**System Implementations**

**Recommended System Requirements**

Processors: Intel® Core™ i3 processor 4300M at 2.60 GHz.

Disk space: 4 to 8 GB.

Operating systems: Windows® 10, MACOS, and UBUNTU.

Python Versions: 3.X.X or Higher.

**Minimum System Requirements**

Processors: Intel Atom® processor or Intel® Core™ i3 processor.

Disk space: 1 GB.

Operating systems: Windows 7 or later, MACOS, and UBUNTU.

Python Versions: 2.7.X, 3.9.X.

**ACKNOWLEDGEMENT**TTT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

We would like to express my deep and sincere gratitude to my subject teacher, Mr. Amit Udiwal, for giving me the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as and honour to work and study under his guidance. We are very much thankful to our Sr. Jasmin for giving valuable time and moral support to develop this software. We would like to take opportunity to extend my sincere thanks and gratitude to our parents for being a source of inspiration and providing time and freedom to develop this software project. We also feel indebted to my friends for the valuable suggestions during the project work.

Rajveer Singh Panwar

[Roll No.

Class XII

**CERTIFICATE**

This is to certify that the project on ‘Customer Management System’ is a work done by Rajveer Singh Panwar fulfilment of CBSE’S AISSCE EXAMINATION 2022-23 and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form any other examination and does not form any other course undergone by the candidate.

Name:

Rajveer Singh Panwar [Roll No.

………………….

Signature of Teacher / Guide

Name: Mr. Amit Udiwal

Designation:

………………. ….………………

**REFERENCE**

The order to work on this project on ‘Customer Management System’ the following books & literature are referred by me during the various phrases of department of the project.

• http://www.python.org/.

• http://www.itsourcecode.org/.

• http://www.wikipedia.org/.

• Informatics Practices for Class XII

- By Sumita Arora

• Together with informatics practices.

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

**Introduction**

The **Customer Management System Code In Python** is a collection of all the systems, processes, and applications required to manage client interactions. Customer relationship management is another term for this.

This **Customer Management System Program In Python** is important because it helps assist organizations in developing relationships with their customers, which leads to loyalty and client retention. Customer Relationship Management (CRM) is a management technique that increases a company’s profitability because customer loyalty and revenue are both factors that influence revenue.

**Objective and**

**Scope of The Project**

The **Customer Management System Project In Python** is a graphical user interface system written and designed in the Python programming language. The project is open source, and it was made for novices who wish to learn Python. This **Customer Management System Project In Python** is a simple project that has been made to help business owners manage their customers for improved growth and revenue. The system has a simple design which allows you to insert all the necessary data like name, last name, gender, age, address, and contact information. This is very helpful and valuable to business. I hope this article can help you a lot.

**Customer Management System**

from tkinter import \*

import sqlite3

import tkinter.ttk as ttk

import tkinter.messagebox as tkMessageBox

root = Tk()

root.title("Customer Information System")

width = 700

height = 400

screen\_width = root.winfo\_screenwidth()

screen\_height = root.winfo\_screenheight()

x = (screen\_width/2) - (width/2)

y = (screen\_height/2) - (height/2)

root.geometry("%dx%d+%d+%d" % (width, height, x, y))

root.resizable(0, 0)

root.config(bg="black")

#============================VARIABLES===================================

FIRSTNAME = StringVar()

LASTNAME = StringVar()

GENDER = StringVar()

AGE = StringVar()

ADDRESS = StringVar()

CONTACT = StringVar()

#============================METHODS=====================================

def Database():

conn = sqlite3.connect("pythontut.db")

cursor = conn.cursor()

cursor.execute("CREATE TABLE IF NOT EXISTS `member` (mem\_id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT, firstname TEXT, lastname TEXT, gender TEXT, age TEXT, address TEXT, contact TEXT)")

cursor.execute("SELECT \* FROM `member` ORDER BY `lastname` ASC")

fetch = cursor.fetchall()

for data in fetch:

tree.insert('', 'end', values=(data))

cursor.close()

conn.close()

def SubmitData():

if FIRSTNAME.get() == "" or LASTNAME.get() == "" or GENDER.get() == "" or AGE.get() == "" or ADDRESS.get() == "" or CONTACT.get() == "":

