A project report

on

BAKERY

management

SYSTEM

SUBMITTED BY

Aksha mishra

CONTENT

1.INTRODUCTION

2.SYSTEM IMPLEMENTATION

3.DATABASE DESIGN

4.MENU DESIGN

5.CODING

6. REFERENCE

INTRODUCTION

This software project is developed to automate the functionalities of a School Automation.The purpose of the software project is to develop the Management Information System

(MIS) to automate the record keeping of school related issue with a view to enhance the decision making of the functionaries.

A MIS mainly consists of a computerized database, a collection of inter-related tables for a particular subject or purpose, capable to produce different reports relevant to the user. An application program is tied with the database for easy access and interface to the database. Using Application program or front-end, we can store, retrieve and manage all information in proper way.

This software, being simple in design and working, does not

require much of training to users, and can be used as a powerful tool for automating a Public Library System.

During coding and design of the software Project, PYTHON IDE,

a powerful front-end tool is used for getting Graphical User

Interface (GUI) based integrated platform and coding simplicity.

As a back-end a powerful, open source RDBMS, My SQL is usedas per requirement of the CBSE curriculum of Informatics Practices Course.

SYSTEM

IMPLEMEMTATION

The Hardware used:

While developing the system, the used hardware are:

PC with Pentium IV processor or sometimes, PC with Celeron (1.7 GHz) processor having 8GB ram, SVGA aand other required devices

The Software used:

Microsoft Windows XP as Operating System.



Anaconda Spyder as Front-End Development environment.

 MySQL as back-end Server with Database for testing.

 MS-Word 2016 for Documentation

Login form

import mysql. connector import tkinter as t def connect (): try:

con= mysql. connector. connect (host=

'localhost’, uSer='root’, dAtAbASe=

'clothmarket')

if con.is\_connected () ==False: print ("database not connected") else:

print("\n") return con except mysql.connector. Error as er:

print(er)

def login():

root=t.Tk()

root.geometry('400x200') f1=t.StringVar() f2=t.StringVar()

def show(): u=f1.get() p=f2.get() if u=="admin"and p=="123": menu() else:

print("please check user name or password")

root.title("welcome") l=t.Label(root,text="login form",font=("arial",30,"bold")).grid(row=0, column=1)

l1=t.Label(root,text="uid").grid(row=1) l2=t.Label(root,text ="password").grid(row=2)

e1=t.Entry(root,textvariable=f1).grid(row=1,c olumn=1)

e2=t.Entry(root,textvariable=f2).grid(row=2,c olumn=1)

t.Button(root,text="exit",command=root.quit).gr id(row=3,column=2)

t.Button(root,text="show",command=show).grid( row=4,column=2) root.mainloop() login()

DATABASE design

create database item;

use item;

create table item

(itemid int,

itemname varchar(20),

price float,itemtype varchar(30));

desc item;

---------+-------------+------+-----+---------+-------+

Field | Type | Null | Key | Default | Extra |

----------+-------------+------+-----+---------+-------+ itemid | int(11) | YES | | NULL | | itemname | varchar(20) | YES | | NULL | price | float | YES | | NULL | | itemtype | varchar(30) | YES | | NULL | | select \* from item;

--------+---------------+-------+----------+

itemid | itemname | price | itemtype |

--------+---------------+-------+----------+

1. | burger | 200 | large |
2. | french fries | 100 | medium |
3. | pizza | 250 | large |
4. | sandwich | 150 | small |
5. | momos | 300 | medium |

420 | cake | 5000 | large |

desc billing;

Create database bill;

Use bill;

Create table \_billing

(price integer,

quantity integer,

total\_price integer,

gst float, cgst float, sgst float, billno int, itemid int, itemtype varchar(20),

itemname varchar(20), finalprice float, customer name varhar (20), tax float,

desc billing;

| Field | Type | Null | Key | Default | Extra |

+--------------+-------------+------+-----+---------+-------+

| price | float | YES | | NULL | |

| qty | int(11) | YES | | NULL | |

| totalprice | float | YES | | NULL | |

| invoicedate | date | YES | | NULL | |

| cgst | float | YES | | NULL | | | sgst | float | YES | | NULL | |

| billno | int(11) | YES | | NULL | |

| itemid | int(11) | YES | | NULL | |

| itemtype | varchar(30) | YES | | NULL | |

| itemname | varchar(20) | YES | | NULL | |

| finalprice | float | YES | | NULL | |

| customername | varchar(20) | YES | | NULL | |

| tax | float | YES | | NULL | |

desc billing;

