तत् त्वं पूषन् अपावृणु

केन्द्रीय विद्यालय संगठन

KENDRIYA VIDYALAYA

SECTOR-3, ROHINI, NEW DELHI-110085

Computer Science Project

" Logistic Management System "

Submitted by

NAME: Tushar Singh Bisht,
Aaradhya Aamera,
Krishna Rawat

Class: XII-B

Roll No.: 41

CERTIFICATE

This is to certify that **Tushar Singh Bisht, Aaradhya Aamera, Krishna Rawat** of class **XII-B** has worked successfully under the supervision of **Mrs. Neelima Singh** during academic year 2023-24 on the project **"Logisitic Management System"** as per the guidelines issued by Central Board of Secondary Education (CBSE).

Sign of Subject Teacher

Sign of external examiner

ACKNOWLEDGEMENT

We are thankful to our CS teacher Mrs. Neelima Singh who helped and guided us while making this project. Without her guidance my project would have been incomplete and imperfect.

The guidance and support received from all the members who contributed and who contributed and who are contributing to this project, was vital for the success of the project.

INTRODUCTION

"Logistic management system" is a project which can be used to keep track of products like their shipment status, whether the product is been delivered or not. which has a wide application in the E-Commerce sector, because for a company it's a important task to keep tracks of all the orders of their user.

OBJECTIVES

Objective of this project is to make use of python and programming skills and use it to make something which can have some practical application in real world and can be used by people.

This problem of managing products data is usually faced by small scale businesses like shopkeepers, they generally keep manual records of what is delivered to whom in a notebook which can become easier and more manageble with help of this simple python program.

ABOUT PROJECT

➔ PROGRAMMING LANGUAGE – PYTHON

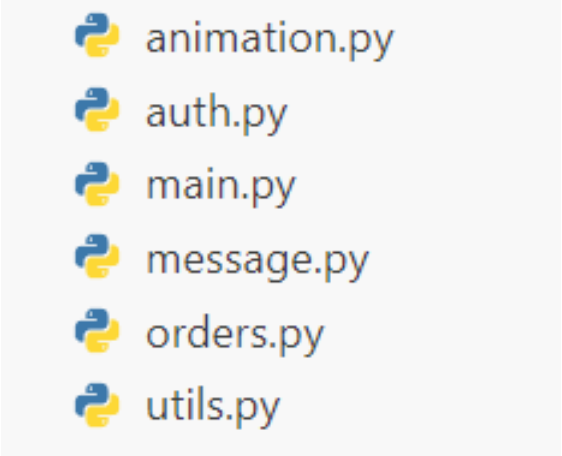
THINGS WHICH CONTRIBUE A MAJOR PART OF CODE IS BINARY FILES MANAGEMENT DONE USING PYTHON.

DATA IS STORED IN BINARY FILES IN .dat FORMAT

➔ MAJOR FETURES OF PROJECT –

- ADDED USERID, PASSWORD BASED AUTHENTICATION.
- USER FRIENDLY INTERFACE
- EASY TO USE COMMANDS

➔ FILES –



SOURCE CODE

animation.py

just contains some function to animate text in terminal

```
import time,sys, random

def animate(txt):
    for i in txt:
        sys.stdout.write(i)
        sys.stdout.flush()
        time.sleep(random.randint(1, 10)/100)
    print("")


def loading():
    x = 0.01
    for i in ".......":
        sys.stdout.write(i)
        sys.stdout.flush()
        x = x + 0.05
        time.sleep(x)
    print("")
```

messages.py : contains code to print colored text

```
from animation import animate

BLUE = '\033[94m'
GREEN = '\033[92m'
YELLOW = '\033[93m'
RED = '\033[91m'
ENDC = '\033[0m'
BOLD = '\033[1m'
PINK = '\033[95m'
CYAN = '\033[96m'


def blue(s):
    animate(BLUE+s+ENDC)

def cyan(s):
    animate(CYAN+s+ENDC)

def green(s):
    animate(GREEN+s+ENDC)

def yellow(s):
    animate(YELLOW+s+ENDC)


def red(s):
    animate(RED+s+ENDC)

def bold(s):
    animate(BOLD+s+ENDC)


def pink(s):
    animate(PINK+s+ENDC)

def color(s, col, anm=True):
    COL = ""
    if col == "blue":
        COL=BLUE
    elif col == "cyan":
        COL=CYAN
```

```
    elif col == "green":
        COL=GREEN
    elif col == "yellow":
        COL=YELLOW
    elif col == "red":
        COL=RED
    elif col == "pink":
        COL=PINK
    else:
        if anm:
            animate(s)
        else:
            print(s)
        return None

    if anm:
        animate(COL+s+ENDC)
    else:
        print(COL+s+ENDC)
```

