

*COMPUTER SCIENCE
PROJECT ON
“COMPANY
MANAGEMENT
SYSTEM (CMS)”*

Name : Priyanshu Jha

Class : XII-B

ADM NO. : 10618

School : MASD Public School, Panipat

Submitted to : Mr. Asis Pal

Roll No. :

ACKNOWLEDGEMENT

I would like to express a deep sense of thank and gratitude to our computer science teacher Mr. ASIS PAL for his coordination throughout the project. He has been the constant source of motivation and giving proper advice in this project.

I would also like to thank CBSE for allocating to us this project. While making of this project, real world problem have been tried to solve and simplify the complex process.

At last, I would like to extend my heartfelt thanks to my parents because without their help this project would not have been successful. Finally, I would like to thank my dear friends who have been with me all the time.

Name : Priyanshu Jha

Class : XII-B

CERTIFICATE

This is to certify that Priyanshu Jha of Class XII-B, M.A.S.D. Public School, Panipat has prepared this project entitled “**Company Management System (CMS)**”. This project is a result of his efforts and endeavour. This report is found worthy of acceptance as final project report for the subject Computer Science of Class XII for the academic year 2021-22. This report has been prepared under my guidance.

Mr. ASIS PAL

Department of Computer Science

M.A.S.D. Public School

CONTENTS

- Abstract
- Introduction
- Purpose
- Modules
- Pros & Cons
- Feasibility of Software (*CMS*)
- Flow Chart
- System Requirements
- Sample Source Code
- User Interface Snapshots
- Future Scope Of this Work
- Conclusion
- Bibliography

ABSTRACT

"Company Management System (CMS)" is complete package of Billing Management, Employee Database Management And Payroll Management System" is designed to make the existing manual system automatic with the help of computerised equipment and full-edged computer software, fulfilling their requirements, so that their valuable data and information can be stored for a longer period with easy access and manipulation of the same. The required software is easily available and easy to work with. This Offline application can maintain and view computerised records without getting redundant entries. The project provide advance and organised way to manage user data, employees, Billing, Sales , Purchase for good performance and provide better services for the client. This Package also include Game Module to engage user to work on this software and create interest for using the software.

INTRODUCTION

The project ""Company Management System (CMS)" has been developed to overcome the problems faced in the practicing of manual system. This software is built to eliminate and in some cases reduce the hardships faced by the existing system. Moreover this system is designed for particular need of the company to carry out its operations in a smooth and effective manner.

This offline application is reduced as much as possible to avoid errors while entering data. It also provides error message while entering invalid data. It is user-friendly as no formal knowledge is required to use the system.

Human resource challenges are faced by every organization which has to be overcome by the organization. Every organization has different employee management and payroll management needs.

Further various organization still uses manual ways to create invoice for sales and purchase which is very prone to human error.

Therefore I have design exclusive Company Management System that are adapted to the organization's Managerial Requirements.

PURPOSE

The purpose of this document is to describe the functionality and specifications of the design of a Offline application for Managing Employees and their payroll and managing Sales & Purchase, Stock, Inventory of the company. The expected audiences of this document are the developers and the user who are self employed. Now with the help of this system the user has the information on his finger tips and can easily prepare a good record based on their requirements. Finally, we can say that this system will not only automate the process but save the valuable time of the manager or the user, which can be well utilized by his institute. This will be an additional advantage and management of power based on their free time from his normal duty.

MODULES

ADMIN

The Admin gets logged in by valid username and password. Admin can add new Employee, Create Invoice, Update Inventory, add new Pay Grade for the employees. Admin can set attendance and overtime worked by an employee in a department with specific pay grade. The Admin can generate an automated monthly wage sheet for his employees. The admin can view all the past records of any recorded employees and invoices. Admin has also option to play KBC which is also included in the package.

PROS AND CONS

PROS :

- It is cost effective as the user control the offline application himself and does not go for professional service.
- It saves time as it speeds up every aspect of the employee database management, payroll process with a range of automated features, and Sales & Inventory Management.
- It is secure as the employee database and the payroll process is managed by the admin in house rather than sending private information to a third party.
- Validating procedures and checks restrict user from making mistakes.
- The software is easy to use and is user friendly so no expertise is required.
- The calculations are automated so no chance of error.
- It does not require internet connection

CONS:

- Currently, It do not automatic fetch data from internet about government notification .
- Currently, The output is not available in PDF format
- It is not a cross-platform application.

FEASIBILITY OF SOFTWARE (CMS)

After identifying the scope of the project, the feasibility study is needed to be carried out. It is basically keeping the following points in mind.

Building the software for meeting the scope: This software has met the scope. As there is no data involved in the system, processing on the file, and the behaviour of this project is already identified and bundled in quantitative manner.

The processing of this software is very simple as it has been designed in python.

Technically feasible: This software is very much technically feasible. This software is very much concerned with specifying equipment and the software will successfully satisfy almost all the admin's requirements. The technical need for this system may vary considerably but might include:

- a. The facility to produce output in a given time.
- b. Response time under certain conditions.
- c. Ability to process data at a particular speed.

Therefore, the basic input/output of data is identified. So, the project can easily be build up and it will also be technically feasible.

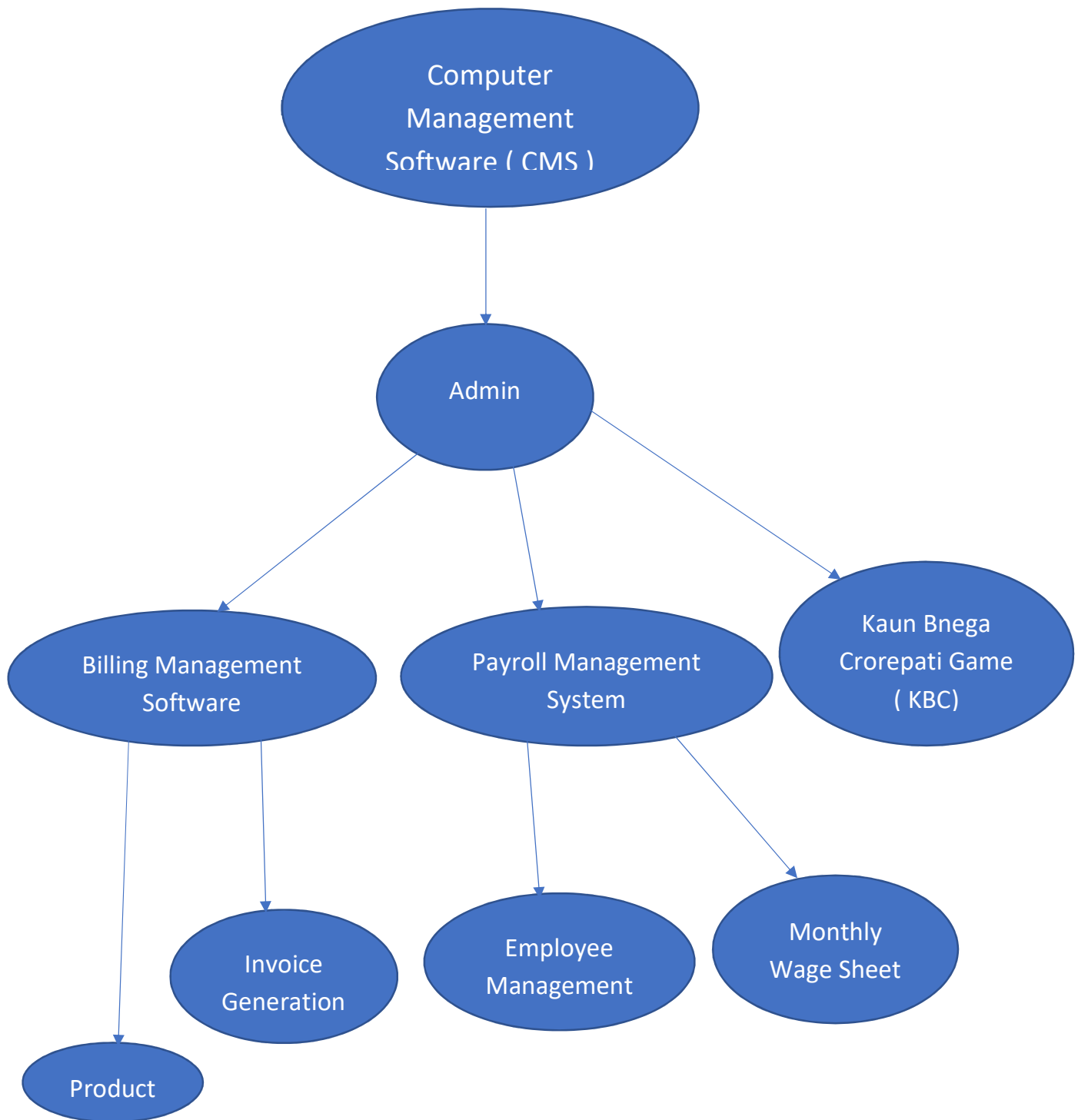
State of Art: The project is very much within the state of art since the project is a WINDOWS based; it uses very modern and common technique.

