Project 1

General Cancer Probability Prediction (GCPP)

Created by:

Alfisyahrin Gusman (2120010016)

Riki Awal Syahputra (2120010136)

Rafie Ananda Suhermawan (2120010109)

Class :

2ISA2

Faculty :

Mhd. Idham Khalif, S.Kom

CEP CCIT

FACULTY OF ENGINEERING UNIVERSITY OF INDONESIA

2022

PROJECT ON

General Cancer Probability Prediction (GCPP)

Developed by

Name:

Alfisyahrin Gusman (2120010016)

Riki Awal Syahputra (2120010136)

Rafie Ananda Suhermawan (2120010109)

FACULTY:

Mhd. Idham Khalif, S.Kom

PROJECT ON

General Cancer Probability Prediction (GCPP)

Batch Code	: 2ISA2
Start Date	: 14st may 2022
End Date	: 16st may 2022
Name Of Faculty	: Mhd. Idham Khalif, S.Kom
Names of Developer	:
	Alfisyahrin Gusman (2120010016)
	Riki Awal Syahputra (2120010136)
	Rafie Ananda Suhermawan (2120010109)
Date of Submission	: 16/05/2022

NIIT

CERTIFICATE

This is to certify that this report titled "General Cancer Probability Prediction (GCPP)" embodies the original work done by Alfisyahrin Gusman, Riki Awal Syahputra, and Rafie Ananda Suhermawan, project in partial fulfillment of their course requirement at NIIT.

Coordinator : Mhd. Idham Khalif, S.Kom

ACKNOWLEDGEMENT

Praise be to God Almighty for giving the author the opportunity to complete this paper. It is because of His grace and guidance that the author was able to complete the paper entitled General Cancer Probability Prediction (GCPP).

GCPP's paper was prepared to fulfill the duties of a lecturer. In addition, the author also hopes that this paper can add insight for readers about GCPP.

The author expresses his deepest gratitude to Mr. Mhd. Idham Khalif, S.Kom as the lecturer of the course. The task that has been given can increase knowledge and insight related to the field occupied by the author. The author would also like to thank all those who have helped in the preparation of this paper.

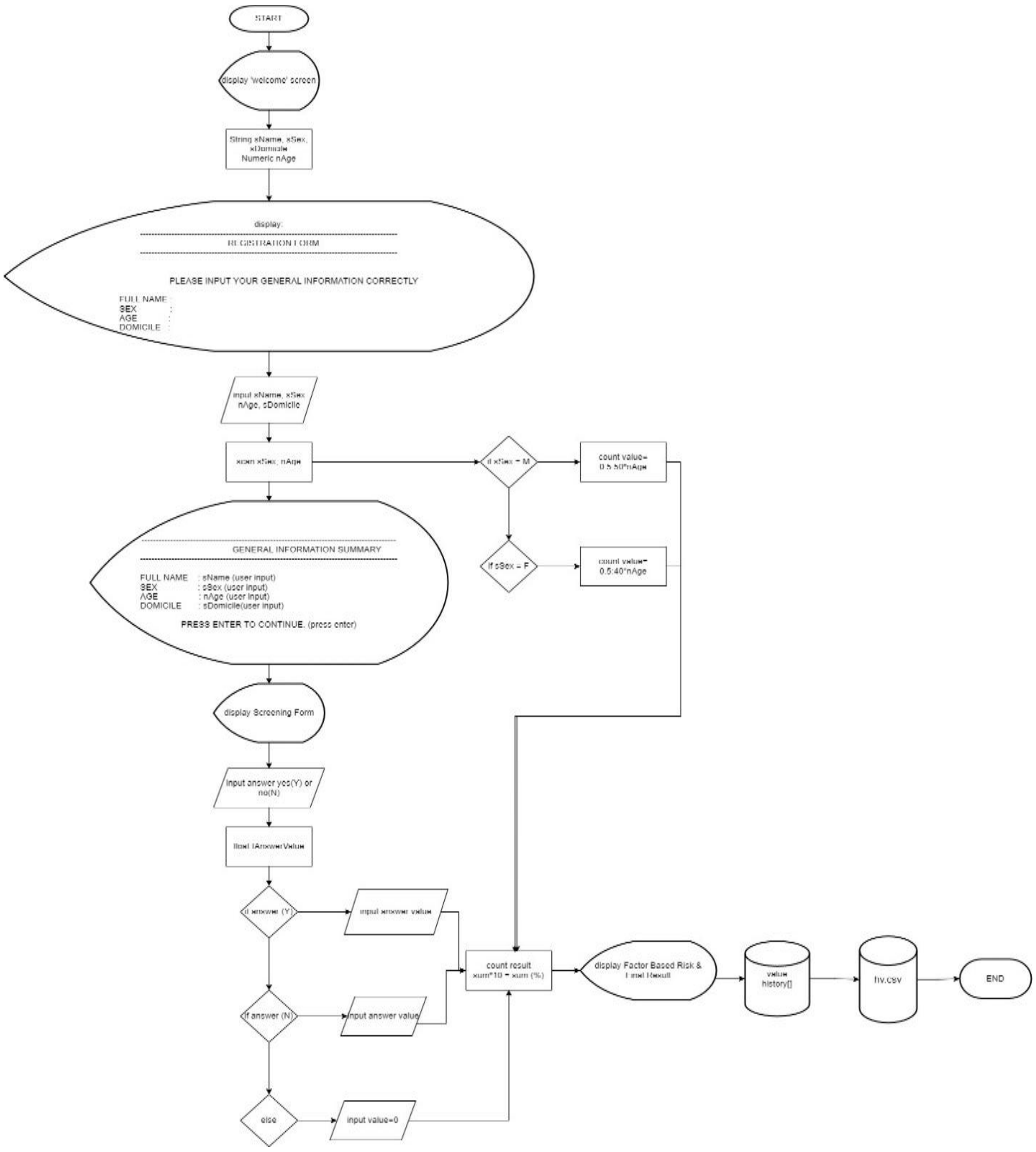
The author realizes that this paper is still far from perfect. Therefore, constructive criticism and suggestions will be accepted by the author for the perfection of this paper.

