**Uitgebreid Voorraadbeheersysteem voor een Winkel - Functies en Uitleg**

**1. Producten inlezen uit CSV**

**We lezen de productinformatie in uit een CSV-bestand. Dit bestand bevat de naam, prijs, en voorraad van elk product.**

**Codevoorbeeld:**

import csv

def lees\_producten\_csv(bestand):

producten = []

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

reader = csv.DictReader(file)

for row in reader:

producten.append(row)

return producten

producten = lees\_producten\_csv('producten.csv')

print(producten)

**2. Product toevoegen**

**We voegen een nieuw product toe aan de voorraad met de naam, prijs, en hoeveelheid op voorraad.**

**Codevoorbeeld:**

def voeg\_product\_toe(producten):

naam = input("Voer de naam van het product in: ")

prijs = float(input("Voer de prijs in: "))

voorraad = int(input("Voer de voorraad in: "))

nieuw\_product = {

"Naam": naam,

"Prijs": prijs,

"Voorraad": voorraad

}

producten.append(nieuw\_product)

print(f"Product {naam} is toegevoegd.")

voeg\_product\_toe(producten)

**3. Product verwijderen**

**Deze functie verwijdert een product uit de voorraad op basis van de ingevoerde naam.**

**Codevoorbeeld:**

def verwijder\_product(producten):

naam = input("Voer de naam van het product in dat je wilt verwijderen: ")

for product in producten:

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

producten.remove(product)

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

return

print(f"Product {naam} is niet gevonden.")

verwijder\_product(producten)

**4. Product wijzigen**

**We kunnen de informatie van een product zoals de prijs of voorraad aanpassen.**

**Codevoorbeeld:**

def wijzig\_product(producten):

naam = input("Voer de naam van het product in dat je wilt wijzigen: ")

for product in producten:

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

nieuwe\_prijs = float(input(f"Nieuwe prijs voor {naam}: "))

nieuwe\_voorraad = int(input(f"Nieuwe voorraad voor {naam}: "))

product["Prijs"] = nieuwe\_prijs

product["Voorraad"] = nieuwe\_voorraad

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

return

print(f"Product {naam} is niet gevonden.")

wijzig\_product(producten)

**5. Productvoorraad weergeven**

**Deze functie toont een overzicht van alle producten met hun voorraadniveaus.**

**Codevoorbeeld:**

def toon\_voorraad(producten):

print("Voorraadoverzicht:")

for product in producten:

print(f"{product['Naam']}: {product['Voorraad']} stuks")

toon\_voorraad(producten)

**6. Voorraad sorteren op naam**

**Deze functie sorteert de producten alfabetisch op naam.**

**Codevoorbeeld:**

def sorteer\_op\_naam(producten):

producten.sort(key=lambda product: product["Naam"].lower())

toon\_voorraad(producten)

sorteer\_op\_naam(producten)

**7. Voorraad sorteren op prijs**

**Deze functie sorteert de producten op prijs (laag naar hoog).**

**Codevoorbeeld:**

def sorteer\_op\_prijs(producten):

producten.sort(key=lambda product: float(product["Prijs"]))

toon\_voorraad(producten)

sorteer\_op\_prijs(producten)

**8. Waarschuwing bij lage voorraad**

**We geven een waarschuwing als de voorraad van een product onder een bepaalde limiet komt.**

**Codevoorbeeld:**

def waarschuwing\_lage\_voorraad(producten, limiet=10):

print("Waarschuwing voor lage voorraad:")

for product in producten:

if product["Voorraad"] < limiet:

print(f"Product {product['Naam']} heeft nog maar {product['Voorraad']} stuks.")

waarschuwing\_lage\_voorraad(producten, limiet=5)

**9. Totale waarde van de voorraad berekenen**

**Deze functie berekent de totale waarde van alle producten in de voorraad.**

**Codevoorbeeld:**

def totale\_voorraad\_waarde(producten):

totale\_waarde = sum(float(product["Prijs"]) \* int(product["Voorraad"]) for product in producten)

print(f"Totale waarde van de voorraad: €{totale\_waarde:.2f}")

totale\_voorraad\_waarde(producten)