result = tkMessageBox.showwarning('', 'Please Complete The Required Field', icon="warning")

else:

tree.delete(\*tree.get\_children())

conn = sqlite3.connect("pythontut.db")

cursor = conn.cursor()

cursor.execute("INSERT INTO `member` (firstname, lastname, gender, age, address, contact) VALUES(?, ?, ?, ?, ?, ?)", (str(FIRSTNAME.get()), str(LASTNAME.get()), str(GENDER.get()), int(AGE.get()), str(ADDRESS.get()), str(CONTACT.get())))

conn.commit()

cursor.execute("SELECT \* FROM `member` ORDER BY `lastname` ASC")

fetch = cursor.fetchall()

for data in fetch:

tree.insert('', 'end', values=(data))

cursor.close()

conn.close()

FIRSTNAME.set("")

LASTNAME.set("")

GENDER.set("")

AGE.set("")

ADDRESS.set("")

CONTACT.set("")

def UpdateData():

if GENDER.get() == "":

result = tkMessageBox.showwarning('', 'Please Complete The Required Field', icon="warning")

else:

tree.delete(\*tree.get\_children())

conn = sqlite3.connect("pythontut.db")

cursor = conn.cursor()

cursor.execute("UPDATE `member` SET `firstname` = ?, `lastname` = ?, `gender` =?, `age` = ?, `address` = ?, `contact` = ? WHERE `mem\_id` = ?", (str(FIRSTNAME.get()), str(LASTNAME.get()), str(GENDER.get()), str(AGE.get()), str(ADDRESS.get()), str(CONTACT.get()), int(mem\_id)))

conn.commit()

cursor.execute("SELECT \* FROM `member` ORDER BY `lastname` ASC")

fetch = cursor.fetchall()

for data in fetch:

tree.insert('', 'end', values=(data))

cursor.close()

conn.close()

FIRSTNAME.set("")

LASTNAME.set("")

GENDER.set("")

AGE.set("")

ADDRESS.set("")

CONTACT.set("")

def OnSelected(event):

global mem\_id, UpdateWindow

curItem = tree.focus()

contents =(tree.item(curItem))

selecteditem = contents['values']

mem\_id = selecteditem[0]

FIRSTNAME.set("")

LASTNAME.set("")

GENDER.set("")

AGE.set("")

ADDRESS.set("")

CONTACT.set("")

FIRSTNAME.set(selecteditem[1])

LASTNAME.set(selecteditem[2])

AGE.set(selecteditem[4])

ADDRESS.set(selecteditem[5])

CONTACT.set(selecteditem[6])

UpdateWindow = Toplevel()

UpdateWindow.title("Customer Information System")

width = 400

height = 300

screen\_width = root.winfo\_screenwidth()

screen\_height = root.winfo\_screenheight()

x = ((screen\_width/2) + 450) - (width/2)

y = ((screen\_height/2) + 20) - (height/2)

UpdateWindow.resizable(0, 0)

UpdateWindow.geometry("%dx%d+%d+%d" % (width, height, x, y))

if 'NewWindow' in globals():

NewWindow.destroy()

#===================FRAMES==============================

FormTitle = Frame(UpdateWindow)

FormTitle.pack(side=TOP)

ContactForm = Frame(UpdateWindow)

ContactForm.pack(side=TOP, pady=10)

RadioGroup = Frame(ContactForm)

Male = Radiobutton(RadioGroup, text="Male", variable=GENDER, value="Male", font=('arial', 14)).pack(side=LEFT)

Female = Radiobutton(RadioGroup, text="Female", variable=GENDER, value="Female", font=('arial', 14)).pack(side=LEFT)

