**Assignment: Python Programming for DL**

Name: N Prajan Selvaraj

Register Number:192321071

Department:Btech IT

Date of Submission:17/07/24

**Problem 2: : Inventory Management System Optimization**

**Scenario:**

You have been hired by a retail company to optimize their inventory management system. The company wants to minimize stockouts and overstock situations while maximizing inventory turnover and profitability.

**Tasks:**

1. **Model the inventory system**: Define the structure of the inventory system, including products, warehouses, and current stock levels.

2. **Implement an inventory tracking application**: Develop a Python application that tracks inventory levels in real-time and alerts when stock levels fall below a certain threshold.

3. **Optimize inventory ordering**: Implement algorithms to calculate optimal reorder points and quantities based on historical sales data, lead times, and demand forecasts.

4. **Generate reports**: Provide reports on inventory turnover rates, stockout occurrences, and cost implications of overstock situations.

5. **User interaction**: Allow users to input product IDs or names to view current stock levels, reorder recommendations, and historical data.

**Deliverables:**

* **Data Flow Diagram**: Illustrate how data flows within the inventory management system, from input (e.g., sales data, inventory adjustments) to output (e.g., reorder alerts, reports).
* **Pseudocode and Implementation**: Provide pseudocode and actual code demonstrating how inventory levels are tracked, reorder points are calculated, and reports are generated.
* **Documentation**: Explain the algorithms used for reorder optimization, how historical data influences decisions, and any assumptions made (e.g., constant lead times).
* **User Interface**: Develop a user-friendly interface for accessing inventory information, viewing reports, and receiving alerts.
* **Assumptions and Improvements**: Discuss assumptions about demand patterns, supplier reliability, and potential improvements for the inventory management system's efficiency and accuracy.

Solution:

