.execute("select count(user_id) from login where user_id = %s",[username])
```

```
        check=list(c)[0][0]
        if check>0:
            c.execute("select * from login where user_id = %s and password = %s",[us
ername,password])
            result=list(map(list,c))
            if len(result)>0 and len(re.findall("admin$",result[0][0]))==1 and passw
ord==result[0][1]:
                c.close()
                db.close()
                self.AdP1 = QtWidgets.QMainWindow()
                self.ui = Ui_AdminPage1(self.lineEdit.text(),self.lineEdit_2.text())
                self.ui.setupUi(self.AdP1)
                self.LoginPage.hide()
                self.AdP1.show()
            elif len(result)>0 and len(re.findall("employee$",result[0][0]))==1 and
password==result[0][1]:
                c.close()
                db.close()
                self.EmP1 = QtWidgets.QMainWindow()
                self.ui = Ui_EmployeePage1(self.lineEdit.text(),self.lineEdit_2.text
())
                self.ui.setupUi(self.EmP1)
                self.LoginPage.hide()
                self.EmP1.show()
            elif len(result)>0 and len(re.findall("seller$",result[0][0]))==1 and pa
ssword==result[0][1]:
                c.close()
                db.close()
                self.SeP1 = QtWidgets.QMainWindow()
                self.ui = Ui_SellerPage1(self.lineEdit.text(),self.lineEdit_2.text()
)
                self.ui.setupUi(self.SeP1)
                self.LoginPage.hide()
                self.SeP1.show()
            else:
                QtWidgets.QMessageBox.warning(self.lineEdit, 'Error','Check your Use
r ID and Password 😊')
        elif username=="" or password=="":
            QtWidgets.QMessageBox.information(self.lineEdit, 'Error','User ID or Pas
sword Cannot be Empty')
        else:
            QtWidgets.QMessageBox.information(self.lineEdit, 'Error','User ID doesno
t Exist!!!')
    def openAdminSignUp(self):
        from AdminSignUp import Ui_AdminSignUp
        self.ASU = QtWidgets.QMainWindow()
        self.ui = Ui_AdminSignUp()
        self.ui.setupUi(self.ASU)
        self.LoginPage.hide()
        self.ASU.show()
    def openEmployeeSignUp(self):
```

```
        from EmployeeSignUp import Ui_EmployeeSignUp
        self.ESU = QtWidgets.QMainWindow()
        self.ui = Ui_EmployeeSignUp()
        self.ui.setupUi(self.ESU)
        self.LoginPage.hide()
        self.ESU.show()
    def openSellerSignUp(self):
        from SellerSignUp import Ui_SellerSignUp
        self.SSU = QtWidgets.QMainWindow()
        self.ui = Ui_SellerSignUp()
        self.ui.setupUi(self.SSU)
        self.LoginPage.hide()
        self.SSU.show()
    def setupUi(self, LoginPage):
        self.LoginPage=LoginPage
        LoginPage.setObjectName("LoginPage")
        #LoginPage.resize(1046, 707)                    # Make a Note of it
        LoginPage.setFixedSize(1046, 707)               # Make a Note of it
        icon = QtGui.QIcon()
        icon.addPixmap(QtGui.QPixmap(":/Logo/Images/Logo.png"), QtGui.QIcon.Normal,
QtGui.QIcon.Off)
        LoginPage.setWindowIcon(icon)
        LoginPage.setWindowFlags(LoginPage.windowFlags() & ~QtCore.Qt.WindowMinMaxBu
ttonsHint) # Make a Note of it
        LoginPage.setAutoFillBackground(False)
        LoginPage.setStyleSheet("background-color: rgb(0, 170, 255);")
        LoginPage.setTabShape(QtWidgets.QTabWidget.Rounded)
        self.centralwidget = QtWidgets.QWidget(LoginPage)
        self.centralwidget.setObjectName("centralwidget")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(30, 30, 1001, 61))
        font = QtGui.QFont()
        font.setPointSize(22)
        font.setBold(True)
        font.setUnderline(True)
        font.setWeight(75)
        self.label.setFont(font)
        self.label.setMouseTracking(True)
        self.label.setObjectName("label")
        self.label_2 = QtWidgets.QLabel(self.centralwidget)
        self.label_2.setGeometry(QtCore.QRect(20, 180, 591, 341))
        self.label_2.setPixmap(QtGui.QPixmap(":/Cow/LoginCow.jpg"))
        self.label_2.setScaledContents(True)
        self.label_2.setObjectName("label_2")
        self.frame = QtWidgets.QFrame(self.centralwidget)
        self.frame.setGeometry(QtCore.QRect(620, 180, 401, 431))
        self.frame.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.frame.setFrameShape(QtWidgets.QFrame.StyledPanel)
        self.frame.setFrameShadow(QtWidgets.QFrame.Raised)
        self.frame.setObjectName("frame")
        self.label_5 = QtWidgets.QLabel(self.frame)
```