Beside it is very much modern and user friendly. It also works as middleware i.e. only in between the user and the file. So, it is completely a state of art project.

Financially Feasible: The project is very much financially feasible. The implementation and development cost of this software under the reach of any college.

Moreover, it requires very basic training for the use. So, training cost can be neglected and the resources of this software are very much available. It also reduces the labour and extra cost to be paid for labour. So indeed, it is financially feasible.

FLOW CHART



SYSTEM REQUIREMENTS

Languages: Python

Database: CSV, Text File

Operating System: Windows 10

Processor: Intel core i5

RAM: 4GB

Hard Disk: 1TB

Python Modules:

pandas

CSV

datetime

SAMPLE SOURCE CODE

Interface.py

```
print(''
```

**COMPANY
MANAGEMENT
SOFTWARE**

```
''')
```

```
ans="Y"
```

```
while ans.upper()=="Y":
```

```
    ID=input("Enter Your USER ID : ")
```

```
    Pass=input ( "Enter your Password : ")
```

```
    if ID.upper()=="ADMIN" and Pass=="Admin123$":
```

```
        print (" Log In Succefully")
```

```
        print(''
```

```
Welcome Admin !!!!
```

1. Choose 1 for Billing Managment
2. Choose 2 for Payroll Managment
3. Choose 3 to Relax and Play KBC :)
4. Choose 4 to Exit the Software

```
''')
```

```
CHOOSSED=int(input(" Choose the above option to Continue : "))

if CHOOSSED==1:

    import main

elif CHOOSSED==2:

    print("Loading.....")

    import PAYROLL

elif CHOOSSED==3:

    import KBCV3

elif CHOOSSED==4:

    break

else :

    print ( " You have choose the Wrong Option ")

    break

else :

    print("Log In Failed !!!!!")

    ans=input(" Again want to retry (Y/N) : ")
```


KBCV3.py

```
wrong= True
```

```
ques_no=0
```

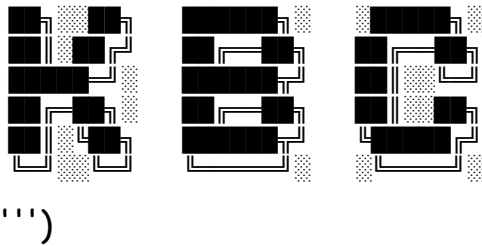
```
random=0
```

```
import random
```

```
from datetime import date
```

```
print('''
```

```
WELCOME TO KAUN BANEGA CROREPATI...LET'S PLAY
```



```
''')
```

```
nme=input("Enter Your Name " )
```

```
print("Hello, Today Mr./Mrs. ",nme," Is sitting with us on the Hot-seat to play KBC ")
```

```
# total prize for each question
```

```
amount_won = [1000, 2000, 3000, 5000, 10000,  
               20000, 40000, 80000, 160000,  
               320000, 640000, 1250000, 2500000,  
               5000000, 10000000, 70000000]
```

```
# Options for 50:50 lifeline
```

```
op1 = [ ' ', ' ', ' ', ' ', 'Srinagar',  
        ' ', ' ', ' ', ' ', 'Cricket', '1920', ' ', ' ', ' ',  
        ' ', ' ', ' ', 'Cricket', ' ', ' ', ' ', ' ', ' ']
```

'Kolkata', 'Wrestling', ' ', ' ', ' ',
 ' ', 'China', 'Thar', ' ', ' ', ' ',
 'Israel', ' ', ' ', ' ', 'Arjan Singh',
 'Parliament of India', ' ', ' ', ' ',
 'Mohd Hamid Ansari', ' ', ' ', 'Mahatma Gandhi',
 'Hanuman']

op2 = ['AB De Villiers', ' ', '9', ' ',
 ' ', ' ', '1928', ' ', 'Cricket',
 'Yuvraj Singh', 'Cricket', 'Football', 'West Indies',
 'Mumbai', 'Swimming', ' ', ' ', ' ',
 ' ', ' ', 'Sahara', 'Mahishmati', ' ',
 'Jordan', ' ', 'Che Guevera', ' ',
 ' ', ' ', ' ', ' ', ' ', ' ',
 ' ']

op3 = ['Shahid Afridi', 'Dishonest', ' ', 'Amritsar',
 'Sindhi', ' ', ' ', 'Pakistan', ' ',
 'MS Dhoni', ' ', ' ', 'South Africa', ' ',
 ' ', 'Henry Becquarrel', 'Tephrosia', 'Dermatitis',
 'Japan', ' ', ' ', 'Bypass', ' ',
 'Mridangam', ' ', ' ', ' ',
 'darjeeling', 'Japanese Encephalitis', 'Mohd Hidayatullah',
 'Saina Nehwal', ' ', 'Shiva']

op4 = [' ', 'Miserly', '8', ' ', 'English', 'Hockey',
 ' ', 'Australia', 'Football', ' ', 'Polo', ' ',

```

        'None of these', 'Indigofera',
        'Cholera', 'Badami', 'Debridement',
        'Dafli', 'Vladimir Lenin', 'Aspy Engineer',
        'Mangalyaan', 'Kohima', 'Plague', 'Jwala Gutta',
        'Mother Teresa', '']

```

```

op = [op1, op2, op3, op4]

```

```

# list of lifelines

```

```

list_life = [1, 2, 3, 4]

```

```

def lifeline(ran, opts, op):

```

```

    m = 1

```

```

    lifelines = ['Audience Poll', 'Fifty Fifty',
                 'Double dip', 'Flip the question']

```

```

    print("Lifelines are \t", lifelines[0], '\t', lifelines[1],
          '\t', lifelines[2], '\t', lifelines[3], '\n\n')

```

```

    if list_life == []:

```

```

        print("You don't have lifelines remaining\t")

```

```

        return None

```

```

    print("Press 1 for audience,2 for 50:50, 3 for double dip\
or 4 for flip the question\t")

```

```
while(m != 0):

    get = int(input())

    if get == 1:

        if get in list_life:

            m = 0

            list_life.remove(1)

            great = audience(ran, opts)

        else:

            print("You don't have audience poll\t")

    elif get == 2:

        if get in list_life:

            m = 0

            great = fifty(ran, op)

            list_life.remove(2)

        else:

            print("You don't have 50:50 \t")

    elif get == 3:

        if get in list_life:

            m = 0

            great = doubleDip(ran)

            list_life.remove(3)
```

```

        else:

            print("You don't have double dip\t")

elif get == 4:

    if get in list_life:

        m = 0

        great = flip()

        list_life.remove(4)

    else:

        print("You don't have lifeline to flip the question\t")

else:

    print("Choose correct option")

return great

```

```

def audience(ran, opts):

```

```

    print("According to audience\n")