# Real-Time Weather Monitoring System

# 1.Data Flow Diagram

**Creating a inventory system**

**Ask the user to choose option**

**Ask the user to for amount in stock**

**Ask user to continue order or exit**

# 2. Implementation

|  |
| --- |
| import pandas as pd  import json  import os.path  import time  import random  # Creating Dictionary to store data  available\_products = {1001: {"name": "avocado", "price": 230,                               "category": "grocery",                               "quantity": 10, "date": "10/03/2021"},                        1002: {"name": "lotion", "price": 250,                               "category": "beauty & personal",                               "quantity": 100,                               "date": "15/07/2021"},                        1003: {"name": "pain reliever", "price": 500,                               "category": "health",                               "quantity": 200, "date": "12/04/2021"},                        1004: {"name": "dry pasta", "price": 20,                               "category": "grocery",                               "quantity": 50, "date": "27/06/2021"},                        1005: {"name": "toothbrush", "price": 700,                               "category": "beauty & personal",                               "quantity": 100,                               "date": "30/01/2021"},                        1006: {"name": "halloween candy", "price": 33,                               "category": "grocery",                               "quantity": 56, "date": "22/02/2021"},                        1007: {"name": "mascara", "price": 765,                               "category": "beauty & personal",                               "quantity": 70,                               "date": "11/03/2021"},                        1008: {"name": "capsicum", "price": 764,                               "category": "grocery",                               "quantity": 90, "date": "16/02/2021"},                        1009: {"name": "blush", "price": 87,                               "category": "beauty & personal",                               "quantity": 50, "date": "17/07/2021"},                        1010: {"name": "granola bars", "price": 24,                               "category": "grocery", "quantity": 60,                               "date": "20/05/2021"},                        }  # Formatting Dictionary into JSON format  js = json.dumps(available\_products)  #  json.dumps() function converts a  # Python object into a json string  js  # so we got all data in json string format here  # Create Json File for DataBase and Write data Into File  fd = open("data.json", 'w')  # it will open file into write mode if file  # does not exists then it will create file too'''  fd.write(js)  # writing string into file  fd.close()  # Close File After Inserting Data  def admin():      print("========\      Welcome to the Admin Inventory Management System \      ==============")      while (1):          print("1)Display DataBase/All Products with there details")          print("2)Display Specific Product with its details")          print("3)Insert Data Into DataBase")          print("4)Update Product in Database")          print("5)Delete Product in DataBase")          print("6)Display User Purchase Reports")          print("7)Exit")          print("Enter Your Choice :- ")          n = int(input())          if (n == 1):              display\_data()          elif (n == 2):              display\_specific\_data()          elif (n == 3):              add\_new()          elif (n == 4):              update\_prod\_data()          elif (n == 5):              delete\_prod()          elif (n == 6):              display\_reports\_admin()          elif (n == 7):              break          else:              print("Invalid Choice...!!!")  def display\_data():      fd = open("data.json", 'r')      txt = fd.read()  # reading data from file      data = json.loads(txt)      # This will parse the JSON data, populates a      # Python dictionary with the data      fd.close()      print("Enter '0' To Display Data Category Wise or '1' \      To Show Data As its Sequence Of Insertion :- ")      n = int(input())      # Display All Records      if (n == 1):          table = pd.DataFrame(              columns=['ID', 'name', 'price', 'category', 'quantity', 'date'])          # Creating Pandas dataframe to show data in table format later          for i in data.keys():              # Fetch all keys in dictionary              temp = pd.DataFrame(columns=['ID'])              temp['ID'] = [i]              for j in data[i].keys():                  temp[j] = [data[i][j]]              table = table.append(temp)          table = table.reset\_index(drop=True)          '''This will reset index of dataframe'''          from IPython.display import display          display(table)      elif (n == 0):          # Display Records by Category          table = pd.DataFrame(              columns=['ID', 'name', 'price', 'category',                       'quantity', 'date'])          cat = []          for i in data.keys():              temp = pd.DataFrame(columns=['ID'])              temp['ID'] = [i]              for j in data[i].keys():                  temp[j] = [data[i][j]]                  if (j == 'category'):                      cat.append(data[i][j])              table = table.append(temp)              table = table.reset\_index(drop=True)              cat = set(cat)              cat = list(cat)          for k in cat:              temp = pd.DataFrame()              temp = table[table['category'] == k]              print("Data Of Products Of Category "+k+" is:- ")              from IPython.display import display              display(temp)      else:          print("Enter Valid Choice...!!!")  # display\_data() # Uncomment This Line To Run This Function  def display\_specific\_data():      fd = open("data.json", 'r')      txt = fd.read()      data = json.loads(txt)      fd.close()      print("Enter Product ID Whose Details You Want to Have a Look on :- ")      i = input()      # Following Code will Filter out Product ID from Records      if i in data.keys():          temp = pd.DataFrame(columns=['ID'])          temp['ID'] = [i]          for j in data[i].keys():              temp[j] = [data[i][j]]          from IPython.display import display          display(temp)      else:          print("You Have Entered Wrong Product ID\          that is not Present in DataBase...!!!")  # display\_specific\_data() # Uncomment This Line To Run This Function  def add\_new():      fd = open("data.json", 'r')      txt = fd.read()      data = json.loads(txt)      fd.close()      print("Enter New Product ID :- ")      id = input()      if id not in data.keys():          print("Enter Product Name :- ")          name = input()          print("Enter Price of Product(price for product quantity as 1) :- ")          price = input()          print("Enter Category of Product :- ")          category = input()          print("Enter Quantity of Product :- ")          quantity = input()          print("Enter The Date on Which Product is Added in Inventory :- ")          date = input()          data[id] = {'name': name, 'price': price,                      'category': category, 'quantity': quantity, 'date': date}          print("Please Press '0' to Add New\          Attributes/Properties of Product or Press '1' to Continue :- ")          z = int(input())          if(z == 0):              print("Enter Number of New Attributes/Properties of Product :- ")              n = int(input())              for i in range(n):                  print("Enter Attribute Name That you Want To Add :- ")                  nam = input()                  print("Enter The "+str(nam)+" of Product :- ")                  pro = input()                  data[id][nam] = pro          print("Product ID "+str(id)+" Added Successfully...!!!")      