```
        self.label_5.setGeometry(QtCore.QRect(90, 280, 251, 31))
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setBold(True)
        font.setWeight(75)
        self.label_5.setFont(font)
        self.label_5.setObjectName("label_5")
        self.pushButton_2 = QtWidgets.QPushButton(self.frame)
        self.pushButton_2.setGeometry(QtCore.QRect(30, 350, 91, 41))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_2.setFont(font)
        self.pushButton_2.setStyleSheet("background-color: rgb(181, 181, 181);")
        self.pushButton_2.setObjectName("pushButton_2")
        self.pushButton_2.clicked.connect(self.openAdminSignUp)          # Make a No
te of it
        self.lineEdit = QtWidgets.QLineEdit(self.frame)
        self.LoginPage.lineEdit = self.lineEdit
        self.lineEdit.setGeometry(QtCore.QRect(180, 20, 211, 41))
        self.lineEdit.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.lineEdit.setAlignment(QtCore.Qt.AlignCenter)                # Make a No
te of it
        self.lineEdit.setText("")
        self.lineEdit.setObjectName("lineEdit")
        self.label_3 = QtWidgets.QLabel(self.frame)
        self.label_3.setGeometry(QtCore.QRect(10, 20, 161, 41))
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        self.label_3.setFont(font)
        self.label_3.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.label_3.setObjectName("label_3")
        self.label_4 = QtWidgets.QLabel(self.frame)
        self.label_4.setGeometry(QtCore.QRect(10, 90, 161, 41))
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        self.label_4.setFont(font)
        self.label_4.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.label_4.setObjectName("label_4")
        self.pushButton = QtWidgets.QPushButton(self.frame)
        self.pushButton.setGeometry(QtCore.QRect(150, 150, 91, 41))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton.setFont(font)
```

```
        self.pushButton.setStyleSheet("background-color: rgb(181, 181, 181);")
        self.pushButton.setObjectName("pushButton")
        self.pushButton.clicked.connect(self.login)        # Make a Note of it
        self.lineEdit_2 = QtWidgets.QLineEdit(self.frame)
        self.lineEdit_2.setGeometry(QtCore.QRect(180, 90, 211, 41))
        self.lineEdit_2.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.lineEdit_2.setEchoMode(QtWidgets.QLineEdit.Password)   # Make a Note of
 it
        self.lineEdit_2.setAlignment(QtCore.Qt.AlignCenter)         # Make a Note of
 it
        self.lineEdit_2.setText("")
        self.lineEdit_2.setObjectName("lineEdit_2")
        self.pushButton_3 = QtWidgets.QPushButton(self.frame)
        self.pushButton_3.setGeometry(QtCore.QRect(280, 350, 91, 41))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_3.setFont(font)
        self.pushButton_3.setStyleSheet("background-color: rgb(181, 181, 181);")
        self.pushButton_3.setObjectName("pushButton_3")
        self.pushButton_3.clicked.connect(self.openSellerSignUp)          # Make a N
ote of it
        self.pushButton_4 = QtWidgets.QPushButton(self.frame)
        self.pushButton_4.setGeometry(QtCore.QRect(140, 350, 121, 41))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_4.setFont(font)
        self.pushButton_4.setStyleSheet("background-color: rgb(181, 181, 181);")
        self.pushButton_4.setObjectName("pushButton_4")
        self.pushButton_4.clicked.connect(self.openEmployeeSignUp)        # Make a N
ote of it
        self.line = QtWidgets.QFrame(self.frame)
        self.line.setGeometry(QtCore.QRect(0, 200, 401, 16))
        self.line.setFrameShadow(QtWidgets.QFrame.Plain)
        self.line.setLineWidth(3)
        self.line.setFrameShape(QtWidgets.QFrame.HLine)
        self.line.setObjectName("line")
        self.line_2 = QtWidgets.QFrame(self.frame)
        self.line_2.setGeometry(QtCore.QRect(0, 240, 401, 16))
        self.line_2.setFrameShadow(QtWidgets.QFrame.Plain)
        self.line_2.setLineWidth(3)
        self.line_2.setFrameShape(QtWidgets.QFrame.HLine)
        self.line_2.setObjectName("line_2")
        self.label_6 = QtWidgets.QLabel(self.frame)
        self.label_6.setGeometry(QtCore.QRect(170, 210, 61, 31))
        font = QtGui.QFont()
        font.setPointSize(16)
        font.setBold(True)
```