**10. Gemiddelde prijs van producten berekenen**

**Deze functie berekent de gemiddelde prijs van alle producten in de voorraad.**

**Codevoorbeeld:**

def gemiddelde\_prijs\_producten(producten):

gemiddelde\_prijs = sum(float(product["Prijs"]) for product in producten) / len(producten)

print(f"Gemiddelde prijs van producten: €{gemiddelde\_prijs:.2f}")

gemiddelde\_prijs\_producten(producten)

**11. Exporteren naar CSV**

**We exporteren de producten en hun gegevens naar een CSV-bestand.**

**Codevoorbeeld:**

def schrijf\_voorraad\_naar\_csv(producten, bestand):

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

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

writer.writeheader()

writer.writerows(producten)

print(f"Producten succesvol opgeslagen in {bestand}.")

schrijf\_voorraad\_naar\_csv(producten, 'voorraad\_export.csv')

**12. Productgegevens zoeken**

**We zoeken naar een product op basis van de naam en geven de details weer.**

**Codevoorbeeld:**

def zoek\_product(producten, naam):

for product in producten:

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

print(f"Gevonden: {product}")

return product

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

return None

naam = input("Voer de naam van het product in om te zoeken: ")

zoek\_product(producten, naam)

**13. Verkoop registreren**

**Bij elke verkoop wordt de voorraad van een product verminderd op basis van de verkochte hoeveelheid.**

**Codevoorbeeld:**

def registreer\_verkoop(producten):

naam = input("Voer de naam van het product in: ")

product = zoek\_product(producten, naam)

if product:

verkocht = int(input(f"Hoeveel {naam} zijn verkocht?: "))

product["Voorraad"] -= verkocht

print(f"Voorraad bijgewerkt voor {naam}: {product['Voorraad']} stuks.")

registreer\_verkoop(producten)

**14. Inkoop registreren**

**Bij elke inkoop wordt de voorraad van een product verhoogd.**

**Codevoorbeeld:**

def registreer\_inkoop(producten):

naam = input("Voer de naam van het product in: ")

product = zoek\_product(producten, naam)

if product:

ingekocht = int(input(f"Hoeveel {naam} zijn ingekocht?: "))

product["Voorraad"] += ingekocht

print(f"Voorraad bijgewerkt voor {naam}: {product['Voorraad']} stuks.")

registreer\_inkoop(producten)

**15. Winst per product berekenen**

**Deze functie berekent de winst voor elk verkocht product (verkoopprijs minus kostprijs).**

**Codevoorbeeld:**

def bereken\_winst(producten, kostprijs):

naam = input("Voer de naam van het product in: ")

product = zoek\_product(producten, naam)

if product:

verkoopprijs = float(product["Prijs"])

winst = verkoopprijs - kostprijs

print(f"Winst per verkocht product {naam}: €{winst:.2f}")

bereken\_winst(producten, kostprijs=10.0)

**16. Voorraad visualiseren met grafieken**

**We kunnen een grafiek tonen met de voorraad van elk product.**

**Codevoorbeeld:**

import matplotlib.pyplot as plt

def visualiseer\_voorraad(producten):

namen = [product["Naam"] for product in producten]

voorraden = [product["Voorraad"] for product in producten]

plt.bar(namen, voorraden, color='blue')

plt.title("Voorraad per product")

plt.xlabel("Producten")

plt.ylabel("Voorraad")

plt.show()

visualiseer\_voorraad(producten)