```

```

    print('1.', opts[0][ran], "%", '\t', '2.',
          opts[1][ran], "%", '\t', '3.', opts[2][ran],
          "%", '\t', '4.', opts[3][ran], "%", '\nenter your choice\t')

```

```

    print("Would you like to take lifeline again,if yes then press\
9 or Press 0 to Quit\t")

```

```
choice = int(input())
```

```
if choice == 9:
```

```
    great = lifeline(ran, opts, op)
```

```
    return great
```

```
elif choice == answer[ran]:
```

```
    great = 1
```

```
    print("Correct answer,well done!..")
```

```
elif choice == 0:
```

```
    great = -2
```

```
else:
```

```
    great = 0
```

```
    print("Incorrect")
```

```
    print("Correct Answer is :", options[answer[ran]-1][ran])
```

```
return great
```

```
def fifty(ran, op):
```

```
    print("Q."+questions[ran])
```

```
    for num, option in enumerate(op):
```

```
        print(str(num+1)+ "." +option[ran])
choice_fifty = int(input("enter your choice \t"))

if choice_fifty == answer[ran]:
    print("Correct Answer.....")
    great = 1

else:
    great = 0
    print("wrong answer")
    print("Correct Answer is :", options[answer[ran]-1][ran])

return great
```

```
def doubleDip(ran):

    # double dip gives 2 chances
    print("Select two options\n")
    trial1 = int(input())

    if answer[ran] == trial1:
        great = 1
        print("Correct Answer,well done.....")

    else:
        print("Your first trial is wrong, choose another\t")
```

```
trial2 = int(input())
```

```
if answer[ran] == trial2:
```

```
    great = 1
```

```
    print("Correct Answer\t")
```

```
else:
```

```
    print("Your second trial is also wrong..Better luck next time..\t")
```

```
    print("Correct Answer is :", options[answer[ran]-1][ran])
```

```
    great = 0
```

```
return great
```

```
def flip():
```

```
    return -1
```

```
def amount(correct_ans):
```

```
    print(amount_won[correct_ans-1])
```

```
    if amount_won[correct_ans-1] == 10000:
```

```
        print("Completed 1st stage")
```

```
    elif amount_won[correct_ans-1] == 320000:
```

```
        print("Completed 2st stage")
```

```
    elif amount_won[correct_ans-1] == 70000000:
```



```
print("You have won Rs 7 CRORE")
```

```
return amount_won[correct_ans-1]
```

```
questions = [
```

```
'In ODI Cricket, who created the record of scoring the \
fastest century in just 31 balls ?',
```

```
' If you call someone 'Makkhichoos' then what are you \
calling him ?',
```

```
'How many players of a Kho-Kho team can play on the field\
during the match ?',
```

```
'Which of these Indian cities is closest to the Pakistani \
city of Lahore ?',
```

```
'The language spoken by the people by Pakistan is ?',
```

```
'The term"Googly" is associated with ?',
```

```
'India first took part in the olympic games in the year ?',
```

```
'Who are Kangaroos ?',
```

```
'Oval stadium in England is associated with ?',
```

```
'In 2011 India won the World Cup. Who was adjudicated as the\
man of the series in the tournament ? ',
```

```
'Eden Gardens in Kolkata is ----- stadium.? ',
```

```
'Ronaldo is associated with ? ',
```

```
'Icc's 2007, the World Cup Cricket was held in ? ',
```

```
'Wankhede Stadium is at ? ',
```

```
'World's most ancient game is ? ',
```

```
'Stethoscope was invented by ? ',
```

'A dye is prepared from ',

'Which disease is caused by the fungi? ',

'Which is the Land of the Rising Sun? ',

'The desert that lies on the boundary between India and Pakistan \

is ',

' In which kingdom is the story of the 'Bahubali' series of films\

mainly set?',

'What is the common name for surgery conducted on coronary arteries\

that supply blood to the heart ?',

' In July 2017, Narendra Modi Become the first Indian Prime Minister\

to visit which country ?',

'Which of these musical instrument is held in one hand and played with\

the other ?',

' On the last day of his life Bhagat Singh was reading a book about\

the Ideology of which revolutionary ?',

'Which Air force officer had the unique honour of leading the fly-post\

over the Red fort in Delhi on 15 August 1947 ?',

'Which image appears on the flip side of the new 2000 Rs Note, launched\

in 2016?',

'Which Indian hill station gets its name from the Tibetan words that mean\

'land of the thunderbolt'?',

'Which of these diseases is transmitted by mosquitoes?',

'Who among these has served as the Ambassador of India to the United Nations?',

' Who was the first Indian to win the World Junior Badminton Championships?',

'Which of the following is a recipient of the Nobel Peace Prize?',

'The cave temples at the historical site of Elephanta are dedicated to which\

God?'

]

option1 = ['Corey Anderson', 'Evil', '10', 'Srinagar', 'Hindi',
'Cricket', '1920', 'Bangladesh', 'polo', 'Virat Kohli',
'Tennis', 'Cricket', 'Australia', 'Kolkata', 'Wrestling',
'Bessemer', 'Sida', 'Polio', 'China', 'Thar', 'Magadh',
'Cataract', 'Israel', 'Tabla', 'Antonio Gramsci',
'Arjan Singh', 'Parliament of India', 'Gangtok', 'Rabies',
'Mohd Hamid Ansari', 'P V Sindhu', 'Mahatma Gandhi', 'Hanuman']

option2 = ['AB De Villiers', 'Humble', '9', 'Jaisalmer', 'Palauan',
'Football', '1928', 'Kenya', 'Cricket', 'Yuvraj Singh',
'Cricket', 'Football', 'West Indies', 'Mumbai', 'Swimming',
'Rane Laennec', 'Tridax', 'Malaria', 'Taiwan', 'Sahara',
'Mahishmati', 'Gastric', 'Jordan', 'Santoor', 'Che Guevera',
'Pratap Chandra Lal', 'Tractor', 'Aizawl', 'Tetanus', 'I K Gujral',
'Aparna Balan', 'Swami Vivekananda', 'Vishnu']

