**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

****

**SHOPIFY**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ACKNOWLEDGEMENT**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

This Project is the result of maximum effort put in by:

->AARON SHENNY

->ARDITH R

->ASWIN ARAVIND

->RISHABH THAMBI

->AARON JIMMY

It is done as per the latest CBSE scheme and is written in PYTHON programming language. It also depicts the imagination and level of creativity of these youngsters. The topic we felt right for our project is **SHOPIFY – AN ONLINE GROCERY SHOP**, so we went forward with it. We appreciate each other’s contribution and are grateful to our Computer Science teacher **Mrs. Reeba John** who taught us the python programming language for the past year and with whose guidance we were able to make this project a complete success and our Principal **Dr. Sheela Seth,** for giving us a golden opportunity to do this project.

**INTRODUCTION**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

PYTHON is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. CBSE has introduced this to the curriculum in the view of this use in the later parts of life. With the rapid development of everyday software and manipulation of old ones for better and efficient management of daily tasks, life is made easier. The bank management system is an application for maintaining a person’s bank account. In this project we tried to replicate the working of banking systems and cover the basic functionality of a bank account management system. This python program is developed with simple function which enables the user of the program to create a bank account with necessary details. To sum up, the project teaches the proper use of file handling and working with multiple modes thus serving as a good reference project.

