Python Project work

On

MOVIE RENTAL SYSTEM

**ABSTRACT**

**INTRODUCTION:**

The purpose is to develop an efficient way to keep records of all the details in movie rental shop using python.

**PROBLEM:**

Keeping a manual record of all details and transactions for hundreds of customers and thousands of CDs, DVDs and Blu Rays and their borrow, return and fine details is extremely hectic. Therefore we need an automatic system that can be easily handled by anyone and can still keep track of all records.

**DETAILS & PROCEDURE:**

Movie Rental Shop Management System project is written in Python. The project file contains python scripts. This is a simple console based system which is very easy to use and understand. Talking about the system, it contains basic functions which include a list of all rentals, rent DVD, filter DVD and return the DVD. In this mini project, there is no such login system. This means he/she can use all those available features easily without any restriction. Each menu item holds a command and the user has to enter the command to use it.

After entering details for each item, he/she can view all the rentals. While renting a DVD, the user has to enter DVD name and client’s name as a module code on the details section. And the user can also return DVDs by entering customer name and phone number as a module code. This simple console based Student Management system provides the simplest rental management of DVD store. There is no database connection but different external text files are used in this mini project to save user’s data permanently and retrieve it.

**MODULE**

import tkinter as tk

from tkinter import ttk

from tkinter import messagebox

import sqlite3

root = tk.Tk()

root.title("Movie Rental Management")

connection = sqlite3.connect('management1.db')

TABLE\_NAME = "management\_table"

CUST\_ID = "cust\_id"

CUST\_NAME = "cust\_name"

CUST\_ADDR = "cust\_addr"

CUST\_PHONE = "cust\_phone"

RENTAL\_DATE = "rental\_date"

connection.execute(" CREATE TABLE IF NOT EXISTS " + TABLE\_NAME + " ( " + CUST\_ID +

" INTEGER PRIMARY KEY AUTOINCREMENT, " +

CUST\_NAME + " TEXT, " + CUST\_ADDR + " TEXT, " +

CUST\_PHONE + " INTEGER, " + RENTAL\_DATE + " INTEGER);")

header = tk.Label(root, text="Movie Rental Management System", fg="red", width=35)

header.config(font=("Times New Roman", 40))

header.grid(row=0, columnspan=2, padx=(10,10), pady=(30, 0))

nameLabel = tk.Label(root, text="Enter name", width=40, anchor='w',

font=("helvetica", 12)).grid(row=1, column=0)

movieLabel = tk.Label(root, text="Enter movie name", width=40, anchor='w',

font=("helvetica", 12)).grid(row=2, column=0)

phoneLabel = tk.Label(root, text="Enter phone number", width=40, anchor='w',

font=("helvetica", 12)).grid(row=3, column=0)

dateLabel = tk.Label(root, text="Enter rental date", width=40, anchor='w',

font=("helvetica", 12)).grid(row=4, column=0)

nameEntry = tk.Entry(root, width = 30)

movieEntry = tk.Entry(root, width = 30)

phoneEntry = tk.Entry(root, width = 30)

dateEntry = tk.Entry(root, width = 30)

nameEntry.grid(row=1, column=1, padx=(0,10), pady=(30, 20))

movieEntry.grid(row=2, column=1, padx=(0,10), pady = 20)

phoneEntry.grid(row=3, column=1, padx=(0,10), pady = 20)

dateEntry.grid(row=4, column=1, padx=(0,10), pady = 20)

def takeNameInput():

global nameEntry, movieEntry, phoneEntry, dateEntry

global list

global TABLE\_NAME, CUST\_NAME, CUST\_ADDR, CUST\_PHONE, RENTAL\_DATE

username = nameEntry.get()

nameEntry.delete(0, tk.END)

movieName = movieEntry.get()

movieEntry.delete(0, tk.END)

phone = int(phoneEntry.get())

phoneEntry.delete(0, tk.END)

date = int(dateEntry.get())

dateEntry.delete(0, tk.END)

connection.execute("INSERT INTO " + TABLE\_NAME + " ( " + CUST\_NAME + ", " +

CUST\_ADDR + ", " + CUST\_PHONE + ", " +

RENTAL\_DATE + " ) VALUES ( '"

+ username + "', '" + movieName + "', '" +

str(phone) + "', " + str(date) + " ); ")

connection.commit()

messagebox.showinfo("Success", "Data Saved Successfully.")