option3 = ['Shahid Afridi', 'Dishonest', '7', 'Amritsar', 'Sindhi',
'Badminton', '1972', 'Pakistan', 'Hockey', 'MS Dhoni',
'Hockey', 'Hockey', 'South Africa', 'Delhi', 'Boxing',
'Henry Becquarrel', 'Tephrosia', 'Dermatitis', 'Japan',
'Gobi', 'Kalinga', 'Bypass', 'Saudi Arabia', 'Mridangam',
'Leon Trotsky', 'Subroto Mukarjee', 'Red Fort', 'darjeeling',
'Japanese Encephalitis', 'Mohd Hidayatullah', 'Saina Nehwal',
'Rabindranath Tagore', 'Shiva']

```
option4 = ['Rohit Sharma', 'Miserly', '8', 'Udhampur', 'English', 'Hockey',  
          '1976', 'Australia', 'Football', 'Zaheer Khan', 'Polo', 'Tennis',  
          'India', 'Jaipur', 'Running', 'None of these', 'Indigofera',  
          'Cholera', 'Australia', 'None of these', 'Badami', 'Debridement',  
          'Qatar', 'Dafli', 'Vladimir Lenin', 'Aspy Engineer', 'Mangalyaan',  
          'Kohima', 'Plague', 'Zakir Hussain', 'Jwala Gutta', 'Mother Teresa',  
          'Kamadeva']
```

```
options = [option1, option2, option3, option4]
```

```
# answer key
```

```
answer = [2, 4, 2, 3, 3, 1, 1, 4, 2, 2, 2, 2, 2, 2, 1,  
          3, 4, 3, 3, 1, 2, 3, 1, 4, 4, 1, 4, 3, 3, 1,  
          3, 4, 3]
```

```
wrong = False
```

```
# correct variable for total correct answer
```

```
correct = 0
```

```
total_amt = 0
```

```
# option list for audience poll
```

```
opt1 = [30, 24, 10, 0, 1, 72, 99, 0, 9, 2, 0, 2, 10, 1,  
        100, 1, 0, 3, 2, 98, 21, 35, 50, 40, 45, 65, 50,  
        48, 5, 70, 20, 30, 20]
```

```
opt2 = [60, 32, 80, 0, 2, 5, 1, 1, 91, 94, 95, 87, 90, 96,
```

```
0, 0, 2, 12, 13, 1, 60, 20, 30, 2, 0, 20, 0, 1, 10,  
12, 20, 20, 10]
```

```
opt3 = [2, 4, 0, 100, 97, 0, 0, 1, 0, 2, 5, 11, 0, 3, 0, 99,  
2, 82, 82, 0, 18, 40, 10, 4, 1, 10, 0, 50, 70, 15,  
35, 10, 64]
```

```
opt4 = [8, 40, 10, 0, 0, 23, 0, 98, 0, 2, 0, 0, 0, 0, 0, 0,  
96, 3, 3, 1, 1, 5, 10, 54, 54, 5, 50, 1, 15, 3, 25,  
40, 6]
```

```
opts = [opt1, opt2, opt3, opt4]
```

```
condition, ques_no = 1, 0
```

```
while(wrong != True):
```

```
    ques_no += 1
```

```
    ran = random.randint(0, len(questions)-1)
```

```
    print("\n\nQ.", ques_no, ":-", end="")
```

```
    print(questions[ran])
```

```
    for num, option in enumerate(options):
```

```
        print(str(num+1)+". "+option[ran])
```

```
    print("Would you like to take lifeline, if yes, press\
```

```
9\n Choose any option: or you can quit by pressing 0 \t\t")
```

```
give_answer = int(input())
```

```
if give_answer == 9:
```

```
    # condition variable is to count lifelines used
```

```
    if condition <= 4:
```

```
        condition += 1
```

```
        great = lifeline(ran, opts, op)
```

```
        if great == 0:
```

```
            if total_amt < 10000:
```

```
                total_amt = 0
```

```
            elif total_amt < 320000:
```

```
                total_amt = 10000
```

```
            elif total_amt < 70000000:
```

```
                total_amt = 320000
```

```
            break
```

```
        elif great == -1:
```

```
            ques_no -= 1
```

```
            pass
```

```
        elif great == None:
```

```
            print("Choose any option or press 0 to quit\t")
```

```
            give_ansr = int(input())
```

```
            if answer[ran] == give_ansr:
```

```
                print("Correct answer, great")
```

```
correct += 1
```

```
elif great == -2:
```

```
    break
```

```
else:
```

```
    correct += 1
```

```
    print("You have won Rs=", end="")
```

```
    total_amt = amount(correct)
```

```
else:
```

```
    print(
```

```
        "You have used your all lifelines\t\t\n Choose any option: \
```

```
        or you can quit by pressing 0\t\t")
```

```
    give_ans = int(input())
```

```
    key = answer[ran]
```

```
    if give_ans == 0:
```

```
        total_amt = amount(correct)
```

```
        break
```

```
    elif key == give_ans:
```

```
        print("Correct, You have won Rs. =", end="")
```

```
        correct += 1
```

```
        total_amt = amount(correct)
```

```
else:
```

```
print("Wrong Answer....")

print("Correct Answer is : ", options[answer[ran]-1][ran])

if total_amt < 10000:

    total_amt = 0

elif total_amt < 320000:

    total_amt = 10000

elif total_amt < 70000000:

    total_amt = 32000

wrong = True
```

else:

```
key = answer[ran]
```

```
if give_answer == 0:
```

```
    if correct != 0:
```

```
        total_amt = amount(correct)
```

```
    break
```

```
elif key == give_answer:
```

```
    print("Correct answer.., You have won Rs.=", end="")
```

```
    correct += 1
```

```
    total_amt = amount(correct)
```

else:

```
    print("Wrong Answer...Better luck next time...")
```

```
    print("Correct Answer is :", options[answer[ran]-1][ran])
```

```
    if total_amt < 10000:
```



```

        total_amt = 0

    elif total_amt < 320000:

        total_amt = 10000

    elif total_amt < 700000000:

        total_amt = 320000

    wrong = True

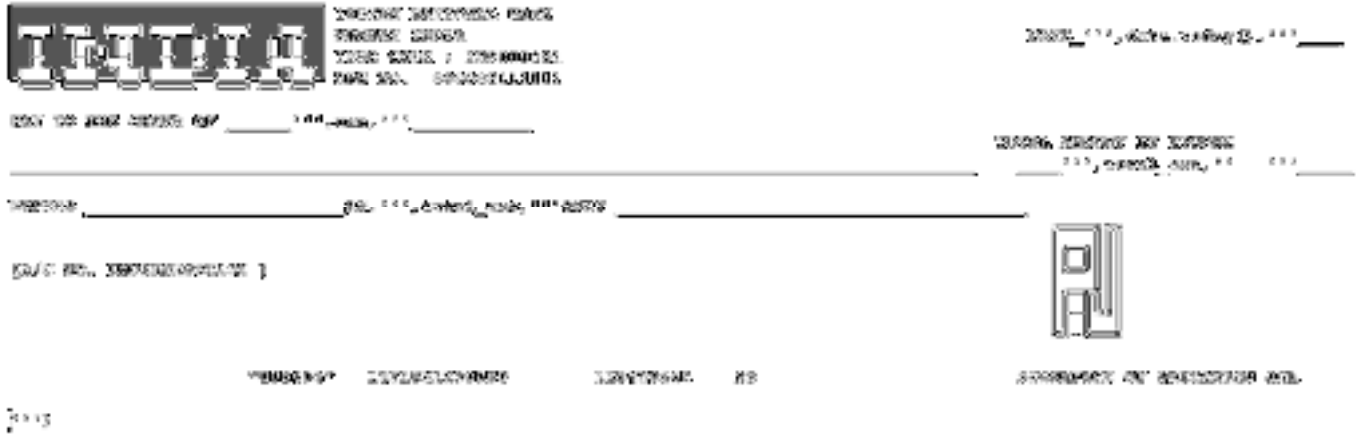
if correct == 16: # total questions are 16

    break

# delete previous question and its options from list
del questions[ran]
del option1[ran]
del option2[ran]
del option3[ran]
del option4[ran]
del answer[ran]
del opts[0][ran]
del opts[1][ran]
del opts[2][ran]
del opts[3][ran]
del op[0][ran]
del op[1][ran]
del op[2][ran]
del op[3][ran]
options = [option1, option2, option3, option4]