```
        font.setWeight(75)
        self.label_6.setFont(font)
        self.label_6.setObjectName("label_6")
        self.frame.raise_()
        self.label.raise_()
        self.label_2.raise_()
        LoginPage.setCentralWidget(self.centralwidget)
        self.statusbar = QtWidgets.QStatusBar(LoginPage)
        self.statusbar.setObjectName("statusbar")
        LoginPage.setStatusBar(self.statusbar)

        self.retranslateUi(LoginPage)
        QtCore.QMetaObject.connectSlotsByName(LoginPage)

    def retranslateUi(self, LoginPage):
        _translate = QtCore.QCoreApplication.translate
        LoginPage.setWindowTitle(_translate("LoginPage", "Welcome"))
        self.label.setText(_translate("LoginPage", "<html><head/><body><p><span styl
e=\" color:#55007f;\">DAIRY INDUSTRY DATABASE MANAGEMENT SYSTEM</span></p></body></h
tml>"))
        self.label_2.setText(_translate("LoginPage", "<html><head/><body><p><img src
=\":/Cow/Images/LoginCow.jpg\"/></p></body></html>"))
        self.label_5.setText(_translate("LoginPage", " NEW USER ? SIGNUP AS"))
        self.pushButton_2.setText(_translate("LoginPage", "ADMIN"))
        self.label_3.setText(_translate("LoginPage", "USERNAME"))
        self.label_4.setText(_translate("LoginPage", "PASSWORD"))
        self.pushButton.setText(_translate("LoginPage", "LOGIN"))
        self.pushButton_3.setText(_translate("LoginPage", "SELLER"))
        self.pushButton_4.setText(_translate("LoginPage", "EMPLOYEE"))
        self.label_6.setText(_translate("LoginPage", " OR"))

import LoginCowR
import LogoR

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    LoginPage = QtWidgets.QMainWindow()
    ui = Ui_LoginPage()
    ui.setupUi(LoginPage)
    LoginPage.show()
    sys.exit(app.exec_())
```

For Further Code :

Please refer to My GitHub account. My GitHub ID is :

https://github.com/R-Ramachandran/Industrial-Database-Management-System-Dairy-Industry-

- **BACKEND PART (MySQL)** :

```
create table Login (
User_Id varchar(20) primary key,
password varchar(20) not null);

create table Admin (
A_Id varchar(15) primary key,
A_Name varchar(40) not null,
A_Gender varchar(2),
A_Ph varchar(15) not null,
A_Dob varchar(15),
A_D_Id varchar(15),
A_Address varchar(40),
A_User_Id varchar(15) unique not null references Login,
A_Password varchar(15) not null,
A_Image varchar(150) default "D:\VIT\Database Management
System\Project\PyQT\Images\Admin.PNG");