#===================LABELS==============================

lbl\_title = Label(FormTitle, text="Updating Contacts", font=('arial', 16), bg="orange", width = 300)

lbl\_title.pack(fill=X)

lbl\_firstname = Label(ContactForm, text="Firstname", font=('arial', 14), bd=5)

lbl\_firstname.grid(row=0, sticky=W)

lbl\_lastname = Label(ContactForm, text="Lastname", font=('arial', 14), bd=5)

lbl\_lastname.grid(row=1, sticky=W)

lbl\_gender = Label(ContactForm, text="Gender", font=('arial', 14), bd=5)

lbl\_gender.grid(row=2, sticky=W)

lbl\_age = Label(ContactForm, text="Age", font=('arial', 14), bd=5)

lbl\_age.grid(row=3, sticky=W)

lbl\_address = Label(ContactForm, text="Address", font=('arial', 14), bd=5)

lbl\_address.grid(row=4, sticky=W)

lbl\_contact = Label(ContactForm, text="Contact", font=('arial', 14), bd=5)

lbl\_contact.grid(row=5, sticky=W)

#===================ENTRY===============================

firstname = Entry(ContactForm, textvariable=FIRSTNAME, font=('arial', 14))

firstname.grid(row=0, column=1)

lastname = Entry(ContactForm, textvariable=LASTNAME, font=('arial', 14))

lastname.grid(row=1, column=1)

RadioGroup.grid(row=2, column=1)

age = Entry(ContactForm, textvariable=AGE, font=('arial', 14))

age.grid(row=3, column=1)

address = Entry(ContactForm, textvariable=ADDRESS, font=('arial', 14))

address.grid(row=4, column=1)

contact = Entry(ContactForm, textvariable=CONTACT, font=('arial', 14))

contact.grid(row=5, column=1)

#==================BUTTONS==============================

btn\_updatecon = Button(ContactForm, text="Update", width=50, command=UpdateData)

btn\_updatecon.grid(row=6, columnspan=2, pady=10)

#fn1353p

def DeleteData():

if not tree.selection():

result = tkMessageBox.showwarning('', 'Please Select Something First!', icon="warning")

else:

result = tkMessageBox.askquestion('', 'Are you sure you want to delete this record?', icon="warning")

if result == 'yes':

curItem = tree.focus()

contents =(tree.item(curItem))

selecteditem = contents['values']

tree.delete(curItem)

conn = sqlite3.connect("pythontut.db")

cursor = conn.cursor()

cursor.execute("DELETE FROM `member` WHERE `mem\_id` = %d" % selecteditem[0])

conn.commit()

cursor.close()

conn.close()

def AddNewWindow():

global NewWindow

FIRSTNAME.set("")

LASTNAME.set("")

GENDER.set("")

AGE.set("")

ADDRESS.set("")

CONTACT.set("")

NewWindow = Toplevel()

NewWindow.title("Customer Information System")

width = 400

height = 300

screen\_width = root.winfo\_screenwidth()

screen\_height = root.winfo\_screenheight()

x = ((screen\_width/2) - 455) - (width/2)

y = ((screen\_height/2) + 20) - (height/2)

NewWindow.resizable(0, 0)

NewWindow.geometry("%dx%d+%d+%d" % (width, height, x, y))

if 'UpdateWindow' in globals():

UpdateWindow.destroy()

#===================FRAMES==============================

FormTitle = Frame(NewWindow)

FormTitle.pack(side=TOP)

ContactForm = Frame(NewWindow)

ContactForm.pack(side=TOP, pady=10)

RadioGroup = Frame(ContactForm)

Male = Radiobutton(RadioGroup, text="Male", variable=GENDER, value="Male", font=('arial', 14)).pack(side=LEFT)

Female = Radiobutton(RadioGroup, text="Female", variable=GENDER, value="Female", font=('arial', 14)).pack(side=LEFT)

#===================LABELS==============================

lbl\_title = Label(FormTitle, text="Adding New Customer", font=('arial', 16), bg="#66ff66", width = 300)

lbl\_title.pack(fill=X)

lbl\_firstname = Label(ContactForm, text="Firstname", font=('arial', 14), bd=5)

lbl\_firstname.grid(row=0, sticky=W)

lbl\_lastname = Label(ContactForm, text="Lastname", font=('arial', 14), bd=5)

lbl\_lastname.grid(row=1, sticky=W)

lbl\_gender = Label(ContactForm, text="Gender", font=('arial', 14), bd=5)

lbl\_gender.grid(row=2, sticky=W)

lbl\_age = Label(ContactForm, text="Age", font=('arial', 14), bd=5)

lbl\_age.grid(row=3, sticky=W)

lbl\_address = Label(ContactForm, text="Address", font=('arial', 14), bd=5)

lbl\_address.grid(row=4, sticky=W)

lbl\_contact = Label(ContactForm, text="Contact", font=('arial', 14), bd=5)

lbl\_contact.grid(row=5, sticky=W)