utils.py

some useful functions

```python
import random
from string import ascii_letters, digits
import message

def getID():
    s = ascii_letters + digits
    ID = ""
    for i in range(20):
        rand_char = s[random.randint(0, len(s)-1)]
        ID += rand_char

    return ID


def inpval(isInt=False, promt="->"):
    while True:
        i = input(promt)
        if i.strip() == "":
            message.red("Please enter something")
            continue
        else:
            if isInt:
                try:
                    i = int(i)
                except ValueError:
                    message.red("Invalid input x_x")
                    continue

            break
    return i
```

auth.py

functions realated to authentication

```
import message
import os
from animation import loading
import pickle
from string import punctuation
from pwinput import pwinput as getpass

def login():
    message.blue("\nEnter USERID: ")
    userID = input("-> ")

    dr = "./data/" + userID
    if os.path.exists(dr):
        f = open(dr + "/info.dat", "rb")
        data = pickle.load(f)
        f.close()
        if data["userID"] == userID:
            message.blue("Enter PASSWORD: ")
            pw = getpass("-> ")
            if pw == data["password"]:
                loading()
                message.green("Logged in Successfully!")
                return ["SUCCESS", data]
            else:
                loading()
                message.red("Oops.. looks like its a wrong password")
                message.cyan("No worries you can try again..")
                return ["FAILURE"]
    else:
        loading()
        message.red("User with this User ID do not exist")
        message.cyan("Try out login with a differnt ID")
        return ["FAILURE"]


def register():
    while True:
        name = ""
        userID = ""
        password = ""

        message.blue("\nEnter NAME :- ")
        name = input("-> ")

        if name.strip() == "":
            message.red("Invalid name")
```

```
            message.blue("Register Again!\n\n")
            continue

        message.blue("Enter USERID:- ")
        message.color("No special characters are allowed!", "yellow", False)
        userID = input("-> ")

        if userID.strip() == "" or any(p in userID for p in punctuation):
            message.red("Invalid User ID")
            message.blue("Register Again!\n\n")
            continue

        if os.path.exists("./data/" + userID):
            message.red("User with similar ID exists")
            message.blue("Try to login please or use a different User ID")
            return ["FAILURE"]

        message.blue("Enter PASSWORD (Must have 8 char atleast) :- ")
        while True:
            password = getpass("-> ")

            if password.strip() == "":
                message.red("Password can't be empty")
                message.blue("Enter password Again!")
                continue

            if len(password) < 8:
                message.red("Password must have 8 character atleast")
                message.blue("Enter password Again!")
                continue

            message.yellow("Please confirm your password -")
            confirm_pw = getpass("->")

            if confirm_pw != password:
                message.red("Passwords don't match")
                message.blue("Enter password Again!")
                continue
            message.cyan("Password confirmed!\n")
            break
        break

    loading()
    dr = "./data/" + userID
    os.mkdir(dr)
    f = open(dr + "/info.dat", "ab")
    data = { "name": name, "userID": userID, "password": password}
    pickle.dump(data,f)
    f.close()
    message.green("Successfully created a new user account!")
    return ["SUCCESS", data]
```