**17. Populairste producten bijhouden**

**We houden het aantal verkochte producten bij en tonen de populairste producten.**

**Codevoorbeeld:**

def populairste\_producten(producten, verkoopdata):

verkoop\_totalen = {}

for verkoop in verkoopdata:

if verkoop["Naam"] in verkoop\_totalen:

verkoop\_totalen[verkoop["Naam"]] += verkoop["Aantal"]

else:

verkoop\_totalen[verkoop["Naam"]] = verkoop["Aantal"]

populairste = sorted(verkoop\_totalen.items(), key=lambda item: item[1], reverse=True)

print("Populairste producten:")

for product, aantal in populairste:

print(f"{product}: {aantal} keer verkocht")

verkoopdata = [{"Naam": "Laptop", "Aantal": 3}, {"Naam": "Telefoon", "Aantal": 5}]

populairste\_producten(producten, verkoopdata)

**18. Voorraad per categorie beheren**

**We beheren de voorraad per categorie, bijvoorbeeld elektronica, kleding, enz.**

**Codevoorbeeld:**

def voorraad\_per\_categorie(producten):

categorieen = {}

for product in producten:

categorie = product.get("Categorie", "Onbekend")

if categorie in categorieen:

categorieen[categorie].append(product)

else:

categorieen[categorie] = [product]

for categorie, producten in categorieen.items():

print(f"Categorie: {categorie}")

for product in producten:

print(f" - {product['Naam']}: {product['Voorraad']} stuks")

voorraad\_per\_categorie(producten)

**19. Bulk toevoeging van producten**

**We kunnen meerdere producten tegelijk toevoegen via een CSV-bestand.**

**Codevoorbeeld:**

def bulk\_toevoeging\_van\_producten(bestand, producten):

nieuwe\_producten = lees\_producten\_csv(bestand)

producten.extend(nieuwe\_producten)

print(f"{len(nieuwe\_producten)} producten toegevoegd.")

bulk\_toevoeging\_van\_producten('nieuwe\_producten.csv', producten)

**20. Geavanceerde filters toepassen**

**We passen filters toe op basis van prijs, voorraad, of categorie.**

**Codevoorbeeld:**

def filter\_producten(producten, prijs=None, voorraad=None, categorie=None):

gefilterde\_producten = producten

if prijs:

gefilterde\_producten = [product for product in gefilterde\_producten if float(product["Prijs"]) <= prijs]

if voorraad:

gefilterde\_producten = [product for product in gefilterde\_producten if product["Voorraad"] >= voorraad]

if categorie:

gefilterde\_producten = [product for product in gefilterde\_producten if product.get("Categorie") == categorie]

print("Gefilterde producten:")

for product in gefilterde\_producten:

print(product)

filter\_producten(producten, prijs=100, voorraad=10, categorie="Elektronica")

import csv

import matplotlib.pyplot as plt

# Functie om producten in te lezen uit een CSV-bestand

def lees\_producten\_csv(bestand):

producten = []

try:

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

reader = csv.DictReader(file)

for row in reader:

# Zorg ervoor dat de waarden de juiste datatypen hebben

row["Prijs"] = float(row["Prijs"])

row["Voorraad"] = int(row["Voorraad"])

producten.append(row)

print(f"{len(producten)} producten geladen uit {bestand}.")

except FileNotFoundError:

print(f"Bestand {bestand} niet gevonden. Beginnen met een lege productenlijst.")

return producten

# Functie om een product toe te voegen

def voeg\_product\_toe(producten):

naam = input("Voer de naam van het product in: ")

prijs = float(input("Voer de prijs in: "))

voorraad = int(input("Voer de voorraad in: "))

nieuw\_product = {

"Naam": naam,

"Prijs": prijs,

"Voorraad": voorraad

}

producten.append(nieuw\_product)

print(f"Product {naam} is toegevoegd.")

# Functie om een product te verwijderen

def verwijder\_product(producten):

naam = input("Voer de naam van het product in dat je wilt verwijderen: ")

for product in producten:

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

producten.remove(product)

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

return

print(f"Product {naam} is niet gevonden.")

# Functie om een product te wijzigen

def wijzig\_product(producten):

naam = input("Voer de naam van het product in dat je wilt wijzigen: ")

for product in producten:

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

nieuwe\_prijs = float(input(f"Nieuwe prijs voor {naam}: "))

nieuwe\_voorraad = int(input(f"Nieuwe voorraad voor {naam}: "))

product["Prijs"] = nieuwe\_prijs

product["Voorraad"] = nieuwe\_voorraad

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

return

print(f"Product {naam} is niet gevonden.")