else:          print("The Product ID you Have Entered Is\          Already Present in DataBase Please Check...!!!")      js = json.dumps(data)      fd = open("data.json", 'w')      fd.write(js)      fd.close()  # add\_new() # Uncomment This Line To Run This Function  def delete\_prod():      fd = open("data.json", 'r')      txt = fd.read()      data = json.loads(txt)      fd.close()      print("Enter The Product ID of The Product Which You Want To Delete :- ")      temp = input()      if temp in data.keys():          data.pop(temp)  # here we are removing that particular data          print("Product ID "+str(temp)+" Deleted Successfully...!!!")      else:          print("Invalid Product ID...!!!")      js = json.dumps(data)      fd = open("data.json", 'w')      fd.write(js)      fd.close()  # delete\_prod() # Uncomment This Line To Run This Function  def update\_prod\_data():      fd = open("data.json", 'r')      txt = fd.read()      data = json.loads(txt)      fd.close()      print("Enter The Product ID of The Product\      Which You Want To Update :- ")      temp = input()      if temp in data.keys():          print("Want to update whole product data\          press '0' else '1' for specific data :- ")          q = int(input())          if (q == 0):              print("Enter Product Name :- ")              name = input()              print("Enter Price of Product(price for\              product quantity as 1) :- ")              price = input()              print("Enter Category of Product :- ")              category = input()              print("Enter Quantity of Product :- ")              quantity = input()              print("Enter The Date on Which Product\              is Added in Inventory :- ")              date = input()              data[temp] = {'name': name, 'price': price,                            'category': category, 'quantity': quantity,                            'date': date}              print(                  "Please Press '0' to Add more Attributes/Properties of Product or Press '1' to Continue :- ")              z = int(input())              if(z == 0):                  print("Enter Number of New Attributes/Properties of Product :- ")                  n = int(input())                  for i in range(n):                      print("Enter Attribute Name That you Want To Add :- ")                      nam = input()                      print("Enter The "+str(nam)+" of Product :- ")                      pro = input()                      data[temp][nam] = pro              print("Product ID "+str(temp)+" Updated Successfully...!!!")          elif(q == 1):              print("Enter Which Attribute of Product You want to Update :- ")              p = input()              if p in data[temp].keys():                  print("Enter "+str(p)+" of Product :- ")                  u = input()                  data[temp][p] = u                  print("Product ID "+str(temp)+"'s attribute " +                        str(p)+" is Updated Successfully...!!!")              else:                  print("Invalid Product Attribute...!!!")          else:              print("Invalid Choice...!!!")      else:          print("Invalid Product ID...!!!")      js = json.dumps(data)      fd = open("data.json", 'w')      fd.write(js)      fd.close()  # update\_prod\_data() # Uncomment This Line To Run This Function  def display\_reports\_admin():      if (os.path.isfile("user\_data.json") is False):          # Check for if file is present or not          # File will be generated only if any user will do some purchase          print("No User Reports are Present")          return      fd = open("user\_data.json", 'r')      txt = fd.read()      user\_data = json.loads(txt)      fd.close()      print("Enter '0' to Check All Bills/Reports\      and '1' To Check Specific User Bills/Reports :- ")      n = int(input())      if (n == 1):          print("Enter User ID Whose Details You Want to Have a Look on")          i = input()          temp = pd.DataFrame()          if i in user\_data.keys():              for j in user\_data[i].keys():                  d = dict()                  d['User ID'] = i                  d['Purchase Number'] = j                  for k in user\_data[i][j].keys():                      d[k] = user\_data[i][j][k]                  temp = temp.append(d, ignore\_index=True)                  d = dict()              temp = temp.reset\_index(drop=True)              from IPython.display import display              display(temp)          else:              print("You Have Entered Wrong User ID that is not Present in DataBase...!!!")      elif (n == 0):          table = pd.DataFrame()          for i in user\_data.keys():              temp = pd.DataFrame()              for j in user\_data[i].keys():                  d = dict()                  d['User ID'] = i                  d['Purchase Number'] = j                  for k in user\_data[i][j].keys():                      d[k] = user\_data[i][j][k]                  temp = temp.append(d, ignore\_index=True)                  d = dict()              table = table.append(temp)          table = table.reset\_index(drop=True)          from IPython.display import display          display(table)      else:          print("Please Enter Valid Choice...!!!")  # display\_reports\_admin() # Uncomment This Line To Run This Function  def delete\_all():      fd = open("data.json", 'r')      txt = fd.read()      data = json.loads(txt)      fd.close()      data = {}  # Replacing Data with NULL Dictionary      js = json.dumps(data)      fd = open("data.json", 'w')      fd.write(js)      fd.close()  def user():      print("======= Welcome to the User Inventory Management System ====")      while (1):          print("1)Display All Products With Details")          print("2)Display Specific Product With Details")          print("3)Display All Purchase Bills")          print("4)Buy The Product")          print("5)Exit")          print("Enter Your Choice :- ")          n = int(input())          if (n == 1):              display\_data()          elif (n == 2):              display\_specific\_data()          elif (n == 3):              display\_user\_data()          elif (n == 4):              buy\_product()          elif (n == 5):              break          else:              print("Invalid Choice...!!!")  