SYSTEM ANALYSIS

System Summary:

This is a simple command line Python application that can be used to analyze the likelihood of developing cancer symptoms. The questions themselves are written as a class (categorized according to predetermined criteria) to get the probability of getting cancer. Which has several questions such as name, age, gender and address. After that proceed to some general diagnostic questions with each question having a different value. After the question is completed, the score will be saved into a .csv file. This system helps many users to measure the likelihood of being affected by cancer symptoms.

FLOWCHART GCPP



THE CODE

```
from re import X

from time import sleep

from turtle import delay

import os

import csv

from csv import writer

import sys

import time




class ProgressBar(object):

    DEFAULT_BAR_LENGTH = 65

    DEFAULT_CHAR_ON  = '■'

    DEFAULT_CHAR_OFF = ' '


    def __init__(self, end, start=0):

        self.end    = end

        self.start  = start

        self._barLength = self.__class__.DEFAULT_BAR_LENGTH


        self.setLevel(self.start)

        self._plotted = False


    def setLevel(self, level):

        self._level = level

        if level < self.start:  self._level = self.start

        if level > self.end:    self._level = self.end


        self._ratio = float(self._level - self.start) / float(self.end - self.start)

        self._levelChars = int(self._ratio * self._barLength)
    def plotProgress(self):
        sys.stdout.write("\r  %3i%% [%s%s]" %(
            int(self._ratio * 100.0),
            self.__class__.DEFAULT_CHAR_ON  * int(self._levelChars),
            self.__class__.DEFAULT_CHAR_OFF * int(self._barLength - self._levelChars),
        ))
        sys.stdout.flush()
        self._plotted = True

    def setAndPlot(self, level):
        oldChars = self._levelChars
        self.setLevel(level)
        if (not self._plotted) or (oldChars != self._levelChars):
            self.plotProgress()

    def __add__(self, other):
        assert type(other) in [float, int], "can only add a number"
        self.setAndPlot(self._level + other)
        return self
    def __sub__(self, other):
        return self.__add__(-other)
    def __iadd__(self, other):
        return self.__add__(other)
    def __isub__(self, other):
        return self.__add__(-other)

    def __del__(self):
        sys.stdout.write("\n")

class color:
   PURPLE = '\033[95m'
   CYAN = '\033[96m'
   DARKCYAN = '\033[36m'
   BLUE = '\033[94m'
   GREEN = '\033[92m'
   YELLOW = '\033[93m'
   RED = '\033[91m'

   BOLD = '\033[1m'

   UNDERLINE = '\033[4m'

   END = '\033[0m'


class inp:

    #LIST FOR STORING ALL DATA

    val = [] #value, this collect all value from the question and sum the inside

    hv = [] #historical value


    #LIST FOR REGISTRATION DATA

    nm = ['KIRO']

    sex = ['M']

    age = [17]

    dm = ['JKT']


    #LIST FOR STORING VALUE IN EACH CATEGORY OF QUESTION

    agec = []

    dsc = []

    ls = []

    job = []

    env = []

    vc = []

    dss = []

    inf = []


    #LIST FOR STORING ALL INPUT DATA IN EACH CATEGORY QUESTION TO SUM IT UP

    xls = []

    xjob = []

    xvc = []

    xinf = []


    def _nm(self,x):

        self.x = x

        inp.nm.append(x)