create table Employee (
E_Name varchar(40) not null,
E_Id varchar(15) primary key,
E_Gender varchar(2),
E_Dob varchar(15),
E_D_Id varchar(15) references Department,
E_A_Id varchar(15) references Admin,
E_Ph varchar(15) not null,
E_Salary varchar(15) not null,
E_Join_Date varchar(15) not null,
E_Address varchar(40),
E_Job_Title varchar(15) not null,
E_User_Id varchar(15) unique not null references Login,
E_Password varchar(15) not null,
E_Image varchar(150) default "D:\VIT\Database Management
System\Project\PyQT\Images\Admin.PNG");
```

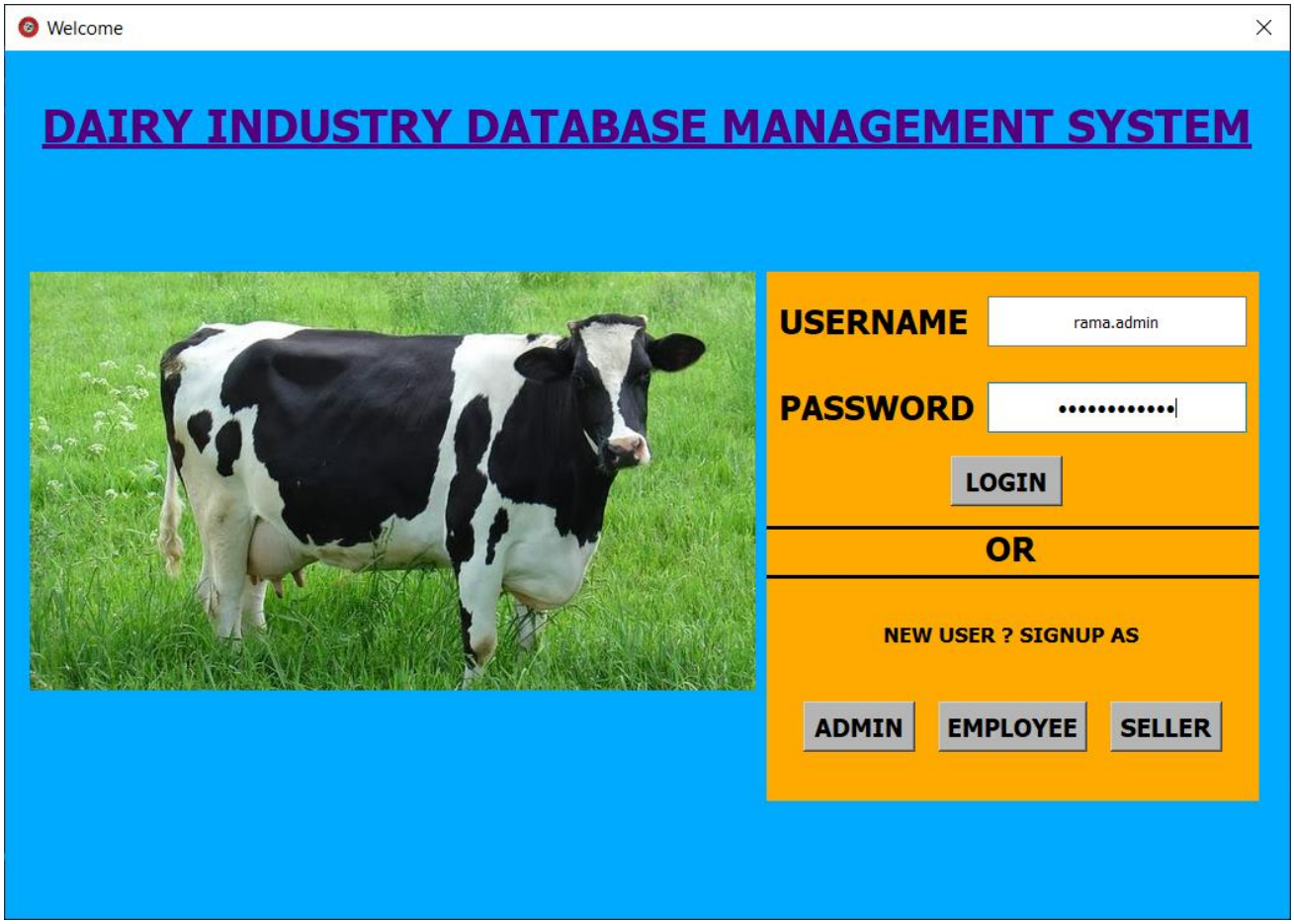
```
create table Department (
D_Id varchar(15) primary key,
D_A_Id varchar(15) unique not null references Admin,
D_Name varchar(15) not null unique);

create table Seller (
S_Name varchar(40) not null,
S_Id varchar(15) primary key,
S_Gender varchar(2),
S_Dob varchar(15),
S_Salary varchar(15) not null,
S_E_Id varchar(15) unique not null references Employee,
S_Ph varchar(15) not null,
S_Address varchar(40),
S_User_Id varchar(15) unique not null references Login,
S_Password varchar(15) not null,
S_Image varchar(150) default "D:\VIT\Database Management System\Project\PyQT
\Images\Admin.PNG");

create table Daily_Milk_Sale (
DS_S_Id varchar(15) unique not null references Seller,
DS_Date varchar(15) not null,
DS_Milk_Qty varchar(15) not null,
DS_Cattle_Type varchar(15),
DS_Expiry_Date varchar(15) not null);