def RootWindow():

#root.destroy()

secondWindow = tk.Tk()

secondWindow.title("Display results")

header = tk.Label(secondWindow, text="Movie Renatal Management System",fg="#06a099", width=40)

header.config(font=("Sylfaen", 30))

header.pack()

tree = ttk.Treeview(secondWindow)

tree["columns"] = ("one", "two", "three", "four")

tree.heading("one", text="Customer Name")

tree.heading("two", text="Movie Name")

tree.heading("three", text="Phone")

tree.heading("four", text="Rental Date")

cursor = connection.execute("SELECT \* FROM " + TABLE\_NAME + " ;")

i = 0

for row in cursor:

tree.insert('', i, text="Customer " + str(row[0]),values=(row[1], row[2],row[3], row[4]))

i = i + 1

tree.pack()

secondWindow.mainloop()

button = tk.Button(root, text="Take input", command=takeNameInput)

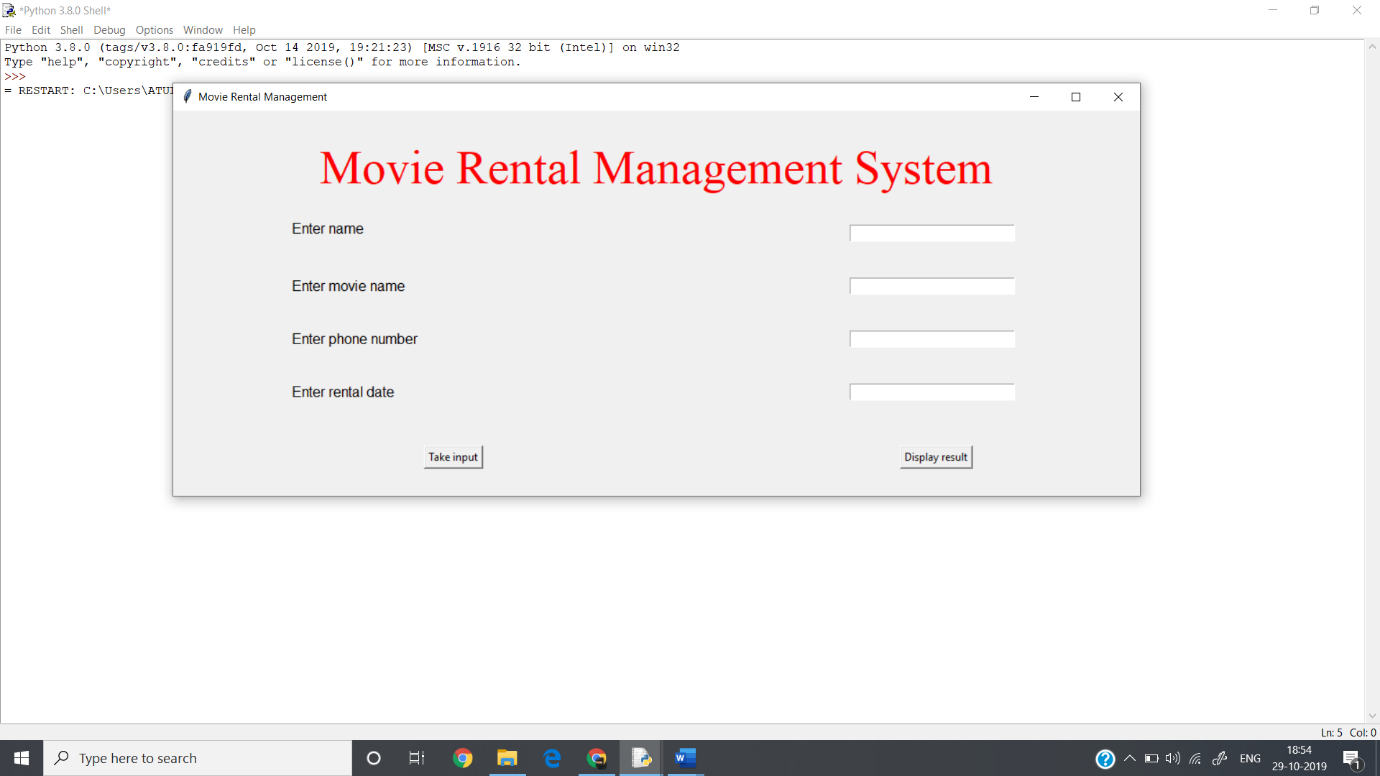
button.grid(row=5, column=0, pady=30)

displayButton = tk.Button(root, text="Display result", command=RootWindow)