orders.py

functions related to orders

```python
import pickle
import message
from utils import getID, inpval
import datetime
from animation import loading


def addOrder(userID):
    try:
        f = open("./data/" + userID + "/orders.dat", "ab")
        message.pink("\nFill order details")
        message.blue("\nOrder Name: ")
        name = inpval()
        message.blue("\nOrdered by: ")
        orderby = inpval()
        message.blue("\nDelivery address: ")
        address = inpval()
        message.blue("\nOrder Cost: ")
        cost = inpval(True)
        if name == "" or orderby == "" or address == "":
            raise ValueError
        data = {
            "orderID": getID(),
            "orderBy": orderby,
            "address": address,
            "ORDERED_at": str(datetime.date.today()),
            "status": "ORDERED",
            "name": name,
            "cost": cost,
            "SHIPPED_at": "NA",
            "DELIVERED_at": "NA",
            "CANCELLED_at": "NA",
            "FAILED_at": "NA",
        }
        pickle.dump(data, f)
        f.close()
        loading()
        message.green("\nOrder successfully added :-).\n")
    except ValueError:
        message.red("Invalid input...X_X\n")

def getOrdersData(userID):
    try:
        f = open("./data/" + userID + "/orders.dat", "rb")
        data = []
        try:
            while True:
                data.append(pickle.load(f))
        except EOFError:
            pass
        f.close()
```

```
        if len(data) == 0:
            raise FileNotFoundError
        return data
    except FileNotFoundError:
        message.red("You don't have any data as of now first add it!")
        return []

def getOrderDetails(order, complete=False):
    print("-> ORDER ID =>", order["orderID"])
    print("-> ORDER NAME =>", order["name"])
    print("-> ORDERED BY =>", order["orderBy"])
    print("-> ADDRESS =>", order["address"])
    print("-> COST =>", order["cost"])
    print("-> STATUS =>", order["status"])
    print("-> ORDERED AT =>", order["ORDERED_at"])
    if complete:
        print("-> SHIPPED AT =>", order["SHIPPED_at"])
        print("-> DELIVERED AT =>", order["DELIVERED_at"])
        print("-> FAILED AT =>", order["FAILED_at"])

def viewAllOrders(userID):
    data = getOrdersData(userID)
    if len(data) == 0:
        return None
    loading()
    print("\n")
    print("_____________________________________________________________________________")
    print("{:<5} {:<25} {:<25} {:<10} {:<10} {:<20}".format("SNO.", "ORDERID", "NAME",
"COST", "STATUS", "DATE OF ORDER"))
    k = 0
    for d in data:
        k += 1
        txt = "{:<5} {:<25} {:<25} {:<10} {:<10} {:<20}".format(k, d["orderID"],
d["name"][0:25] + "..", d["cost"], d["status"],d["ORDERED_at"])

        if d["status"] == "ORDERED":
            print(txt)
        elif d["status"] == "SHIPPED":
            message.color(txt, "cyan", False)
        elif d["status"] == "CANCELLED":
            message.color(txt, "yellow", False)
        elif d["status"] == "DELIVERED":
            message.color(txt, "green", False)
        elif d["status"] == "FAILED":
            message.color(txt, "red", False)
    print("_____________________________________________________________________________")
    print("\n")
    message.pink("Total orders till now => " + str(len(data)))


def viewOrder(userID, status, color):
    data = getOrdersData(userID)
    if len(data) == 0:
        return None
    loading()
    print("\n")
```

```
    print("_____________________________________________________________________________")
    message.color("{:<5} {:<25} {:<25} {:<10} {:<10} {:<20}".format("SNo.", "ORDERID",
"NAME", "COST", "STATUS", status + " AT"), color, False,)
    k = 0
    for d in data:
        if d["status"] == status:
            k += 1
            print("{:<5} {:<25} {:<25} {:<10} {:<10} {:<20}".format(k, d["orderID"],
d["name"][0:25] + "..", d["cost"], d["status"],d[status + "_at"]))
    print("_____________________________________________________________________________")
    print("\n")
    message.color("Total orders " + status + " => " + str(k), color)


def deleteOrder(userID):
    message.blue("Enter order ID to delete order: ")
    ID = input("->")
    data = getOrdersData(userID)
    data_new = []
    found = False
    for i in data:
        if i["orderID"] == ID:
            found = True