    def _sex(self,x):
        self.x = x
        inp.sex.append(x)
    def _age(self,x):
        self.x = x
        inp.age.append(int(x))
    def _dm(self,x):
        self.x = x
        inp.dm.append(x)

    def _agec(self,x):
        self.x = x
        inp.agec.append(x)
    def _dsc(self,x):
        self.x = x
        inp.dsc.append(x)
    def _ls(self,x):
        self.x = x
        inp.ls.append(x)
    def _job(self,x):
        self.x = x
        inp.job.append(float(x))
    def _env(self,x):
        self.x = x
        inp.env.append(x)

    def _vc(self,x):
        self.x = x
        inp.vc.append(x)
    def _dss(self,x):
        self.x = x
        inp.dss.append(x)
    def _inf(self,x):
        self.x = x
        inp.inf.append(x)

    def _val(self,x):
        self.x = x
        inp.val.append(float(x))
    def _hv(self,x):
        self.x = x
        inp.hv.append(x)

def animate(a,count):

    print (a)
    pb = ProgressBar(count)

    #pb.plotProgress()
    for i in range(0, count):
        pb += 1
        #pb.setAndPlot(i + 1)
        time.sleep(0.01)
    del pb

def animate1(n,t):

    # String to be displayed when the application is loading
    load_str = n
    ls_len = len(load_str)

    # String for creating the rotating line
    animation = "|/-\\"
    anicount = 0

    # used to keep the track of
    # the duration of animation
    counttime = 0

    # pointer for travelling the loading string
    i = 0

    while (counttime != t):
```

```
# used to change the animation speed
# smaller the value, faster will be the animation
time.sleep(0.075)

# converting the string to list
# as string is immutable
load_str_list = list(load_str)

# x->obtaining the ASCII code
x = ord(load_str_list[i])

# y->for storing altered ASCII code
y = 0

# if the character is "." or " ", keep it unaltered
# switch uppercase to lowercase and vice-versa
if x != 32 and x != 46:
    if x>90:
        y = x-32
    else:
        y = x + 32
    load_str_list[i]= chr(y)

# for storing the resultant string
res =''
for j in range(ls_len):
    res = res + load_str_list[j]

# displaying the resultant string
sys.stdout.write("\r"+res + animation[anicount])
sys.stdout.flush()

# Assigning loading string
# to the resultant string
load_str = res
```

```
        anicount = (anicount + 1)% 4

        i =(i + 1)% ls_len

        counttime = counttime + 1


    # for windows OS

    if os.name =="nt":

        os.system("cls")


    # for linux / Mac OS

    else:

        os.system("clear")




def wellcome():


    print('\t')

    print('▪'*90)

    print('\t')

    print(color.BLUE+color.BOLD+'\t\tGENERAL CANCER PROBABILITY PREDICTION SYSTEM (GCPPS)'+
color.END)

    print('\t')

    print('▪'*90)

    sleep(1)

    print('\t'*2)

    sleep(1)

    print(color.RED+'\t\t\t\t READ IT CAREFULLY! '+color.END)

    print('\t'*2)

    print('''

    GCPP SYSTEM IS ONLY INTENDED FOR PREDICTION OF COVERING POTENTIAL CANCER RISK IN

    GENERAL (NOT SPECIFIC) AND NOT FOR DIAGNOSIS, REFERENCE DATA COVERED THROUGH

    cancer.net, halodoc.com, AND ATSDR (Agency for Toxic Disease Registry), FOR MORE

    INFORMATION AND DISEASE REGISTRATION FURTHER, PLEASE CONTACT YOUR

    DOCTOR OR EXPERT IN CANCER

    ''')

    sleep(2)
    print('\t'*2)
    print('▪'*90)
    print('\t'*2)