```

**YOUR CHEQUE IS READY FOR WINNING AMOUNT,
CONGO!!**



PAYROLL.PY

```
import pandas as pd
```

```
import datetime
```

```
import csv
```

```
def emp_entry():
```

```
    ec = input("Enter employee code : ")
```

```
    nm = input("Enter Name of Employee: ")
```

```
    fn = input("Enter Father Name of Employee: ")
```

```
    dg = input("Enter Designation : ")
```

```
    ge=input("Enter Gender : ") or "M"
```

```
    db = input("Enter Date of Birth (YYYY-MM-DD) : ")
```

```
    dj = input("Enter Date of Joining (YYYY-MM-DD) : ")
```

```
    uan = input("Enter UAN of Employee : ")
```

```
    ip = input("Enter IP ( ESIC) of the Employee : ")
```

```
    mb =input("Enter Mobile Number : ")
```

```
    ac= input("Enter Bank Account Number: ")
```

```
    fc = input("Enter IFSC code of Bank Account : ")
```

```
    hr=input("Enter employee's Department : ")
```

```
    lc=input("Enter employee's Location : ")
```

```
    lc_A=input("Enter employee's Location Category ( A,B,C) : ") or "C"
```

```
    wage=input("Enter Per Day Wage Acc. As per Current Rates ") or "0"
```

```
    data = [[ec, nm,fn, dg , ge, db, dj,uan,ip, mb, ac , fc, hr,lc,lc_A,wage,]]
```

```
    df =
```

```
pd.DataFrame(data,columns=['EMP_CODE','NAME','FATHER_NAME','DESIGNATION','GENDER','DOB','DOJ','UAN','IP','MOB','AC_NO','IFSC','DEPARTMENT',
```

```

        'LOCATION','LOCATION_TYPE','BASIC_WAGE'])

df.to_csv('MASTERS.csv', mode = 'a',header=False, index=False)

df.to_csv('MASTERSW.csv', mode = 'a', index=False)

def pay():

    US_A=float(input("Enter Per Day Wages for Unskilled Staff (A) : "))

    SS_A=float(input("Enter Per Day Wages for Semi-Skilled Staff (A) : "))

    S_A=float(input("Enter Per Day Wages for Skilled Staff (A) : "))

    HS_A=float(input("Enter Per Day Wages for Highly Skilled Staff (A) : "))

    US_B=float(input("Enter Per Day Wages for Unskilled Staff (B) : "))

    SS_B=float(input("Enter Per Day Wages for Semi-Skilled Staff (B) : "))

    S_B=float(input("Enter Per Day Wages for Skilled Staff (B) : "))

    HS_B=float(input("Enter Per Day Wages for Highly Skilled Staff (B) : "))

    US_C=float(input("Enter Per Day Wages for Unskilled Staff (C) : "))

    SS_C=float(input("Enter Per Day Wages for Semi-Skilled Staff (C) : "))

    S_C=float(input("Enter Per Day Wages for Skilled Staff (C) : "))

    HS_C=float(input("Enter Per Day Wages for Highly Skilled Staff (C) : "))

    BNS=float(input("Enter BONUS % : "))

    GRTY=float(input("Enter GRATUITY/ RETIREMENT COMPONSATION % : "))

    LWW=float(input("Enter LEAVE WITH WAGE % : "))

    HOLIDAY=''

    LWF=float(input("Enter LWF % : "))

    EPF=float(input("Enter EPF % : "))

    ESI=float(input("Enter ESIC % : "))

    data = {'WAGECOMPONENET' : ['Min. Wage for Part A (Unskilled)','Min. Wage for Part A (Semi Skilled)','Min. Wage for Part A (Skilled)','Min. Wage for Part A (Highly skilled)',

        'Min. Wage for Part B (Unskilled)','Min. Wage for Part B (Semi Skilled)','Min. Wage for Part B (Skilled)','Min. Wage for Part B (Highly skilled)',

```

'Min. Wage for Part C (Unskilled)', 'Min. Wage for Part C (Semi skilled)', 'Min. Wage for Part C (Skilled)', 'Min. Wage for Part C (Highly skilled)',

'Bonus', 'Gratuity/Retirement Compensation', 'Leave with Wages', 'Labour Welfare Fund',

'Employee Provident Fund', 'Employee State Insurance'],

'WAGEVALUES' :

[US_A,SS_A,S_A,HS_A,US_B,SS_B,S_B,HS_B,US_C,SS_C,S_C,HS_C,BNS,GRTY,LWW,LWF,EPF,ESI],

'WAGE_N'

:['US_A','SS_A','S_A','HS_A','US_B','SS_B','S_B','HS_B','US_C','SS_C','S_C','HS_C','BNS','GRTY','LWW','LWF','EPF','ESI']}]

```
df = pd.DataFrame.from_dict(data)
```

```
df.to_csv('PAY.csv', mode = 'w', header=False, index=False)
```

```
df.to_csv('PAYW.csv', mode = 'w', index=False)
```

```
def salary_entry():
```

```
    y = input(" Enter the salary year (press enter for current year otherwise input new year: " +  
str(datetime.datetime.today().strftime('%Y')) or  
(str(datetime.datetime.today().strftime('%Y')))
```

```
    m = input("Enter the salary month (press enter for current month otherwise input new  
month:" + str(datetime.datetime.today().strftime('%m'))or  
(str(datetime.datetime.today().strftime('%m')))
```

```
master=open('Masters.csv','r')
```

```
master2=open('Mastersw.csv','r')
```

```
master_r=csv.reader(master)
```

```
master_d=csv.DictReader(master2)
```

```
count=0
```

```
for rec in master_r:
```

```
masterdata=[rec]
```

```
count=count+1
```

```
cond={y[2]:float(y[1]) for y in [x.split(',') for x in open('pay.csv').read().split('\n') if x]}  
#DICT COMPREHENSION
```

```
print("Enter salary details for the " + str(m) + "/" + str(y))
```

```
Main=open('WAGEDATA.csv','w',newline='')
```

```
cwrt=csv.writer(Main)
```

```
cwrt.writerow(['EMP_NO','NAME','FATHER_NAME','UAN','ESI_NO','DESIG','ATTENDANC  
E_A','ATTENDANCE_B','ATTENDANCE_C','ATTENDANCE_G','OVERTIME','HOLIDAY',  
               'BASIC_WAGE','OVERTIME','BONUS','GRATUITY/RETIREMENT  
COMPONSATION','LEAVE WITH WAGES','OTHER ALLOWANCE','HOLIDAY AMT','TOTAL',  
               'EMPLOYEE PROVIDENT FUND','EMPLOYEE STATE INSURANCE','LABOUR  
WELFARE FUND','TOTAL DEDUCTION','NET SALARY ','WAGE MONTH','YEAR'])
```

```
df_emp=[]
```

```
wage_r=[]
```

```
empno=[]
```

```
name=[]
```

```
fname=[]
```

```
desig=[]
```

```
uan=[]
```

```
ip=[]
```

```
rate=[]
```

```
for col in master_d:
```

```
    empno.append(col['EMP_CODE'])
```

```

name.append(col['NAME'])
fname.append(col['FATHER_NAME'])
desig.append(col['DESIGNATION'])
uan.append(col['UAN'])
ip.append(col['IP'])
rate.append(col['BASIC_WAGE'])

```

```

while count > 0:

```

```

for x in empno:

    print("Employee Code : " + str(x) + "\n")

    ata=(eval(input("Enter No. of days worked in Shift A : ")))
    atb=(eval(input("Enter No. of days worked in Shift B : ")))
    atc=(eval(input("Enter No. of days worked in Shift C : ")))
    atg=(eval(input("Enter No. of days worked in General Shift : ")))
    ot=(eval(input("Enter No. of Over Time worked in Hrs. : ")))
    hldy=(eval(input("Enter No. of Paid Holidays : ")))
    allw=(eval(input("Enter other allowance (or 0): ") or '0'))
    deduc=(eval(input("Enter other deductions (or 0): ") or '0'))

    for r,n,f,u,i,d in zip(rate,name,fname,uan,ip,desig) :

        r=float(r)

        BASIC_WAGE = r*(ata+atb+atc+atg)

        OVERTIME = (r/4) * ot

        BONUS = BASIC_WAGE * (cond['BNS']/100)

        GRATUITY= BASIC_WAGE * (cond['GRTY']/100)

        LEAVE_WITH_WAGES = BASIC_WAGE * (cond['LWW']/100)

        OTHER_ALLW = allw

```

```

        HOLIDAY = BASIC_WAGE * hldy

        TOTAL =
BASIC_WAGE+OVERTIME+BONUS+GRATUITY+LEAVE_WITH_WAGES+OTHER_ALLW

        if BASIC_WAGE<15000:

            EPF=BASIC_WAGE*12/100

            else : EPF=1800

            ESIC = BASIC_WAGE* 0.75/10

            ODEDUCT = deduc

            LWF= BASIC_WAGE* 0.2/100

            TOTAL_DEDUC=EPF+ESIC+ODEDUCT

            NETSAL = TOTAL-TOTAL_DEDUC+HOLIDAY


df_emp=[x,n,f,u,i,d,ata,atb,atc,atg,ot,hldy,BASIC_WAGE,OVERTIME,BONUS,GRATUITY,LEAV
E_WITH_WAGES,OTHER_ALLW,

        HOLIDAY,TOTAL,EPF,ESIC,LWF,TOTAL_DEDUC,NETSAL,m,y]

print(df_emp)

count=count-1

cwrt.writerow(df_emp)

Main.close()

master.close()

master2.close()