+--------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------------+-------------+------+-----+---------+-------+

| price | float | YES | | NULL | |

| qty | int(11) | YES | | NULL | |

| totalprice | float | YES | | NULL | |

| invoicedate | date | YES | | NULL | |

| cgst | float | YES | | NULL | | | sgst | float | YES | | NULL | |

| billno | int(11) | YES | | NULL | |

| itemid | int(11) | YES | | NULL | |

| itemtype | varchar(30) | YES | | NULL | |

| itemname | varchar(20) | YES | | NULL | |

| finalprice | float | YES | | NULL | |

| customername | varchar(20) | YES | | NULL | |

| tax | float | YES | | NULL | | select \* from billing;

| price | qty | totalprice | invoicedate | cgst | sgst | billno | itemid | itemtype | itemname | finalprice | customername | tax |

| 500 | 6 | 600 | 2019-03-10 | 7 | 8 | 89 | 103 | large | pizza | NULL | NULL |nuLL |

Menu design

Coding

CODE FOR CREATING bakery

print(----------------------WELCOME TO bakery MANAGEMENT-------------------------)

import mysql.connector as sq import tkinter as t

def menu ():

print("1. add item\n 2.display\n 3.update\n 4. delete\n 5. billing\n") ch=int(input("enter your choice")) if ch==1:

print("add item detail")

insert() if ch==2:

print("display item details") display() if ch==3: print("modify item details") update() if ch==4: print("remove item") delete() if ch==5: print("billing") bill()

if ch=='f':

print ("1. Add Invoice details\n2.Generate bill\n")

f=int (input ("enter your choice")) if f==1: print ("add Item Details in bill") insert5() if f==2: print ("display item details in bill") display5()

enter your choice

1.insert

2.display

3.update

4.delete

5.bill

CODE FOR INSERTION OF DATA IN ITEM table

def insert(): try:

con=sq.connect(host="localhost",user="root",passw ord="tiger",database="item") if con.is\_connected():

print(" database connected") iid=int(input("enter itemid")) iname=input("enter iname") price= int(input("enter price")) itemtype=input("enter itemtype") cur=con.cursor()

cur.execute("insert into item values (%d,'%s',%f,'%s')"%(iid,iname,price,itemtype)) con.commit()

print("data inserted successfully")

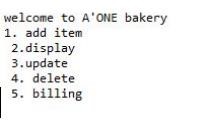
else:

print("database not connected") except sq.Error as er:

print(er)

CODE FOR INSERTING STUDENT DETAILS AND

SEEING



MODIFIED DETAILS

--------+---------------+-------+----------+

itemid | itemname | price | itemtype |

--------+---------------+-------+----------+

1. | burger | 200 | large |
2. | french fries | 100 | medium |
3. | pizza | 250 | large |
4. | sandwich | 150 | small |
5. | momos | 300 | medium |