def display\_user\_data():      if (os.path.isfile("user\_data.json") is False):          print("No User Reports are Present")          return      fd = open("user\_data.json", 'r')      txt = fd.read()      user\_data = json.loads(txt)      fd.close()      print("Enter your User ID to Display All your Bills :- ")      i = input()      temp = pd.DataFrame()      if i in user\_data.keys():          for j in user\_data[i].keys():              d = dict()              d['User ID'] = i              d['Purchase Number'] = j              for k in user\_data[i][j].keys():                  d[k] = user\_data[i][j][k]              temp = temp.append(d, ignore\_index=True)              d = dict()          temp = temp.reset\_index(drop=True)          from IPython.display import display          display(temp)      else:          print("You Have Entered Wrong User ID that is not Present in DataBase...!!!")  def generate\_bill(user\_id, prod\_id, price, time\_date, purchase\_no,                    name, category, quantity\_all, transaction\_id):      print("========= Bill ========")      print("#######################")      print("   User ID :-", user\_id)      print("#################")      amount = 0      n = len(purchase\_no)      for i in range(n):          print("-----------------------------------------")          amount = amount+float(price[i])\*float(quantity\_all[i])          print("Purchase number", purchase\_no[i],                "\nPurchase Time :-", time\_date[i], "\nProduct ID :-",                prod\_id[i], "\nName Of Product :-",                name[i], "\nCategory Of Product :-", category[i],                "\nPrice of Product per Item :-", price[i],                "\nPurchase Quantity :-", quantity\_all[i])          print("-----------------------------------")      print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")      print("   Total Payable Bill :-",            amount, "Transaction ID :-", transaction\_id)      print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  def buy\_product():      if (os.path.isfile("user\_data.json") is False):          user\_data = {}      else:          fd = open("user\_data.json", 'r')          txt = fd.read()          user\_data = json.loads(txt)          fd.close()      fd = open("data.json", 'r')      txt = fd.read()      data = json.loads(txt)      fd.close()      print("Enter Your User ID if You are Old \      Customer else press '0' To New User ID :- ")      p = int(input())      if (p == 0):          if (len(user\_data.keys()) == 0):              user\_id = 1000          else:              user\_id = int(list(user\_data.keys())[-1])+1      else:          if str(p) in user\_data.keys():              user\_id = p          else:              user\_id = -1      if (user\_id != -1):          user\_id = str(user\_id)          price = []          time\_date = []          purchase\_no = []          name = []          category = []          quantity\_all = []          prod\_id = []          transaction\_id = ''.join(random.choice(              '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ') for i in range(10))          print("Enter Number of Products You Want To Buy :- ")          n = int(input())          print("Enter Data As Follows :- ")          if user\_id not in user\_data.keys():              user\_data[user\_id] = {}              g = 0          else:              g = int(list(user\_data[user\_id].keys())[-1])+1          for i in range(n):              print("Enter Product ID of Product " +                    str(i+1)+" that you want to buy")              id = input()              if id in data.keys():                  user\_data[user\_id][str(i+1+g)] = {}                  user\_data[user\_id][str(i+1+g)]['time\_date'] = str(time.ctime())                  time\_date.append(str(time.ctime()))                  if(float(data[id]['quantity']) == 0.0):                      print("Product You Want is Currently Out Of Stock...!!!")                      continue                  purchase\_no.append(i+1+g)                  name.append(data[id]['name'])                  user\_data[user\_id][str(i+1+g)]['name'] = data[id]['name']                  prod\_id.append(id)                  user\_data[user\_id][str(i+1+g)]['product\_id'] = id                  category.append(data[id]['category'])                  user\_data[user\_id][str(                      i+1+g)]['category'] = data[id]['category']                  print("For Product "+str(data[id]['name']) +                        " Available Quantity is :- "+str(data[id]['quantity']))                  print("Enter Quantity of Product " +                        str(i+1)+" that you want to buy")                  quantity = input()                  if (float(quantity) <= float(data[id]['quantity'])):                      data[id]['quantity'] = str(                          float(data[id]['quantity'])-float(quantity))                      quantity\_all.append(quantity)                      user\_data[user\_id][str(i+1+g)]['quantity'] = str(quantity)                      price.append(data[id]['price'])                      user\_data[user\_id][str(i+1+g)]['price'] = data[id]['price']                      user\_data[user\_id][str(                          i+1+g)]['Transaction ID'] = str(transaction\_id)                  else:                      print(                          "The Quantity You Have Asked is Quite High Than\                          That is Available in Stock")                      print(                          "Did you Want To buy According to The Quantity\                          Available in Stock then Enter '0' Else '1'\                          to skip This Product")                      key = int(input())                      if (key == 0):                          print("Enter Quantity of Product " +                                str(i+1)+" that you want to buy")                          quantity = intput()                          if (float(quantity) <= float(data[id]['quantity'])):                              data[id]['quantity'] = str(                                  float(data[id]['quantity'])-float(quantity))                              quantity\_all.append(quantity)                              user\_data[user\_id][str(                                  i+1)]['quantity'] = str(quantity)                              price.append(data[id]['price'])                              user\_data[user\_id][str(                                  i+1)]['price'] = data[id]['price']                              user\_data[user\_id][str(                                  i+1+g)]['Transaction ID'] = str(transaction\_id)                          else:                              print("Invalid Operation Got Repeated...!!!")                      elif (key == 1):                          continue                      else:                          print("Invalid Choice...!!!")              else:                  print("Invalid Product ID...!!!")          if(len(purchase\_no) != 0):              generate\_bill(user\_id, prod\_id, price, time\_date, purchase\_no,                            name, category, quantity\_all, transaction\_id)      else:          print("User ID Doesn't Exists...!!!")      js = json.dumps(data)      fd = open("data.json", 'w')      fd.write(js)      fd.close()      js = json.dumps(user\_data)      fd = open("user\_data.json", 'w')      fd.write(js)      fd.close()  while (1):      print("Choose Any One of The Following :- ")      print("1)Admin")      print("2)User")      print("3)Exit")      print("Enter Your Choice Here :- ")      n = int(input())      if (n == 1):          admin()      elif (n == 2):          user()      elif (n == 3):          break      else:          print("Invalid Choice...!!!") |