```

```

def Date_operations():

    x = datetime.datetime.today().strftime('%Y-%m-%d')

    print(x)

def Sdf_show():

```



```

df = pd.read_csv('WAGEDATA.csv')

print(df)

def Show_Rates():

    df = pd.read_csv('PAYW.csv')

    print(df)

def Show_EMP():

    df = pd.read_csv('MASTERSW.csv')

    print(df)

def Salary_show():

    df = pd.read_csv('WAGEDATA.csv')

    df.to_csv(r'C:\Users\Public\Documents\SALARY.csv' , mode='w')

    print("File Succesfully Generated at Following Path : C:\Users\Public\Documents\Salary.csv" )


while (True):

    print("1 : ADD EMPLOYEE DETAILS")

    print("2 : SHOW EMPLOYEE DETAILS")

    print("3 : FIX STRUCTURAL RATES")

    print("4 : SHOW CURRENT DA AND HRA RATES")

    print("5 : WAGESHEET ENTRY ")

    print("6 : SHOW WAGESHEET")

    print("7 : EXPORT WAGESHET (CSV FILE IN EXCEL)")

    print("8 : Exit")

    choice = int(input("Please Select An Above Option: "))

    if(choice == 1):

        emp_entry()

    elif (choice==2):

        Show_EMP()

```

```
elif (choice==3):  
    pay()  
elif (choice==4):  
    Show_Rates()  
elif (choice==5):  
    salary_entry()  
elif (choice == 6):  
    Sdf_show()  
elif (choice == 7):  
    Salary_show()  
elif (choice == 8):  
    break  
else:  
    print(" Wrong choice.....")
```

Main.py

```
# module mane: main

import read

import purchase

import write


again = "Y"

while again.upper() == "Y":

    a = read.read_file()

    b = purchase.purchase(a)

    write.over_write(a, b)

    again = input("\nAre there any new Invoice you want to Generate?(Y/N) ")

print("\nThank you Using our Software!!!")

print("Please check your invoice for your further details, \nWhich we have created in .txt file format.")
```

Purchase.py

"""

module name: purchase

function name: purchase

overview of this function:

- 1) Interaction for what and how much have been sold.
- 2) Check the user interaction valid or not with exception handling.
- 3) Calculating the customer purchase product with discount(if discountable)
- 4) Show the last update of the product
- 5) Write the invoice for customer with unique naming

"""

def purchase(List):

 L = List # assign list with variable name 'L'.

 a_name = input("Please enter Party Name: ")

 cname='SPD International Infratech Private Limited'

 print("\nHello " + a_name + "! Welcome to ",cname, ".\nPlease select product as per Item sold .")

 q = {} # assign empty dictionary with variable name 'q'.

 flag = "Y"

 while flag.upper() == "Y": # check and go if flag is 'Y' or 'y'.

 product = input("\nWhich products have been Sold ? ")

 product_name = product.upper() # change the string value in the upper case.

 if product_name == L[0][0].upper() \

 or product_name == L[1][0].upper() \

 or product_name == L[2][0].upper(): # check the user entered product name with stock of store

```

p = True

while p == True:

    try:

        p_quantity = int(input("How many " + product + " has been sold : "))

        p = False

    except: # executes, if customer entered unexpected value.

        print("\t\tError!!!\nPlease enter integer value!! ")

if product_name == L[0][0].upper() and p_quantity <= int(L[0][2]):

    q[product_name] = p_quantity

elif product_name == L[1][0].upper() and p_quantity <= int(L[1][2]):

    q[product_name] = p_quantity

elif product_name == L[2][0].upper() and p_quantity <= int(L[2][2]):

    q[product_name] = p_quantity

else:

    print(

        "\nSorry!! " + a_name + "! , " + product + " is Currently out of stock.\nWe will add
stock of " + product + " later. \n Kindly Purchase"+ product+" to sell it.\n")

    flag = (input(a_name + " do you want add more products?(Y/N)"))

else:

    print("sorry!! " + product + " is not available in our store.")

    print("\nChoose from following products please!")

    print("-----")

    print("PRODUCT\t\tPRICE\t\tQUANTITY")

    print("-----")

    for i in range(len(L)):

        print(L[i][0], "\t\t", L[i][1], "\t\t",

```

```

        L[i][2]) # print, last updated product name, quantity and price.

print("-----")

print("\nYou Chooosed Items and it's Quantity respectively:\n", q, "\n")
'''

```

In the following operation:

- 1) change every string value in the upper case latter.
- 2) check what is the product entered by user.
- 3) executes respective condition if product is available to the user".

```

'''

f_amount = 0 # final amount

for keys in q.keys():

    if keys == L[0][0].upper():

        p_price = int(L[0][1])

        p_num = int(q[keys])

        p_amount = (p_price * p_num)

        f_amount += (p_price * p_num)

        print("\nTotal cost for phone: ", p_amount)

    elif keys == L[1][0].upper():

        l_price = int(L[1][1])

        l_num = int(q[keys])

        l_amount = (l_price * l_num)

        f_amount += (l_price * l_num)

        print("Total cost for laptop: ", l_amount)

    else:

        h_price = int(L[2][1])

        h_num = int(q[keys])

```

```

h_amount = (h_price * h_num)

f_amount += (h_price * h_num)

print("Total cost for HDD: ", h_amount)

print("\nYour discountable total amount is: ", f_amount)

```

...

In the following operation:

1) ask the user for any discount % in total purchase amount & Check the availibility of discount on the amount.

...

```

disc = float(input("Please enter your expected discount (%): "))

dis = 0.0

if (f_amount >= 5000):

    discount = disc

    dis = (discount * f_amount) / 100

    total = float(f_amount - dis)

    print("You got your expected " + str(disc) + "% discount and amount is: ", dis)

else:

    discount = 0.0

    total = float(f_amount)

    print("You did not got your expected " + str(
        disc) + "% discount\nBecause, your totel purchase is not meet the minimum criteria for "
+ str(
    disc) + "% discount.")

    print("Your payable amount is: ", total)

...

```

In the following operation:

1) write a each unique involve name which includes current date and time with customer name as well.

2) write a purchase product name and details in file (invoice).

3) write a discounted amount and final payable amount in file (invoice).

...

```
import datetime # import system date and time for create a unique invoive name.

dt = str(datetime.datetime.now().year) + "-" + str(datetime.datetime.now().month) + "-" + str(
    datetime.datetime.now().day) + "-" + str(datetime.datetime.now().hour) + "-" + str(
    datetime.datetime.now().minute) + "-" + str(datetime.datetime.now().second)

invoice = str(dt) # unique invoice

t = str(datetime.datetime.now().year) + "-" + str(datetime.datetime.now().month) + "-" + str(
    datetime.datetime.now().day) # date

d = str(t) # date

u = str(datetime.datetime.now().hour) + ":" + str(datetime.datetime.now().minute) + ":" + str(
    datetime.datetime.now().second) # time

e = str(u) # time

file = open(invoice + " (" + a_name + ").txt", "w") # generate a unique invoive name and open it
in write mode.

file.write("=====")

file.write("\n"+cname+"\t\t\t\t\tINVOICE")

file.write("\n\nInvoice: " + invoice + "\t\t\tDate: " + d + "\n\t\t\t\t\tTime: " + e + "")

file.write("\nName of Customer: " + str(a_name) + "")

file.write("\n=====")

file.write("\nPARTICULAR\tQUANTITY\tUNIT PRICE\tTOTAL")

file.write("\n-----")
```



```
for keys in q.keys(): # In this loop, write in a file only those product which is purchase by user.
```

```
    if keys == "PHONE":
```

```
        file.write(
```

```
            str("\n" + str(keys) + " \t\t " + str(q['PHONE']) + " \t\t " + str(L[0][1]) + " \t\t " + str(p_amount)))
```

```
    elif keys == "LAPTOP":
```

```
        file.write(str(
```

```
            "\n" + str(keys) + " \t\t " + str(q['LAPTOP']) + " \t\t " + str(L[1][1]) + " \t\t " + str(l_amount)))
```

```
    else:
```

```
        file.write(
```

```
            str("\n" + str(keys) + " \t\t " + str(q['HDD']) + " \t\t " + str(L[2][1]) + " \t\t " + str(h_amount)))
```

```
file.write("\n\n-----")
```

```
file.write("\n\t\t\tYour discountable amount: " + str(f_amount))
```

```
file.write("\n-----")
```

```
file.write("\n\t\t\t Your " + str(discount) + "% discounted amount is: " + str(dis))
```

```
file.write("\n-----")
```

```
file.write("\n\t\t\t\tYour payable amount is: " + str(total))
```

```
file.write("\n-----")
```

```
file.write("\n\n\t\tThank You " + a_name + " for your shopping.\n\t\t\t\tSee you again!")
```

```
file.write("\n=====")
```

```
file.close()
```

```
return q
```