420 | cake | 5000 | large |

106 | sandwich| 50 | medium |

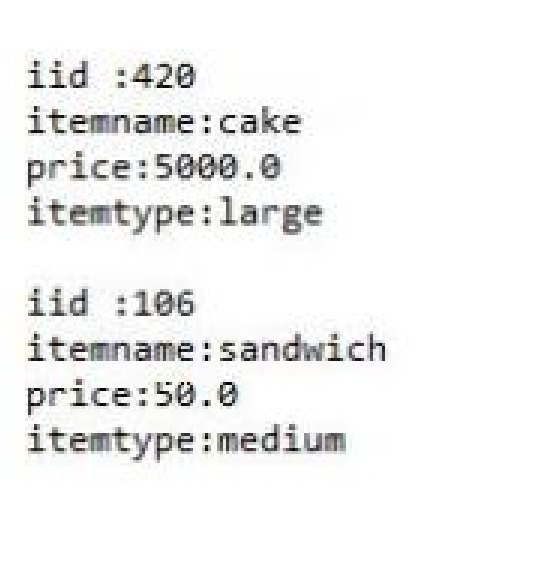
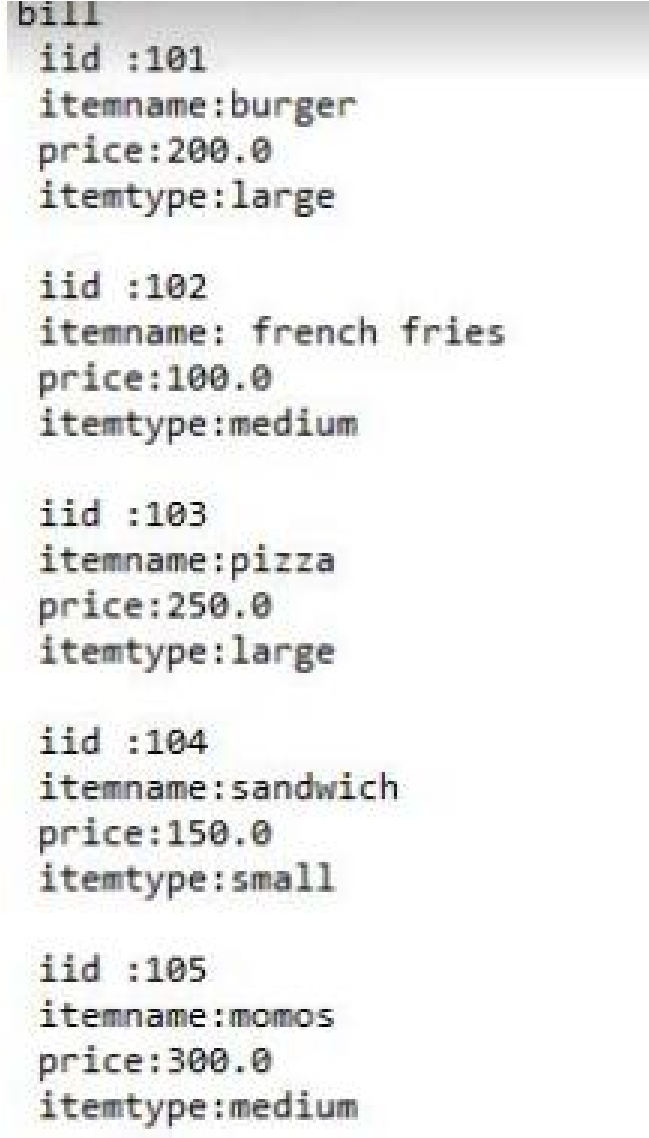
CODE FOR DISPLAY DATA OF item table

def display(): try:

con=sq.connect(host="localhost",user="root",pa ssword="tiger",database="item") if con.is\_connected()==False: print("database not connected") cur=con.cursor() cur.execute("select \* from item")

row= cur.fetchall() for i in row : a=i[0] b=i[1] c=i[2] d=i[3]

print(f' iid :{a}\n itemname:{b}\n price:{c}\n itemtype:{d}\n') con.commit() except sq.Error as er: print(er) display()



CODE FOR UPDATINg data of item table

def update(): try: con=sq.connect(host="localhost",user="root",pa ssword="tiger",database="item") if con.is\_connected()==False:

print("database not connected") cur=con.cursor() iid=int(input("enter itemid")) iname=input("enter iname") price= int(input("enter price")) itemtype=input("enter itemtype") cur.execute("update item set(iid=%d and iname=%s and print=%f)where itemtype=%s"(iid,iname,price,itemtype)) con.commit() print(" database inserted secessfully") except sq.Error as er: print(er) update()