#===================ENTRY===============================

firstname = Entry(ContactForm, textvariable=FIRSTNAME, font=('arial', 14))

firstname.grid(row=0, column=1)

lastname = Entry(ContactForm, textvariable=LASTNAME, font=('arial', 14))

lastname.grid(row=1, column=1)

RadioGroup.grid(row=2, column=1)

age = Entry(ContactForm, textvariable=AGE, font=('arial', 14))

age.grid(row=3, column=1)

address = Entry(ContactForm, textvariable=ADDRESS, font=('arial', 14))

address.grid(row=4, column=1)

contact = Entry(ContactForm, textvariable=CONTACT, font=('arial', 14))

contact.grid(row=5, column=1)

#==================BUTTONS==============================

btn\_addcon = Button(ContactForm, text="Save", width=50, command=SubmitData)

btn\_addcon.grid(row=6, columnspan=2, pady=10)

#============================FRAMES======================================

Top = Frame(root, width=500, bd=1, relief=SOLID)

Top.pack(side=TOP)

Mid = Frame(root, width=500, bg="black")

Mid.pack(side=TOP)

MidLeft = Frame(Mid, width=100)

MidLeft.pack(side=LEFT, pady=10)

MidLeftPadding = Frame(Mid, width=370, bg="black")

MidLeftPadding.pack(side=LEFT)

MidRight = Frame(Mid, width=100)

MidRight.pack(side=RIGHT, pady=10)

TableMargin = Frame(root, width=500)

TableMargin.pack(side=TOP)

#============================LABELS======================================

lbl\_title = Label(Top, text="Customer Information System", font=('arial', 16), width=500,bg="blue")

lbl\_title.pack(fill=X)

#============================ENTRY=======================================

#============================BUTTONS=====================================

btn\_add = Button(MidLeft, text="+ ADD NEW", bg="#66ff66", command=AddNewWindow)

btn\_add.pack()

btn\_delete = Button(MidRight, text="DELETE", bg="red", command=DeleteData)

btn\_delete.pack(side=RIGHT)

#============================TABLES======================================

scrollbarx = Scrollbar(TableMargin, orient=HORIZONTAL)

scrollbary = Scrollbar(TableMargin, orient=VERTICAL)

tree = ttk.Treeview(TableMargin, columns=("MemberID", "Firstname", "Lastname", "Gender", "Age", "Address", "Contact"), height=400, selectmode="extended", yscrollcommand=scrollbary.set, xscrollcommand=scrollbarx.set)

scrollbary.config(command=tree.yview)

scrollbary.pack(side=RIGHT, fill=Y)

scrollbarx.config(command=tree.xview)

scrollbarx.pack(side=BOTTOM, fill=X)

tree.heading('MemberID', text="MemberID", anchor=W)

tree.heading('Firstname', text="Firstname", anchor=W)

tree.heading('Lastname', text="Lastname", anchor=W)

tree.heading('Gender', text="Gender", anchor=W)

tree.heading('Age', text="Age", anchor=W)

tree.heading('Address', text="Address", anchor=W)

tree.heading('Contact', text="Contact", anchor=W)

tree.column('#0', stretch=NO, minwidth=0, width=0)

tree.column('#1', stretch=NO, minwidth=0, width=0)

tree.column('#2', stretch=NO, minwidth=0, width=80)

tree.column('#3', stretch=NO, minwidth=0, width=120)

tree.column('#4', stretch=NO, minwidth=0, width=90)

tree.column('#5', stretch=NO, minwidth=0, width=80)

tree.column('#6', stretch=NO, minwidth=0, width=120)

tree.column('#7', stretch=NO, minwidth=0, width=120)

tree.pack()

tree.bind('<Double-Button-1>', OnSelected)

#============================INITIALIZATION==============================

if \_\_name\_\_ == '\_\_main\_\_':

Database()

root.mainloop()