read.py

```
def read_file(): # Function is defined with name : 'read_file'

    file = open("products.txt", "r") # open stock file (products.txt) in read mode.

    lines = file.readlines()

    L = [] # assign empty list with name 'L'

    for line in lines:

        L.append(line.replace("\n", "").split(","))

    file.close()

    print("Following products are available in the Store")

    print("-----")

    print("PRODUCT\t\tPRICE\t\tQUANTITY")

    print("-----")

    for i in range(len(L)):

        print(L[i][0], "\t\t", L[i][1], "\t\t", L[i][2]) # prints the available product, price and
quantity

    print("-----")

    return L
```

wrtite.py

```
def over_write(List, Dictionary): # an overwrite function

    L = List # assign list with variable name 'L'

    d = Dictionary # assign Dictionary with variable name 'd'

    """

    Update quantity of product after user purchased some quantity of any product.

    And print remaining stock products.

    """

    for keys in d.keys():

        if keys == "PHONE":

            L[0][2] = str(int(L[0][2])-d['PHONE'])

        elif keys == "LAPTOP":

            L[1][2] = str(int(L[1][2])-d['LAPTOP'])

        else:

            L[2][2] = str(int(L[2][2])-d['HDD'])

    print("\nRemaining Stock Products:\n",L)

    files = open("products.txt", "w") # opens stock file (products.txt) file in write mode.

    for each in L:

        files.write(str(", ".join(each)))

        files.write("\n")

    files.close()

    return
```

USER INTERFACE SNAPSHOT

COMPANY MANAGEMENT SOFTWARE

Enter Your USER ID : Pj
Enter your Password : Pj123
Log In Failed !!!!!
Again want to retry (Y/N) : Y
Enter Your USER ID : Admin
Enter your Password : Admin123\$
Log In Succesfully

Welcome Admin !!!!!

1. Choose 1 for Billing Managment
2. Choose 2 for Payroll Management
3. Choose 3 to Relax and Play KBC :)
4. Choose 4 to Exit the Software

Choose the above option to Continue : |

WELCOME TO MAUN BANEGA CROKOPATI...LET'S PLAY



Enter Your Name Privanah Jha

Hello. Today Mr./Mrs. Privanebu Jna Is sitting with us on the Hot-seat to play KBC

Q. 1 :-Which of the following is a recipient of the Nobel Peace Prize?

1. Mahatma Gandhi
2. Swami Vivekananda
3. Rabindranath Tagore
4. Mother Teresa

Would you like to take lifeline, if yes, press 9

Choose any option; or you can quit by pressing 0

2

[Lifelines are](#) [Audience Poll](#) [Fifty Fifty](#) [Double dip](#) [Flip the question](#)

Press 1 for audience, 2 for 30:30, 3 for double dip or 4 for flip the question

1

According to audience

1. 30 % 2. 20 % 3. 10 % 4. 40 %

enter your choice

Would you like to take lifeline again,if yes then press 9 or Press 0 to Quit

1

Correct answer, well done!...

You have won R=1000

Q. 11 :- On the last day of his life Bhagat Singh was reading a book about the ideology of which revolutionary ?

1. Antonio Gramsci
2. Che Guevara
3. Leon Trotsky
4. Vladimir Lenin

Would you like to take lifeline, if yes, press 9

Choose any options: or you can quit by pressing 0

on

Wrong Answer...Better luck next time...

Correct Answer is : Vladimir Lenin

Your winning amount is Bz. 330000

YOUR CHECKOUT IS READY FOR WINNING AMOUNT. COME!!



INDIAN NATIONAL BANK

NUMBAI INDIA

INSC CODE : INSD0001

BEG NO. 0000011111

DATE 2022-02-17

PAY TO THE ORDER OF Privanshu Jha

TOTAL AMOUNT IN FIGURE

320000

RUPES RS. 320000 ONLY

IA/C NO. TNDIAN2932132 1



008602 1821031230020

12345688 29

SIGNATURE OF PRIYANSHU JHA

```
1. Choose 1 for Billing Management
2. Choose 2 for Payroll Management
3. Choose 3 to Relax and Play KBC : )
4. Choose 4 to Exit the Software
```

| PRODUCT | PRICE | QUANTITY |
|---------|-------|----------|
| Phone | 700 | 92 |
| Laptop | 1400 | 78 |
| HDD | 200 | 153 |

```
Which products have been sold ? HDD
How many HDD has been sold : 13
Privanshu Jha do you want add more products?(Y/N)N
```

```
Your discountable total amount is: 2600
Please enter your expected discount (%): 5
You did not get your expected 5.0% discount
Because, your total purchase is not meet the minimum criteria for 5.0% discount.
Your payable amount is: 2600.0
```

Are there any new Invoice you want to Generate? (Y/N)

| PARTICULAR | QUANTITY | UNIT PRICE | TOTAL |
|------------|----------|------------|-------|
| HDD | 13 | 200 | 2600 |

Thank You Priyanshu Jha for your shopping.
See you again!

Python 3.10.2 (tags/v3.10.2:a58ebcc, Jan 17 2022, 14:12:15) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

***** RESTART: C:\Users\DELL\Desktop\PROJECT\FINAL\INTERFACE.py *****

COMPANY MANAGEMENT SOFTWARE

Enter Your USER ID : Admin
Enter your Password : Admin123\$
Log In Succesfully

Welcome Admin !!!!

1. Choose 1 for Billing Managment
2. Choose 2 for Payroll Managment
3. Choose 3 to Relax and Play KBC :)
4. Choose 4 to Exit the Software

Choose the above option to Continue : 2

Loading.....

- 1 : ADD EMPLOYEE DETAILS
- 2 : SHOW EMPLOYEE DETAILS
- 3 : FIX STRUCTURAL RATES
- 4 : SHOW CURRENT DA AND HRA RATES
- 5 : WAGESHEET ENTRY
- 6 : SHOW WAGESHEET
- 7 : EXPORT WAGESHET (CSV FILE IN EXCEL)
- 8 : Exit

Please Select An Above Option: 1

Enter employee code : SPD01
Enter Name of Employee: Priyanshu
Enter Father Name of Employee: Dharmender
Enter Designation : HS
Enter Gender : M
Enter Date of Birth (YYYY-MM-DD) : 2003-05-24
Enter Date of Joining (YYYY-MM-DD) : 2022-02-01
Enter UAN of Employee : 983283438423
Enter IP (ESIC) of the Employee : 420032111
Enter Mobile Number : 9999999876
Enter Bank Account Number: 50002001231
Enter IFSC code of Bank Account : HDFC0003606
Enter employee's Department : Management
Enter employee's Location : Panipat
Enter employee's Location Category (A,B,C) : C
Enter Per Day Wage Acc. As per Current Rates 769