CODE FOR DELETING DATA FROM item table

def delete(): try:

con=sq.connect(host="localhost",user="root",pa ssword="tiger",database="item") if con.is\_connected()==False:

print("database not connected") cur=con.cursor() a=input("enter itemtype which u want to delete") cur.execute("delete from item where itemtype=%s"%(a)) con.commit() print("n\delete succesfully") cur.execute("select \* from item")

except sq.Error as er: print(er)

enter4 remove itemtype connected enter itemtype which u want to delete105 data delete successfully

--------+---------------+-------+----------+

itemid | itemname | price | itemtype |

--------+---------------+-------+----------+

1. | burger | 200 | large |
2. | french fries | 100 | medium |
3. | pizza | 250 | large |
4. | sandwich | 150 | small |

106| cake | 5000| large|

CODE FOR GENRATING BILL

def insert(): try:

con=sq.connect(host="localhost",user="root",pa ssword="tiger",database="bill") if con.is\_connected():

print(" database connected") price= int(input("enter price")) qty=int(input("enter qty")) totalprice=(price\*qty) print("totalprice") invoicedate=input("enter invoicedate") cgst=int(input("enter cgst")) sgst=int(input("enter sgst")) billno=int(input("enter billno"))

iid=int(input("enter itemid"))

itemtype=input("enter itemtype") gst=tax\*tax/100 cgst= gst/2 sgst=gst/2 final price=tax+gst itemname=input("enter itemname") cur=con.cursor() cur.execute("insert into billing values

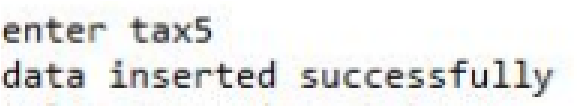
( '%f','%d','%f','%s','%f','%f','%d','%d','%s','%s')"%(price,qty, totalpric,invoicedate,cgst,sgst,billno,iid,itemty pe,itemname))

con.commit() print("data inserted successfully")

else:

print("database not connected") except sq.Error as er: print(er)





| price | qty | totalprice | invoicedate | cgst | sgst | billno | itemid | itemtype | itemname | finalprice | customername | tax |

| 500 | 6 | 600 | 2019-03-10 | 7 | 8 | 89 | 103 | large | pizza | NULL | NULL |NULL |

| 104 | 7 | 728 | 2020-02-21 | 18.2 | 18.2 | 90 | 107 | large | pastry | 764.4 | mahesh| 5 |

CODE FOR DISPLAY BILL def display(): try:

con=sq.connect(host="localhost",user="root",pa ssword="tiger",database="bill") h=int(input("enter itenid which u want to display")) if con.is\_connected()==False:

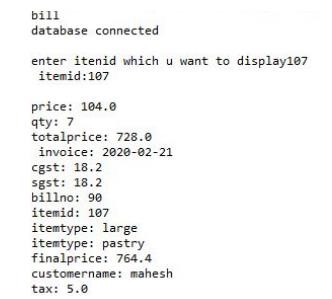
print("database not connected") else:

cur=con.cursor()

cur.execute("select \* from billing where itemid=%d"%(h)) for i in cur.fetchall() : a=i[0] b=i[1] c=i[2] d=i[3] e=i[4] f=i[5] g=i[6] h=i[7] i1=i[8] j=i[9] k=[10] l=[11] m=[12]

print(f' price :{a}\n qty:{b}\n totalprice:{c}\n invoice :{d}\n cgst:{e}\n sgst:{f}\n billno: {g}\n itemid :{h}\n itemtype:{i1}\n itemtype:{j}\n' finalprice:{k}\N’ customername:{l}\N’ tAx:{M}\N’) except sq.Error as er:

print(er)



REFERENCES

In order to work on this project titled bakery MANAGEMENT,

the following books and literature are referred by me during the various phrases of development of the project.

1. MySQL BLACK BOOK

-by STEVEN HOLZNEN

1. Understanding SQL

-Gruber

1. http://www.mysql.org/
2. Informatics practices for class XII

-by Sumita Arora

1. Together with Informatics Practices
2. Various Website of Discussion Forum and Software

Development activities

Other than the above-mentioned books, the suggestion and supervision of my teacher and my class experience also helped me to develop the software project.

Thankyou

Made by Akshamishra