        else:
            data_new.append(i)
    loading()
    if found:
        f = open("./data/" + userID + "/orders.dat", "wb")
        for i in data_new:
            pickle.dump(i, f)
        message.green("Deleted order with ID - " + ID)

    else:
        message.red("Can't find the order with id provided ;-;")

def updateOrderDetails(userID):
    message.blue("\nEnter order ID to update: ")
    ID = input("->")
    data = getOrdersData(userID)
    data_new = []
    order = None
    found = False
    updated = True
    for i in data:
        if i["orderID"] == ID:
            found = True
            message.pink("\nOrder details -\n")
            getOrderDetails(i)
            message.yellow("\nWhat do you want to update ? enter code to select")
            while True:
                print("\n1 = Edit name")
                print("2 = Edit cost")
                print("3 = Edit ordered by")
                print("4 = Edit address")
                print("5 = Edit order status")
                print("6 = No updates")
                c = inpval(True, "Enter Code: ")
```

```
            if c == 1:
                message.blue("\nEnter New Name: ")
                i["name"] = inpval()
                break
            elif c == 2:
                message.blue("\nEnter New Cost: ")
                i["cost"] = inpval(True)
                break
            elif c == 3:
                message.blue("\nEnter new Reciever: ")
                i["orderBy"] = inpval()
                break
            elif c == 4:
                message.blue("\nEnter new address: ")
                i["address"] = inpval()
                break
            elif c == 5:
                message.yellow("\nUse follwing codes to set order status")
                print("\n1 => ORDERED")
                print("2 => SHIPPED")
                print("3 => DELIVERED")
                print("4 => CANCELLED")
                print("5 => FAILED\n")
                while True:
                    message.color("CURRENT STATUS: " + i["status"], "cyan", False)
                    x = inpval(True, "\nEnter: ")
                    if x == 1:
                        i["status"] = "ORDERED"
                        break
                    elif x == 2:
                        i["status"] = "SHIPPED"
                        i["SHIPPED_at"] = str(datetime.date.today())
                        break
                    elif x == 3:
                        i["status"] = "DELIVERED"
                        i["DELIVERED_at"] = str(datetime.date.today())
                        break
                    elif x == 4:
                        i["status"] = "CANCELLED"
                        i["CANCELLED_at"] = str(datetime.date.today())
                        break
                    elif x == 5:
                        i["status"] = "FAILED"
                        i["FAILED_at"] = str(datetime.date.today())
                        break
                    else:
                        message.red("\nEnter a valid status code (1 to 5)")
                break
            elif c == 6:
                message.green("\nSure, if you want no updates!")
                updated = False
                break
            else:
                message.red("\nEnter valid code (1 to 6)!")
        order = i
        data_new.append(i)
    else:
```

```
            data_new.append(i)
    loading()

    if found and updated and order != None:
        f = open("./data/" + userID + "/orders.dat", "wb")
        for i in data_new:
            pickle.dump(i, f)
        message.green("Updated order with ID - " + ID)
        message.pink("\nOrder details -\n")
        getOrderDetails(order)
    elif found:
        message.green("No updates are done to order with ID - " + ID)
    else:
        message.red("Can't find any order with id provided ;-;")

def searchOrder(userID):
    message.yellow("\nEnter order ID to search: ")
    ID = input("->")
    data = getOrdersData(userID)
    order = None
    found = False
    for i in data:
        if i["orderID"] == ID:
            found = True
            order = i
    loading()
    if found and order != None:
        message.green("\nSuccessfully Found order the with ID - " + ID)
        message.green("Order Complete Details are - ")
        getOrderDetails(order, True)
    else:
        message.red("Can't find any order with id provided ;-;")
```