create table Bill (
B_Id varchar(15) primary key,
B_S_Id varchar(15) unique not null references Seller on delete set null,
B_Amount varchar(15) not null,
B_Milk_Qty varchar(15) not null,
```

B_Date varchar(15) not null,

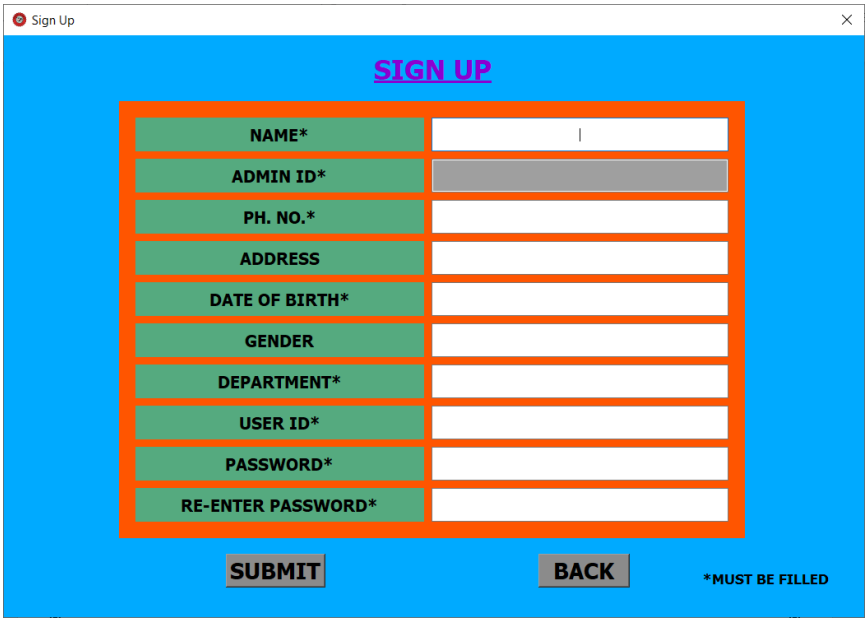
B_E_Id varchar(15) unique not null references Employee on delete set null);

- **SAMPLE SCREENSHOTS OF OUTPUT :**

Login Page :



Signup Page :



CONCLUSION AND FUTURE ENHANCEMENTS :

Thus, in this way, we can make a healthy communication (billing system) between farmers and the industry.

In Future, we can add messaging feature, which helps to have a proper conversation between employees, admins and sellers. Also if there is any change or update in the user account by admin, employee or seller, the changes will be sent to respective user accounts.

We can add forget password field, which helps the user to change the password when he/she forget their password (through OTP or email).

We can also add payment transaction facility, which is helpful to clear the bill in one stretch. Thus, transportation and time waste can be limited.

REFERENCES :

[1] Babua D, VermaNK. Value Chains of Milk and Milk Products in Organised Sector of Tamil Nadu: A Comparative Analysis. Agricultural Economics Research Review 2010; 23: 479-486.

[2] Rangasamy N, Dhaka JP. Marketing Efficiency of Dairy Products for Co-operative and Private Dairy Plants in Tamil Nadu: A Comparative Analysis. Agricultural Economics Research Review 2008; 21: 235-242.

[3] Saravanakumara V, Jainb DK. Evolving Milk Pricing Model for Agribusiness Centres: An Econometric Approach. Agricultural Economics Research Review 2009; 22: 155-160.

[4] Srikanth KN. Performance of dairy cooperatives and their impact on milk production, income and employment in Kolar district: An economic analysis. Department of agricultural economics college of agriculture, Dharwad, University of agricultural sciences, Dharwad. Nov. 2007.

[5] Harold A. Cooperative Dairy Development in Karanataka India: An Assessment. International Food Policy Research Institute, Dec. 1987.

[6] HimaBindu T, Subrahmanyam SEV. A Study of Financial Health of dairy industry in Andhra Pradesh based on Z Score analysis. International Journal of Marketing, Financial Services & Management Research 2012; 1(12).

[7] Rathod PK. SWOT analysis of dairy cooperatives: A case study of Western Maharashtra. International Journal of Research in commerce and Management 2011; (8).

[8] Rathod PK, Nikam TR, Landge S, Hatey A. Perceived Constraints in Livestock Service Delivery by Dairy Cooperatives: A Case Study of Western Maharashtra, India. Indian J. Dairy Sci. 2012; 65(5).

[9] http://meeraacademy.com

[10] https://aavinmilk.com/web/guest/milk

[11] https://www.youtube.com/playlist?list=PL1FgJUcJJ03uO70zDLDF3oaTu6s2QLOPa

[12] https://www.researchgate.net/