# Functie om de voorraad weer te geven

def toon\_voorraad(producten):

print("\nVoorraadoverzicht:")

for product in producten:

print(f"{product['Naam']}: {product['Voorraad']} stuks")

# Functie om de voorraad te sorteren op naam

def sorteer\_op\_naam(producten):

producten.sort(key=lambda product: product["Naam"].lower())

toon\_voorraad(producten)

# Functie om de voorraad te sorteren op prijs

def sorteer\_op\_prijs(producten):

producten.sort(key=lambda product: float(product["Prijs"]))

toon\_voorraad(producten)

# Functie om waarschuwingen te geven bij lage voorraad

def waarschuwing\_lage\_voorraad(producten, limiet=10):

print("\nWaarschuwing voor lage voorraad:")

lage\_voorraad = False

for product in producten:

if product["Voorraad"] < limiet:

print(f"Product {product['Naam']} heeft nog maar {product['Voorraad']} stuks.")

lage\_voorraad = True

if not lage\_voorraad:

print("Geen producten met lage voorraad.")

# Functie om de totale waarde van de voorraad te berekenen

def totale\_voorraad\_waarde(producten):

totale\_waarde = sum(float(product["Prijs"]) \* int(product["Voorraad"]) for product in producten)

print(f"\nTotale waarde van de voorraad: €{totale\_waarde:.2f}")

# Functie om de gemiddelde prijs van producten te berekenen

def gemiddelde\_prijs\_producten(producten):

gemiddelde\_prijs = sum(float(product["Prijs"]) for product in producten) / len(producten)

print(f"\nGemiddelde prijs van producten: €{gemiddelde\_prijs:.2f}")

# Functie om de voorraad naar CSV te schrijven

def schrijf\_voorraad\_naar\_csv(producten, bestand):

if producten:

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

fieldnames = ["Naam", "Prijs", "Voorraad"]

writer = csv.DictWriter(file, fieldnames=fieldnames)

writer.writeheader()

writer.writerows(producten)

print(f"\nProducten succesvol opgeslagen in {bestand}.")

else:

print("Geen producten om op te slaan.")

# Functie om een product te zoeken

def zoek\_product(producten, naam):

for product in producten:

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

print(f"Gevonden: {product}")

return product

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

return None

# Functie om een verkoop te registreren

def registreer\_verkoop(producten):

naam = input("Voer de naam van het product in: ")

product = zoek\_product(producten, naam)

if product:

verkocht = int(input(f"Hoeveel {naam} zijn verkocht?: "))

if verkocht <= product["Voorraad"]:

product["Voorraad"] -= verkocht

print(f"Voorraad bijgewerkt voor {naam}: {product['Voorraad']} stuks.")

else:

print("Niet genoeg voorraad voor deze verkoop.")

# Functie om een inkoop te registreren

def registreer\_inkoop(producten):

naam = input("Voer de naam van het product in: ")

product = zoek\_product(producten, naam)

if product:

ingekocht = int(input(f"Hoeveel {naam} zijn ingekocht?: "))

product["Voorraad"] += ingekocht

print(f"Voorraad bijgewerkt voor {naam}: {product['Voorraad']} stuks.")

# Functie om winst per product te berekenen

def bereken\_winst(producten):

naam = input("Voer de naam van het product in: ")

kostprijs = float(input(f"Voer de kostprijs van {naam} in: "))

product = zoek\_product(producten, naam)

if product:

verkoopprijs = float(product["Prijs"])

winst = verkoopprijs - kostprijs

print(f"Winst per verkocht product {naam}: €{winst:.2f}")

# Functie om de voorraad te visualiseren

def visualiseer\_voorraad(producten):

namen = [product["Naam"] for product in producten]

voorraden = [product["Voorraad"] for product in producten]

plt.bar(namen, voorraden, color='blue')

plt.title("Voorraad per product")

plt.xlabel("Producten")

plt.ylabel("Voorraad")