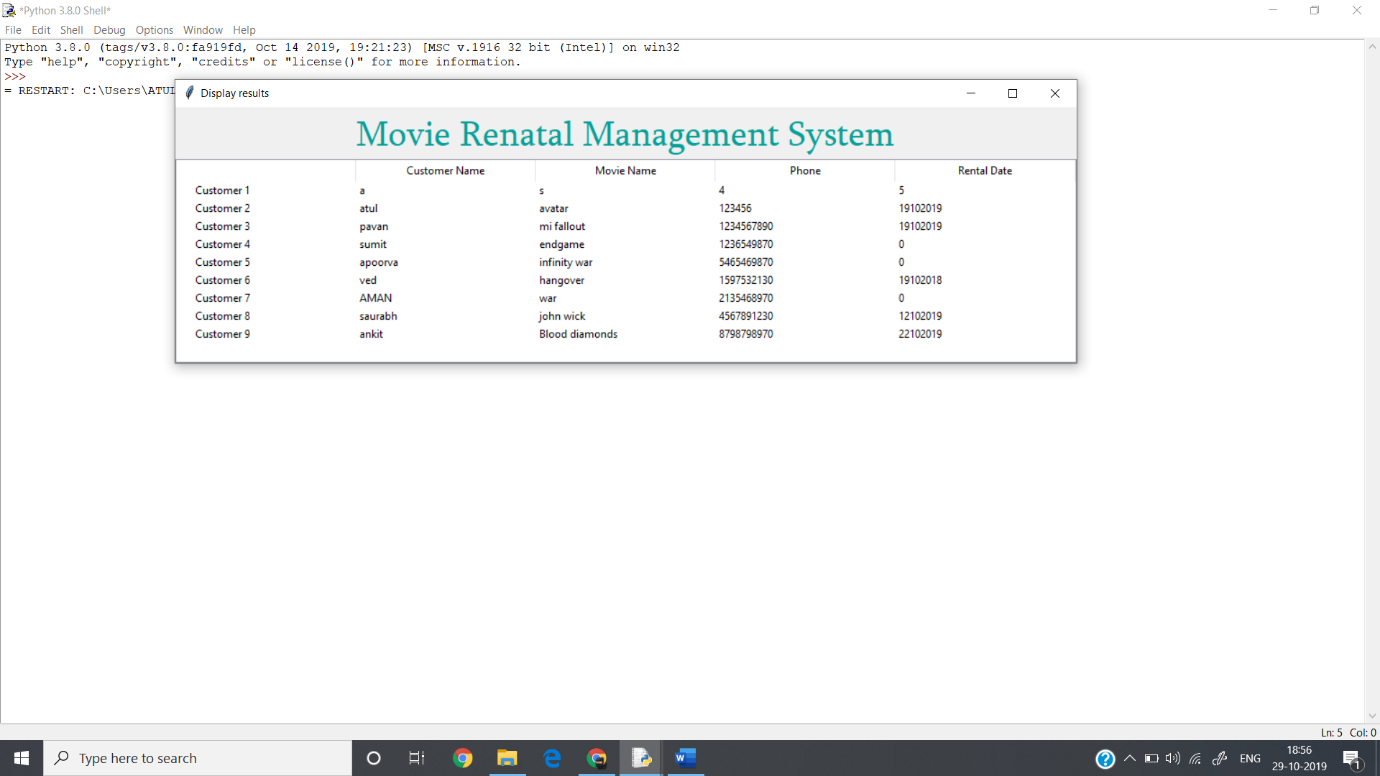
displayButton.grid(row=5, column=1)

root.mainloop()

**OUTPUT**

Entry Form:

Movie Rental Database:



**RESULT:**

A simple python is successfully used to keep records of transaction of borrowing, returning, fine adding and removing CDs and DVDs etc. in a database using any python application.

**REFERENCES:**

1. TUTORIALSPOINT
2. GEEKS FOR GEEKS
3. PYTHON DOCUMENTATION