**PYTHON CODE**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**import getpass**

**import time**

**user\_buy =  {}**

**database = {**

**'user' : { 'aaronshenny': {   'name' : 'Aaron Shenny',**

**'password' : '123'**

**},**

**'user': {    'name' : 'Guest',**

**'password' :'root'**

**},**

**'aswinaravind27': {       'name' : 'Aswin Aravind',**

**'password':'aswi'**

**}**

**},**

**'vegetables': { 'tomato' : { 'name' : 'Tomato',**

**'price' : '48RS',**

**'stock' : 10**

**},**

**'onion': {         'name':'Onion',**

**'price':'79RS',**

**'stock':15**

**},**

**'greenchilli': {     'name':'Green chilli',**

**'price':'46RS',**

**'stock':12**

**},**

**'beetroot': {             'name':'Beetroot',**

**'price':'34RS',**

**'stock':14**

**},**

**'potato': {             'name':'Potato',**

**'price':'40RS',**

**'stock':16**

**},**

**'cabbage': {            'name':'Cabbage',**

**'price':'25RS',**

**'stock': 13**

**},**

**'carrot':{     'name':'Carrot',**

**'price':'39RS',**

**'stock':17**

**},**

**'corn': {           'name':'Corn',**

**'price':'35RS',**

**'stock':19**

**},**

**'coconut': {            'name':'Coconut',**

**'price':'37RS',**

**'stock':16**

**},**

**'ginger': {            'name':'Ginger',**

**'price':'111RS',**

**'stock':20**

**},**

**'elephantyam': {     'name':'Elephant Yam',**

**'price':'34RS',**

**'stock':15**

**},**

**'brinjal': {            'name':'Brinjal',**

**'price':'33RS',**

**'stock':18**

**}**

**},**

**'fruits':{         'apple':{       'name':'Apple',**

**'price':'190Rs',**

**'stock':21**

**},**

**'banana':{           'name':'Banana',**

**'price':'55Rs',**

**'stock': 24**

**},**

**'orange':{             'name':'Orange',**

**'price':'65Rs',**

**'stock':27**

**},**

**'mango':{           'name':'Mango',**

**'price':'89Rs',**

**'stock':13**

**},**

**'watermelon':{       'name':'Watermelon',**

**'price':'28Rs',**

**'stock':28**

**},**

**'grapes': {       'name':'Grapes',**

**'price':'150Rs',**

**'stock':12**

**},**

**'papaya': {         'name':'Papaya',**

**'price':'35Rs',**

**'stock':19**

**},**

**'guava': {             'name':'Guava',**

**'price':'89Rs',**

**'stock':11**

**},**

**'pineapple': {      'name':'Pineapple',**

**'price':'35Rs',**

**'stock':27**

**},**

**'pomegranate': {   'name':'Pomegranate',**

**'price':'189Rs',**

**'stock':30**

**},**

**'avocado': {             'name':'Avocado',**

**'price':'260Rs',**

**‘stock':32**

**},**

**'dragonfruit': {           'name':'Dragonfruit',**

**'price':'299Rs',**

**'stock':31**

**}**

**}**

**}**

**# Function to create a new user account**

**def create\_user(name):**

**print('SIGN-UP')**

**print('NOTE : Sorry, Due to the limited knowlegde, Now creating account will be deleted**

**after the program closes. Use the default username and password...')**

**print('Creating a user account...')**

**username = input('Username : ')**

**if username in database['user']:  #This will check if the user had already created account**

**print('Same user has been found in our database. Please login ...')**

**else:**

**try:**

**password = getpass.getpass(prompt = 'Create Your Account Password : ')**

**except Exception as Error:**

**print('Error : ', Error)**

**try:**

**database['user'][username] = {**

**'name': name,                            #Adds Name and password into the database**

**'password': password**

**}**

**except Exception as Error:**

**print('Error : ', Error)**

**print('Account created successfully...')**

**# Function for user sign-in**

**def sign\_in():**

**while True:**

**print()**

**print()**

**print('\t\t\tLOGIN')**

**print()**

**username = input('Username : ')**

**if username in database['user']:         #Checking given Username is matching with**

**# usernames in databse**

**password1 = getpass.getpass(prompt = 'Password : ')**

**if password1 == database['user'][username]['password']:    #Checking if the given**

**# password is correct with database**

**time.sleep(1)**

**print('Account logined..')**

**print()**

**print('Welcome',database['user'][username]['name'])**

**username1 = username**

**login = True                                           #Intializing the varible as True**

**return username,login                                  #Returning username and login variable**

**break**

**else:**

**login = False                                          #Intializing the varible as True**

**print('Incorrect Password...')