main.py

main file to be executed

```
import message
from animation import loading, animate
import orders
import auth
import os

if os.path.exists("./data") == False:
    os.mkdir("./data")
message.blue("\n\nHello, Welcome to THE LOGISTIC MANAGEMENT SYSTEM!")
user = None
a_ = True
while True:
    message.color("\nEnter 1 to CREATE NEW ACCOUNT and 2 to LOGIN", "pink", a_)
    a_= False
    try:
        inp = int(input("-> "))
        if inp == 1:
            d = auth.register()
            if d[0] == "FAILURE":
                continue
            if d[0] == "SUCCESS":
                user = d[1]
                break
        elif inp == 2:
            d = auth.login()
            if d[0] == "FAILURE":
                continue
            if d[0] == "SUCCESS":
                user = d[1]
                break
        else:
            message.red("Invalid input")
            continue
    except ValueError:
        message.red("Sorry, Please enter either 1 or 2")
        continue
if user == None:
    exit()

loading()

animate("\nWelcome " + user["name"] +", feel free to use all the services\n\n")
message.blue("\nENTER FOLLOWING CODES TO GET YOUR WORK DONE -")
loading()
while True:
    print("____________________________________________________")
    print("\n1. ADD ORDER")
    print("2. UPDATE ORDER")
```

```
        print("3. DELETE ORDER")
        print("4. VIEW ALL ORDERS")
        print("5. VIEW IN PROGRESS ORDERS")
        print("6. VIEW SHIPPED ORDER")
        print("7. VIEW DELIVERED ORDER")
        print("8. VIEW CANCELLED ORDER")
        print("9. VIEW FAILED ORDER")
        print("10. SEARCH FOR ORDER")
        print("11. EXIT.")
        message.color("\nWhat you want to do next?", "blue", False)
        message.blue("ENTER CHOICE :")
        try:
            code = int(input("-> "))
        except ValueError:
            message.red("Invalid input X_X")
            continue
        if code == 1:
            orders.addOrder(user["userID"],)
            loading()
        elif code == 2:
            orders.updateOrderDetails(user["userID"])
            loading()
        elif code == 3:
            orders.deleteOrder(user["userID"])
            loading()
        elif code == 4:
            orders.viewAllOrders(user["userID"])
            loading()
        elif code == 5:
            orders.viewOrder(user["userID"],"ORDERED", "pink")
            loading()
        elif code == 6:
            orders.viewOrder(user["userID"],"SHIPPED", "cyan")
            loading()
        elif code == 7:
            orders.viewOrder(user["userID"],"DELIVERED", "green")
            loading()
        elif code == 8:
            orders.viewOrder(user["userID"],"CANCELLED", "yellow")
            loading()
        elif code == 9:
            orders.viewOrder(user["userID"],"FAILED", "red")
            loading()
        elif code == 10:
            orders.searchOrder(user["userID"])
            loading()
        elif code == 11:
            message.green(
    "Exited successfully... ^_^ have a nice day " + user["name"] + " !")
            break

        else:
            message.red("Please enter a valid code -_-")
            loading()
```

OUTPUT

Creating a new account

```
D:\projects\cbse2023>python main.py


Hello, Welcome to THE LOGISTIC MANAGEMENT SYSTEM by tushar_coder!

Enter 1 to CREATE NEW ACCOUNT and 2 to LOGIN
-> 1

Enter NAME :-
-> Tushar
Enter USERID:-
No special characters are allowed!
-> tushar
Enter PASSWORD (Must have 8 char atleast) :-
-> ********
Please confirm your password -
->********
Password confirmed!

........
Successfully created a new user account!
........
```

Loggin in to an account

```
D:\projects\cbse2023>python main.py


Hello, Welcome to THE LOGISTIC MANAGEMENT SYSTEM by tushar_coder!

Enter 1 to CREATE NEW ACCOUNT and 2 to LOGIN
-> 2

Enter USERID:
-> tushar
Enter PASSWORD:
-> ********
........
Logged in Successfully!
```

All Features

```
Welcome Tushar, feel free to use all the services



ENTER FOLLOWING CODES TO GET YOUR WORK DONE -
........
--------------------------------------------------

1. ADD ORDER
2. UPDATE ORDER
3. DELETE ORDER
4. VIEW ALL ORDERS
5. VIEW IN PROGRESS ORDERS
6. VIEW SHIPPED ORDER
7. VIEW DELIVERED ORDER
8. VIEW CANCELLED ORDER
9. VIEW FAILED ORDER
10. SEARCH FOR ORDER
11. EXIT.

What you want to do next?
ENTER CHOICE :
-> |
```

Adding a order

```
What you want to do next?
ENTER CHOICE :
-> 1

Fill order details

Order Name:
->camera

Ordered by:
->marry

Delivery address:
->ABC

Order Cost:
->25000
........

Order successfully added :-).
```

Viewing order

```
What you want to do next?
ENTER CHOICE :
-> 4
........


--------------------------------------------------------------------------------------------
SNO.  ORDERID                    NAME                      COST       STATUS     DATE OF ORDER
1     35CPG6ogjTGCY6ec0dd0       Laptop..                  500        ORDERED    2024-01-16
2     3vJuyVGCtNJ9cD4cPQmQ       camera..                  25000      ORDERED    2024-01-16
--------------------------------------------------------------------------------------------


Total orders till now => 2
........
```