# 3.Display the Current weather information

Choose Any One of The Following :-

1)Admin

2)User

3)Exit

Enter Your Choice Here :-

2

======= Welcome to the User Inventory Management System ====

1)Display All Products With Details

2)Display Specific Product With Details

3)Display All Purchase Bills

4)Buy The Product

5)Exit

Enter Your Choice :-

4

Enter Your User ID if You are Old Customer else press '0' To New User ID :-

0

Enter Number of Products You Want To Buy :-

2

Enter Data As Follows :-

Enter Product ID of Product 1 that you want to buy

1002

For Product lotion Available Quantity is :- 100

Enter Quantity of Product 1 that you want to buy

1

Enter Product ID of Product 2 that you want to buy

1002

For Product lotion Available Quantity is :- 99.0

Enter Quantity of Product 2 that you want to buy

2

========= Bill ========

#######################

User ID :- 1000

#################

-----------------------------------------

Purchase number 1

Purchase Time :- Wed Jul 17 15:33:31 2024

Product ID :- 1002

Name Of Product :- lotion

Category Of Product :- beauty & personal

Price of Product per Item :- 250

Purchase Quantity :- 1

-----------------------------------

-----------------------------------------

Purchase number 2

Purchase Time :- Wed Jul 17 15:33:47 2024

Product ID :- 1002

Name Of Product :- lotion

Category Of Product :- beauty & personal

Price of Product per Item :- 250

Purchase Quantity :- 2

-----------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Total Payable Bill :- 750.0 Transaction ID :- XBKHA864B2

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1)Display All Products With Details

2)Display Specific Product With Details

3)Display All Purchase Bills

4)Buy The Product

5)Exit

Enter Your Choice :-

5

**5.Documentation**

#### Inventory Management System Documentation

#### Table of Contents

#### [Introduction](https://chatgpt.com/#introduction)

#### [Features](https://chatgpt.com/#features)

#### [Installation](https://chatgpt.com/#installation)

#### [Usage](https://chatgpt.com/#usage)

#### [Conclusion](https://chatgpt.com/#conclusion)

### Introduction

The Inventory Management System is designed to help businesses manage their inventory effectively. It allows users to track inventory levels, manage stock, and generate reports

**Features**

#### Add, update, and delete inventory items

#### Track inventory levels

#### Generate inventory reports

#### User authentication

#### Search functionality

### Installation

* **Clone the repository.**
* **Create a virtual environment and activate it.**
* **Install the required dependencies.**
* **Set up the database.**
* **Run the application**

**Usage**

* **Starting the Application:** After running **python app.py**, the application will start, and you can access it through your web browser at **http://localhost:5000**.
* **User Authentication:**
  + Register a new user or log in with existing credentials.
  + Only authenticated users can manage the inventory.
* **Managing Inventory:**
  + **Add Item:** Navigate to the "Add Item" page to add new inventory items.
  + **Update Item:** Edit item details from the inventory list.
  + **Delete Item:** Remove items from the inventory.
  + **View Inventory:** View all inventory items, including their details and current stock levels.
* **Generating Reports:**
  + Navigate to the "Reports" section to generate and view inventory reports.

**Conclusion**

* This Inventory Management System provides a simple yet effective way to manage and track inventory. It can be extended with additional features like advanced reporting, barcode scanning, and integration with other business systems.