**

**login\_checker(login)**

**return username,login                                  #Returning username and login variable**

**else:**

**time.sleep(1)**

**print('Account not Found')**

**time.sleep(1)                                              # If the account didnt found on the database**

**# then create\_user() is called**

**print('Creating an account...')**

**time.sleep(1)**

**print()**

**print()**

**name  = input('Full name : ')**

**create\_user(name)**

**# Function for purchasing items**

**def buy(l,username):**

**brougth\_items = []**

**while True:**

**print()**

**item = input('Enter an item : ').lower()                      #User enters the product they need**

**if item == 'exit' or item == '0':                               #Exiting the loop**

**break**

**elif item in brougth\_items:**

**print()                                                  #Checking the cart if the user had already brought**

**print('Item is already in the cart!!')**

**for i in l:**

**if item.title() == i[0]:**

**print(f'Product : {i[0]}')**

**print(f'Quantity : {i[1]}')**

**print()**

**change = input('Do you want to change the quantity  ? : [yes/no] ')**

**print()                    #Asking the user if they want to change the quantity**

**if change == 'yes':**

**for i in l:**

**if item.title() == i[0]:**

**if i[0].lower in database['vegetables']:**

**product,quantity = i   #Unpacking the tuple to change**

**quantity = float(input(f'How much kilo you need for**

**{database["vegetables"][item]["name"]} : ')) #Asking the change**

**t = product,quantity   #Packing the tuple**

**l.remove(i)            #Removing the existing tuple**

**l.append(t)            #Adding the new tuple into list**

**else:**

**product,quantity = i   #Unpacking the tuple to change**

**quantity = float(input(f'How much kilo you need for**

**{database["fruits"][item]["name"]} : ')) #Asking the change**

**t = product,quantity   #Packing the tuple**

**l.remove(i)            #Removing the existing tuple**

**l.append(t)**

**else:**

**for i in l:**

**if item in i[0]:**

**print()**

**print('Item is already added')**

**else:**

**try:**

**if item.lower() in database['vegetables'] or database['fruits']:          #Checking the**

**#product is in database**

**if item.lower() in database['vegetables'] :**

**qut = float(input(f'How much kilo you need for**

**{database["vegetables"][item]["name"]} : ')) #Asking the quantity**

**if qut < 0:**

**print('The quantity should be more than 0')                    #Checking the quantity is**

**#more than 0**

**buy(l,username)**

**break**

**if qut > database['vegetables'][item]['stock']:                    #Checking the given**

**#quantity is less than the stock**

**print(f'The quantity should be less than the TOTAL STOCK, Remaining Stock :**

**{database["vegetables"][item]["stock"]}')**

**buy(l,username)**

**break**

**brougth\_items.append(item)                                          #Adding the item into the cart**

**items = (database['vegetables'][item]['name'],qut)**

**l.append(items)**

**database['vegetables'][item]['stock'] = database['vegetables'][item]['stock'] - qut**

**print(f"Remaing Stocks = {database['vegetables'][item]['stock']}")**

**if database['vegetables'][item]['stock'] == 0:**

**del database['vegetables'][item]**

**elif item.lower() in database['fruits']:**

**qut = float(input(f'How much kilo you need for {database["fruits"][item]["name"]} : '))**

**if qut < 0:**

**print('The quantity should be more than 0')   #Checking the quantity is more than 0**

**buy(l,username)**

**break**

**if qut > database['fruits'][item]['stock']:    #Checking the given quantity is less than the stock**

**print(f'The quantity should be less than the TOTAL STOCK, Remaining Stock :**

**{database["fruits"][item]["stock"]}')**

**buy(l,username)**

**break**

**brougth\_items.append(item)                                          #Adding the item into the cart**

**items = (database['fruits'][item]['name'],qut)**

**l.append(items)**

**database['fruits'][item]['stock'] = database['fruits'][item]['stock'] - qut**

**print(f"Remaing Stocks = {database['fruits'][item]['stock']}")**

**if database['fruits'][item]['stock'] == 0:**

**del database['fruits'][item]**

**else:**

**print('item not found')**

**except ValueError:                                                    #Exception handling**

**print('Please enter an valid value...')**

**user\_buy[username] = l**

**return user\_buy**

**#Function for listing the items**

**def list1(database):**

**vegetable\_data = database.get('vegetables')**

**fruits\_data = database.get('fruits')**

**if not vegetable\_data:**

**print("No vegetable data found!")                #Checking if the database is empty or not**

**return**

**print()**

**print("------------------------------------------\t\t -----------------------------------------")**

**print("|   Vegetable   |     Price     | Stock  |\t\t|    Fruits     |     Price     | Stock  |")**

**print("------------------------------------------\t\t -----------------------------------------")**

**veg\_keys = list(database['vegetables'].keys())**

**fru\_keys = list(database['fruits'].keys())**

**for i, j in zip(veg\_keys,fru\_keys):**

**veg\_name = database['vegetables'][i]['name'].ljust(15)**

**veg\_price = database['vegetables'][i]['price'].ljust(15)**

**veg\_stock = str(database['vegetables'][i]['stock']).ljust(8)**

**fruit\_name = database['fruits'][j]['name'].ljust(15)**

**fruit\_price = database['fruits'][j]['price'].ljust(15)**

**fruit\_stock = str(database['fruits'][j]['stock']).ljust(8)**

**print(f'|{veg\_name}|{veg\_price}|{veg\_stock}|\t\t|{fruit\_name}|{fruit\_price}|{fruit\_stock}|')**

**print("------------------------------------------\t\t -----------------------------------------")**

**def recipt(username):                                                   #Function for printing the recipt**

**confirm =  input('Anything else ..? : ').lower()      #Asking the user if they want to buy anything else**

**if confirm == 'yes':**

**l =  user\_buy.get(username)**

**buy(l,username)**

**brougth\_items = user\_buy.get(username)**

**total\_amount = 0  # Initialize the total amount variable**

**print()**

**print('=' \* 70)**

**print('RECEIPT'.center(70))**

**print('=' \* 70)**

**time2 = time.asctime()                                              #Getting the current time**

**print('Name : ',database['user'][username]['name'],'\t\t\t','Date : ',time2)**

**print('=' \* 70)**

**print(''.ljust(8),'ITEM'.ljust(19),'RATE'.ljust(14),'QUANTITY'.ljust(17),'TOTAL')**

**print('=' \* 70)**

**for i in  brougth\_items:**

**product\_name, quantity = i**

**price\_per\_kilo = 0**

**# Check if the product is a vegetable or a fruit**

**if product\_name.lower() in database['vegetables']:**

**price\_per\_kilo = float(database['vegetables'][product\_name.lower()]['price'][:-2])**

**# Extract price per kilo**

**elif product\_name.lower() in database['fruits']:**

**price\_per\_kilo = float(database['fruits'][product\_name.lower()]['price'][:-2])**

**# Extract price per kilo**

**total\_price = price\_per\_kilo \* quantity**

**total\_amount += total\_price**

**print(product\_name.ljust(20) ,'|'.ljust(3),str(price\_per\_kilo).ljust(5), " RS/kg".ljust(8)**

**,'|'.ljust(4),str(quantity).ljust(3) ," kg".ljust(6) ,'|'.ljust(3) ,str(total\_price).ljust(5) ,**

**" RS".ljust(20))**

**print()**

**print('=' \* 70)**

**print('Total Amount :',total\_amount,' RS')**

**def login\_checker(login):**

**if login != True:**

**sign\_in()**

**# main body**

**print()**

**print('='\*55)**

**print()**

**print('  / \_\_\_\_| |  | |/ \_\_ \|  \_\_ \\_   \_|  \_\_\_\_\ \   / /')**

**print(' | (\_\_\_ | |\_\_| | |  | | |\_\_) || | | |\_\_   \ \\_/ / ')**

**print('  \\_\_\_ \|  \_\_  | |  | |  \_\_\_/ | | |  \_\_|   \   /  ')**

**print('  \_\_\_\_) | |  | | |\_\_| | |    \_| |\_| |       | |   ')**

**print(' |\_\_\_\_\_/|\_|  |\_|\\_\_\_\_/|\_|   |\_\_\_\_\_|\_|       |\_|   ')**

**print()**

**print('='\*55)**

**time.sleep(1)**

**n=0**

**def main():**

**username = None**

**while True:**

**time.sleep(1)**

**username,login = sign\_in()**

**time.sleep(1)**

**list1(database)**

**print()**

**buyacceot =  input('Wanna buy something from our store ...?? [yes/no] : ').lower()**

**#Asking the user if they want to buy anything..reconfirming**

**if buyacceot == 'yes':**

**time.sleep(1)**

**l = []**

**print()**

**print('NOTE : Type  "0" or "exit" after finishing adding the products')**

**buy(l,username)**

**if user\_buy[username] == []:**

**pass**

**else:**

**recipt(username)**

**else:**

**time.sleep(1)**

**print('Thank you for comming')**

**time.sleep(5)**

**break**

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

**main()**

**while True:**

**time.sleep(2)**

**choice = input("Press 'q' to quit or any other key to continue shopping...: ")**

**#Asking the user if they want to continue shopping**

**if choice.lower() == 'q':**

**print('Thank you for coming\nVisit again!!')**

**print("Exiting the program...")**

**break**

**else:**

**print('NEXT CUSTOMER PLEASE...')**

**time.sleep(2)**

**main()**