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

# Functie om de voorraad per categorie te beheren

def voorraad\_per\_categorie(producten):

categorieen = {}

for product in producten:

categorie = product.get("Categorie", "Onbekend")

if categorie in categorieen:

categorieen[categorie].append(product)

else:

categorieen[categorie] = [product]

for categorie, producten\_lijst in categorieen.items():

print(f"\nCategorie: {categorie}")

for product in producten\_lijst:

print(f" - {product['Naam']}: {product['Voorraad']} stuks")

# Functie om producten te filteren

def filter\_producten(producten):

prijs = input("Maximale prijs (laat leeg om over te slaan): ")

voorraad = input("Minimale voorraad (laat leeg om over te slaan): ")

categorie = input("Categorie (laat leeg om over te slaan): ")

gefilterde\_producten = producten

if prijs:

prijs = float(prijs)

gefilterde\_producten = [product for product in gefilterde\_producten if float(product["Prijs"]) <= prijs]

if voorraad:

voorraad = int(voorraad)

gefilterde\_producten = [product for product in gefilterde\_producten if product["Voorraad"] >= voorraad]

if categorie:

gefilterde\_producten = [product for product in gefilterde\_producten if product.get("Categorie", "").lower() == categorie.lower()]

print("\nGefilterde producten:")

for product in gefilterde\_producten:

print(f"{product['Naam']} - Prijs: €{product['Prijs']} - Voorraad: {product['Voorraad']} stuks")

# Hoofdprogramma

def main():

producten = lees\_producten\_csv('producten.csv')

while True:

print("\nVoorraadbeheersysteem Menu:")

print("1. Toon voorraad")

print("2. Voeg product toe")

print("3. Verwijder product")

print("4. Wijzig product")

print("5. Sorteer op naam")

print("6. Sorteer op prijs")

print("7. Waarschuwing lage voorraad")

print("8. Totale waarde voorraad")

print("9. Gemiddelde prijs producten")

print("10. Schrijf voorraad naar CSV")

print("11. Zoek product")

print("12. Registreer verkoop")

print("13. Registreer inkoop")

print("14. Bereken winst per product")

print("15. Visualiseer voorraad")

print("16. Voorraad per categorie")

print("17. Filter producten")

print("18. Afsluiten")

keuze = input("Maak een keuze (1-18): ")

if keuze == '1':

toon\_voorraad(producten)

elif keuze == '2':

voeg\_product\_toe(producten)

elif keuze == '3':

verwijder\_product(producten)

elif keuze == '4':

wijzig\_product(producten)

elif keuze == '5':

sorteer\_op\_naam(producten)

elif keuze == '6':

sorteer\_op\_prijs(producten)

elif keuze == '7':

limiet = int(input("Voer de limiet in voor lage voorraad: "))

waarschuwing\_lage\_voorraad(producten, limiet)

elif keuze == '8':

totale\_voorraad\_waarde(producten)

elif keuze == '9':

gemiddelde\_prijs\_producten(producten)

elif keuze == '10':

bestand = input("Voer de bestandsnaam in om op te slaan (bijv. voorraad\_export.csv): ")

schrijf\_voorraad\_naar\_csv(producten, bestand)

elif keuze == '11':

naam = input("Voer de naam van het product in om te zoeken: ")

zoek\_product(producten, naam)

elif keuze == '12':

registreer\_verkoop(producten)

elif keuze == '13':

registreer\_inkoop(producten)

elif keuze == '14':

bereken\_winst(producten)

elif keuze == '15':

visualiseer\_voorraad(producten)

elif keuze == '16':

voorraad\_per\_categorie(producten)

elif keuze == '17':

filter\_producten(producten)

elif keuze == '18':

print("Programma wordt afgesloten.")

break

else:

print("Ongeldige keuze, probeer opnieuw.")

if \_\_name\_\_ == "\_\_main\_\_":

main()