**Klantenbeheer Systeem (CRM) - Gecombineerd Systeem**

**Dit uitgebreide CRM-systeem bevat 16+ functies die samen alle elementen van CRUD, zoek- en sorteerfunctionaliteit, statistieken, bestandsbeheer, en GUI integreren.**

**Hoofdstuk 1: Klanteninformatie Beheer**

**1. Klanten toevoegen:**

def voeg\_klant\_toe(klanten):

naam = input("Voer de naam van de klant in: ")

telefoon = input("Voer het telefoonnummer in: ")

email = input("Voer het e-mailadres in: ")

stad = input("Voer de stad in: ")

aankopen = float(input("Voer het totale aankoopbedrag in: "))

nieuwe\_klant = {

"Naam": naam,

"Telefoonnummer": telefoon,

"E-mail": email,

"Stad": stad,

"Aankopen": aankopen

}

klanten.append(nieuwe\_klant)

print(f"Klant {naam} is succesvol toegevoegd.")

klanten = []

**2. Klanteninformatie bijwerken:**

def update\_klant(klanten):

naam = input("Voer de naam van de klant in die je wilt bijwerken: ")

for klant in klanten:

if klant["Naam"].lower() == naam.lower():

nieuwe\_telefoon = input(f"Nieuwe telefoonnummer voor {naam}: ")

klant["Telefoonnummer"] = nieuwe\_telefoon

nieuwe\_aankopen = float(input(f"Nieuw aankoopbedrag voor {naam}: "))

klant["Aankopen"] = nieuwe\_aankopen

print(f"Gegevens van {naam zijn bijgewerkt.")

return

print(f"Klant {naam} niet gevonden.")

**3. Klanten verwijderen:**

def verwijder\_klant(klanten):

naam = input("Voer de naam van de klant in die je wilt verwijderen: ")

klanten[:] = [klant for klant in klanten if klant["Naam"].lower() != naam.lower()]

print(f"Klant {naam} is verwijderd.")

**Hoofdstuk 2: Zoeken en Sorteren**

**4. Klanten zoeken:**

def zoek\_klant(klanten, zoekterm):

resultaten = [klant for klant in klanten if zoekterm.lower() in klant["Naam"].lower() or zoekterm in klant["Telefoonnummer"]]

if resultaten:

for klant in resultaten:

print(klant)

else:

print("Geen klant gevonden.")

**5. Klanten sorteren op naam of stad:**

def sorteer\_klanten(klanten, sleutel="Naam"):

klanten.sort(key=lambda x: x[sleutel])

for klant in klanten:

print(klant)

**6. Meervoudige zoekfilters:**

def filter\_klanten(klanten, stad=None, minimaal\_aankoopbedrag=None):

gefilterd = klanten

if stad:

gefilterd = [klant for klant in gefilterd if klant["Stad"].lower() == stad.lower()]

if minimaal\_aankoopbedrag:

gefilterd = [klant for klant in gefilterd if klant["Aankopen"] >= minimaal\_aankoopbedrag]

for klant in gefilterd:

print(klant)

**Hoofdstuk 3: Statistieken**

**7. Totale waarde van aankopen berekenen:**

def totale\_aankopen(klanten):

totaal = sum(klant["Aankopen"] for klant in klanten)

print(f"Totale waarde van aankopen: €{totaal:.2f}")

**8. Gemiddelde aankoop per klant berekenen:**

def gemiddeld\_aankoopbedrag(klanten):

if klanten:

gemiddelde = sum(klant["Aankopen"] for klant in klanten) / len(klanten)

print(f"Gemiddeld aankoopbedrag per klant: €{gemiddelde:.2f}")

else:

print("Geen klanten gevonden.")

**9. Standaardafwijking van aankoopbedragen:**

import statistics

def standaardafwijking\_aankopen(klanten):

aankopen = [klant["Aankopen"] for klant in klanten]

if len(aankopen) > 1:

print(f"Standaardafwijking: {statistics.stdev(aankopen):.2f}")

else:

print("Niet genoeg gegevens voor standaardafwijking.")

**10. Meest waardevolle klanten identificeren:**

def meest\_waardevolle\_klanten(klanten):

hoogste\_klant = max(klanten, key=lambda klant: klant["Aankopen"])

print(f"Meest waardevolle klant: {hoogste\_klant['Naam']} met een totaal van €{hoogste\_klant['Aankopen']:.2f}")

**Hoofdstuk 4: Bestandsbeheer (CSV/JSON)**

**11. Klanten exporteren naar CSV:**

import csv

def schrijf\_klanten\_naar\_csv(klanten, bestand):

with open(bestand, mode='w', newline='', encoding='utf-8') as file:

writer = csv.DictWriter(file, fieldnames=klanten[0].keys())

writer.writeheader()

writer.writerows(klanten)

print(f"Klanteninformatie succesvol geëxporteerd naar {bestand}")

**12. Klanten importeren vanuit CSV:**

def lees\_klanten\_van\_csv(bestand):

with open(bestand, newline='', encoding='utf-8') as file:

reader = csv.DictReader(file)

klanten = list(reader)

return klanten

**13. Klanten exporteren naar JSON:**

import json

def schrijf\_klanten\_naar\_json(klanten, bestand):

with open(bestand, 'w') as file:

json.dump(klanten, file, indent=4)

print(f"Klanteninformatie succesvol geëxporteerd naar {bestand}")

**Hoofdstuk 5: Klantenquiz en Leren**

**14. Klantenquiz:**

import random

def klantenquiz(klanten):

klant = random.choice(klanten)

vraag = f"Hoeveel heeft {klant['Naam']} in totaal uitgegeven?"

antwoord = float(input(f"{vraag}: "))

if antwoord == klant['Aankopen']:

print("Correct!")

else:

print(f"Onjuist! Het juiste antwoord is €{klant['Aankopen']:.2f}")

**Hoofdstuk 6: GUI-Interface voor Klantenbeheer**

**15. Klantenbeheer GUI:**

import tkinter as tk

from tkinter import messagebox

def klantenbeheer\_gui():

root = tk.Tk()

root.title("Klantenbeheer Systeem")

label = tk.Label(root, text="Voer klantgegevens in:")

label.pack()

naam\_label = tk.Label(root, text="Naam:")

naam\_label.pack()

naam\_entry = tk.Entry(root)

naam\_entry.pack()

telefoon\_label = tk.Label(root, text="Telefoonnummer:")

telefoon\_label.pack()

telefoon\_entry = tk.Entry(root)

telefoon\_entry.pack()

stad\_label = tk.Label(root, text="Stad:")

stad\_label.pack()

stad\_entry = tk.Entry(root)

stad\_entry.pack()

aankoop\_label = tk.Label(root, text="Aankoopbedrag:")

aankoop\_label.pack()

aankoop\_entry = tk.Entry(root)

aankoop\_entry.pack()

def voeg\_toe():

naam = naam\_entry.get()

telefoon = telefoon\_entry.get()

stad = stad\_entry.get()

aankopen = float(aankoop\_entry.get())

nieuw\_klant = {"Naam": naam, "Telefoonnummer": telefoon, "Stad": stad, "Aankopen": aankopen}

klanten.append(nieuw\_klant)

messagebox.showinfo("Succes", f"Klant {naam} toegevoegd.")

button\_toevoegen = tk.Button(root, text="Voeg Klant Toe", command=voeg\_toe)

button\_toevoegen.pack()

root.mainloop()

gebruikers\_gui()