- 1 : ADD EMPLOYEE DETAILS
- 2 : SHOW EMPLOYEE DETAILS
- 3 : FIX STRUCTURAL RATES
- 4 : SHOW CURRENT DA AND HRA RATES

5 : WAGESHEET ENTRY
 6 : SHOW WAGESHEET
 7 : EXPORT WAGESHET (CSV FILE IN EXCEL)
 8 : Exit
 Please Select An Above Option: 2
 EMP_CODE NAME FATHER_NAME ... LOCATION LOCATION_TYPE BASIC_WAGE
 0 SPD01 Priyanshu Dharmender ... Panipat C 769

[1 rows x 16 columns]
 1 : ADD EMPLOYEE DETAILS
 2 : SHOW EMPLOYEE DETAILS
 3 : FIX STRUCTURAL RATES
 4 : SHOW CURRENT DA AND HRA RATES
 5 : WAGESHEET ENTRY
 6 : SHOW WAGESHEET
 7 : EXPORT WAGESHET (CSV FILE IN EXCEL)
 8 : Exit
 Please Select An Above Option: 4

| | WAGECOMPONENT | WAGEVALUES | WAGE_N |
|----|---------------------------------------|------------|--------|
| 0 | Min. Wage for Part A (Unskilled) | 540.00 | US_A |
| 1 | Min. Wage for Part A (Semi Skilled) | 619.00 | SS_A |
| 2 | Min. Wage for Part A (Skilled) | 737.00 | S_A |
| 3 | Min. Wage for Part A (Highly skilled) | 845.00 | HS_A |
| 4 | Min. Wage for Part B (Unskilled) | 511.00 | US_B |
| 5 | Min. Wage for Part B (Semi Skilled) | 602.00 | SS_B |
| 6 | Min. Wage for Part B (Skilled) | 705.00 | S_B |
| 7 | Min. Wage for Part B (Highly skilled) | 804.00 | HS_B |
| 8 | Min. Wage for Part C (Unskilled) | 437.00 | US_C |
| 9 | Min. Wage for Part C (Semi skilled) | 512.00 | SS_C |
| 10 | Min. Wage for Part C (Skilled) | 617.00 | S_C |
| 11 | Min. Wage for Part C (Highly skilled) | 769.00 | HS_C |
| 12 | Bonus | 8.33 | BNS |
| 13 | Gratuity/Retirement Compensation | 4.80 | GRTY |
| 14 | Leave with Wages | 5.77 | LWW |
| 15 | Labour Welfare Fund | 0.20 | LWF |
| 16 | Employee Provident Fund | 12.00 | EPF |
| 17 | Employee State Insurance | 0.75 | ESI |

1 : ADD EMPLOYEE DETAILS
 2 : SHOW EMPLOYEE DETAILS
 3 : FIX STRUCTURAL RATES
 4 : SHOW CURRENT DA AND HRA RATES
 5 : WAGESHEET ENTRY
 6 : SHOW WAGESHEET
 7 : EXPORT WAGESHET (CSV FILE IN EXCEL)
 8 : Exit

Please Select An Above Option: 5
 Enter the salary year (press enter for current year otherwise input new year: 2022)
 Enter the salary month (press enter for current month otherwise input new month:02)
 Enter salary details for the 02/2022
 Employee Code : SPD01

Enter No. of days worked in Shift A : 2
 Enter No. of days worked in Shift B : 6
 Enter No. of days worked in Shift C : 8
 Enter No. of days worked in General Shift : 2
 Enter No. of Over Time worked in Hrs. : 60
 Enter No. of Paid Holidays : 1
 Enter other allowance (or 0): 0
 Enter other deductions (or 0): 0
 ['SPD01', 'Priyanshu', 'Dharmender', '983283438423', '420032111', 'HS', 2, 6, 8, 2, 60, 1, 13842.0, 11535.0, 1153.8380, 664.416, 798.6833999999999, 0, 13842.0, 27993.138, 1661.04, 1038.15, 27.684, 2699.19, 39135.948000000004, '02', '2022']
 1 : ADD EMPLOYEE DETAILS
 2 : SHOW EMPLOYEE DETAILS
 3 : FIX STRUCTURAL RATES
 4 : SHOW CURRENT DA AND HRA RATES
 5 : WAGESHEET ENTRY
 6 : SHOW WAGESHEET
 7 : EXPORT WAGESHET (CSV FILE IN EXCEL)

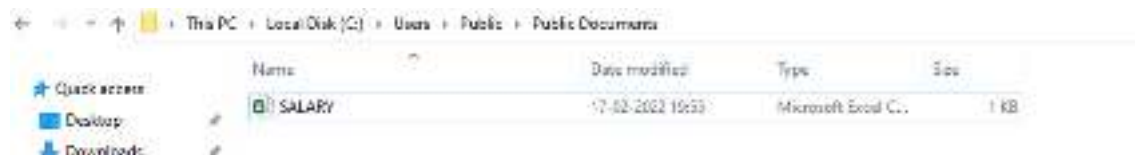

```

8 : Exit
Please Select An Above Option: 6
  EMP_NO      NAME FATHER_NAME ... NET SALARY  WAGE MONTH  YEAR
0 SPD01 Priyanshu Dharmender ... 39135.948           2 2022

[1 rows x 7 columns]
1 : ADD EMPLOYEE DETAILS
2 : SHOW EMPLOYEE DETAILS
3 : FIX STRUCTURAL RATES
4 : SHOW CURRENT DA AND HRA RATES
5 : WAGESHEET ENTRY
6 : SHOW WAGESHEET
7 : EXPORT WAGESHEET (CSV FILE IN EXCEL)
8 : Exit
Please Select An Above Option: 77
Wrong choice.....
1 : ADD EMPLOYEE DETAILS
2 : SHOW EMPLOYEE DETAILS
3 : FIX STRUCTURAL RATES
4 : SHOW CURRENT DA AND HRA RATES
5 : WAGESHEET ENTRY
6 : SHOW WAGESHEET
7 : EXPORT WAGESHEET (CSV FILE IN EXCEL)
8 : Exit
Please Select An Above Option: 7
File Succesfully Generated at Following Path : C:\Users\Public\Documents\Salary.csv
1 : ADD EMPLOYEE DETAILS
2 : SHOW EMPLOYEE DETAILS
3 : FIX STRUCTURAL RATES
4 : SHOW CURRENT DA AND HRA RATES
5 : WAGESHEET ENTRY
6 : SHOW WAGESHEET
7 : EXPORT WAGESHEET (CSV FILE IN EXCEL)
8 : Exit
Please Select An Above Option:

```

CSV File is generated at the following location



FUTURE SCOPE OF THE WORK

- ✓ The option to print the records in portable formats in future.
- ✓ I intend to add various other features like payment gateways in the future.
- ✓ I would like to implement a regular backup mechanism to back up the employee database to avoid disasters.
- ✓ I would like to implement cloud database to enable hassle free data handling.
- ✓ I intend to make it cross-platform so even Android or Mac user can use it efficiently.
- ✓ I also want to implement separate Dashboards for multiple users with different authentication methods.
- ✓ KBC game can be modified to Company Introduction Test to increase awareness about company rules to all employees
- ✓ I also intend to add Training Modules for ease in use.
- ✓ The system can be developed in such a way that its existing features can be modified to better versions.

CONCLUSION

This project is built keeping in mind that it is to be used by only one user that is the admin. It is built for use in small scale organization where the number of employees is limited. According to the requested requirement the admin can add, manipulate, update employee data in his organization. The admin can manage pay structure and update accordingly. The Admin can also add predefined pay grades for the employees. The required records can be easily viewed by the admin anytime time he wants in an instant. The payment of the employee is based on monthly basis. KBC Game has added to create user interest in the software and gain knowledge also. The main objective of this framework is to save time, make the system cost effective and management records efficiently.

BIBLIOGRAPHY

Websites:

- *www.geeksforgeeks.com*
- *www.youtube.com*

Books:

- *Computer Science with Python Class 12- Dhanpat Rai Publications*