Deleting order

```
What you want to do next?
ENTER CHOICE :
-> 4
........


--------------------------------------------------------------------------------------------
SNO.  ORDERID                    NAME                      COST       STATUS     DATE OF ORDER
1     35CPG6ogjTGCY6ec0dd0       Laptop..                  500        ORDERED    2024-01-16
2     3vJuyVGCtNJ9cD4cPQmQ       camera..                  25000      ORDERED    2024-01-16
--------------------------------------------------------------------------------------------


Total orders till now => 2
........
```

```
What you want to do next?
ENTER CHOICE :
-> 3
Enter order ID to delete order:
->3vJuyVGCtNJ9cD4cPQmQ
........
Deleted order with ID - 3vJuyVGCtNJ9cD4cPQmQ
........
```

```
What you want to do next?
ENTER CHOICE :
-> 4
........


--------------------------------------------------------------------------------------------
SNO.  ORDERID                    NAME                      COST       STATUS     DATE OF ORDER
1     35CPG6ogjTGCY6ec0dd0       Laptop..                  500        ORDERED    2024-01-16
--------------------------------------------------------------------------------------------


Total orders till now => 1
```

Updating order

```
What you want to do next?
ENTER CHOICE :
-> 2

Enter order ID to update:
->35CPG6ogjTGCY6ec0dd0

Order details -

-> ORDER ID => 35CPG6ogjTGCY6ec0dd0
-> ORDER NAME => Laptop
-> ORDERED BY => Joe
-> ADDRESS => XYZ
-> COST => 500
-> STATUS => ORDERED
-> ORDERED AT => 2024-01-16

What do you want to update ? enter code to select

1 = Edit name
2 = Edit cost
3 = Edit ordered by
4 = Edit address
5 = Edit order status
6 = No updates
Enter Code: 1

Enter New Name:
->Laptop Asus
........
Updated order with ID - 35CPG6ogjTGCY6ec0dd0

Order details -

-> ORDER ID => 35CPG6ogjTGCY6ec0dd0
-> ORDER NAME => Laptop Asus
-> ORDERED BY => Joe
-> ADDRESS => XYZ
-> COST => 500
-> STATUS => ORDERED
-> ORDERED AT => 2024-01-16
```

Searching for order

```
What you want to do next?
ENTER CHOICE :
-> 10

Enter order ID to search:
->35CPG6ogjTGCY6ec0dd0
........

Successfully Found order the with ID - 35CPG6ogjTGCY6ec0dd0
Order Complete Details are -
-> ORDER ID => 35CPG6ogjTGCY6ec0dd0
-> ORDER NAME => Laptop Asus
-> ORDERED BY => Joe
-> ADDRESS => XYZ
-> COST => 500
-> STATUS => ORDERED
-> ORDERED AT => 2024-01-16
-> SHIPPED AT => NA
-> DELIVERED AT => NA
-> FAILED AT => NA
```

Changing order status

```
What you want to do next?
ENTER CHOICE :
-> 2

Enter order ID to update:
->35CPG6ogjTGCY6ec0dd0

Order details -

-> ORDER ID => 35CPG6ogjTGCY6ec0dd0
-> ORDER NAME => Laptop Asus
-> ORDERED BY => Joe
-> ADDRESS => XYZ
-> COST => 500
-> STATUS => ORDERED
-> ORDERED AT => 2024-01-16

What do you want to update ? enter code to select

1 = Edit name
2 = Edit cost
3 = Edit ordered by
4 = Edit address
5 = Edit order status
6 = No updates
Enter Code: 5

Use follwing codes to set order status

1 => ORDERED
2 => SHIPPED
3 => DELIVERED
4 => CANCELLED
5 => FAILED

CURRENT STATUS: ORDERED

Enter: 2
........
Updated order with ID - 35CPG6ogjTGCY6ec0dd0

Order details -

-> ORDER ID => 35CPG6ogjTGCY6ec0dd0
-> ORDER NAME => Laptop Asus
-> ORDERED BY => Joe
-> ADDRESS => XYZ
-> COST => 500
-> STATUS => SHIPPED
-> ORDERED AT => 2024-01-16
```

```
What you want to do next?
ENTER CHOICE :
-> 4
........


-----------------------------------------------------------------------------------------------
SNO.  ORDERID                    NAME                        COST       STATUS     DATE OF ORDER
1     35CPG6ogjTGCY6ec0dd0       Laptop Asus..               500        SHIPPED    2024-01-16
-----------------------------------------------------------------------------------------------
```