    print(color.UNDERLINE+'\t\t\t   TEST YOUR RISK BEFORE IT LATE   '+color.END)
    print('\t'*2)

    input("\t\t       PRESS ENTER TO CONTINUE. (press enter)")

def regist():

    print('\t')
    print('▪'*90)
    print(color.GREEN+color.BOLD+'\t\t\t\t    REGISTRATION FORM'+ color.END)
    print('▪'*90)
    sleep(1)
    print('\t')
    sleep(1)
    print('\t'*2)
    print('\t\t     PLEASE INPUT YOUR GENERAL INFORMATION CORRECTLY')
    print('\t'*2)

    nm = input('() NAME        : ')
    sex = input('() SEX (M/F)   : ')
    age = input('() AGE         : ')
    dm = input('() DOMICILE    : ')

    inp._nm(inp,nm)
    inp._sex(inp,sex)
    inp._age(inp,age)
    inp._dm(inp,dm)

    #AGE

    if sex == 'M' or sex == 'm':
        agec = (0.534782608695651/50)*float(age)
        inp._val(inp,agec)
        inp._agec(inp,agec)
    elif sex == 'F' or sex == 'f':
        agec = (0.534782608695651/40)*float(age)
        inp._val(inp,agec)
        inp._agec(inp,agec)
    else:
        inp._val(inp,0)
        inp._agec(inp,0)

    print('\t')
    animate('\t\t\t      PROCESSING YOUR REGISTRATION',200)



    os.system('cls||clear')
    print('\t')
    print('-'*90)
    print(color.BOLD+'\t\t\t\tGENERAL INFORMATION SUMMARY'+ color.END)
    print('-'*90)

    sleep(1)
    print('\t')
    print('▪ FULL NAME    :', inp.nm[-1])
    sleep(0.5)
    print('▪ SEX          :', inp.sex[-1])
    sleep(0.5)
    print('▪ AGE          :', inp.age[-1])
    sleep(0.5)
    print('▪ DOMICILE     :', inp.dm[-1])
    print('\t')
    sleep(1)
    input("PRESS ENTER TO CONTINUE. (press enter)")

def screen():

    print('\t')
    print('-'*90)
    print(color.CYAN+color.BOLD+'\t\t\t\t     SCREENING FORM'+ color.END)

    print('.'*90)

    sleep(1)

    print('\t')

    sleep(1)

    print('\t'*2)

    print('\t\t    PLEASE INPUT YOUR CONDITION INFORMATION CORRECTLY')

    print('\t')

    print(color.GREEN+color.BOLD+'\t\t\t              [Y] FOR YES'+ color.END)


    print(color.RED+color.BOLD+'\t\t\t              [N] FOR NO'+ color.END)

    print('\t'*2)



    dsc = input('(1) HAVE ANY FAMILY MEMBERS OR GENERATIONS BEFORE YOU COLLECTED CANCER? (Y/N)
[')


    ls1 = input('(2) DO YOU SMOKE? (Y/N) [')

    ls2 = input('(3) DO YOU CONSUME ALCOHOL? (Y/N) [')

    ls3 = input('(4) DO YOU WORK IN A UV RADIATION SCOPE (NO PROTECTIVE)? (Y/N) [')

    ls4 = input('(5) DURING THE DAY DO YOU USE A JACKET WHEN OUTSIDE? (Y/N) [')

    ls5 = input('(6) DURING THE DAY DO YOU WEAR TROUSERS WHEN OUTDOORS? (Y/N) [')

    ls6 = input('(7) DURING THE DAY DO YOU USE A HAT WHEN OUTSIDE? (Y/N) [')

    ls7 = input('(8) DO YOU USE SUN SCREEN OUTSIDE? (Y/N) [')

    ls8 = input('(9) ARE YOU KEEPING YOUR WEIGHT IDEAL? (Y/N) [')

    ls9 = input('(10) DO YOU EXERCISE ROUTINELY? (Y/N) [')


    job1 = input('(11) DO YOU WORK IN URANIUM MINING? (Y/N) [')

    job2 = input('(12) DO YOU WORK IN THE NUCLEAR INDUSTRY? (Y/N) [')

    job3 = input('(13) DO YOU WORK IN THE CHEMICAL INDUSTRY? (Y/N) [')

    job4 = input('(14) DO YOU WORK IN SHIP BUILDING? (Y/N) [')




    print ('(15) IN YOUR DAILY ACTIVITIES, ARE YOU FREQUENTLY EXPOSED TO ANY OF THE FOLLOWING
CHEMICALS')

    env = input('     (benzene, beryllium, asbestos, vinyl chloride or arsenic)? (Y/N) [')
    print('\t')
    vc1 = input('(16) HAVE YOU EVER TAKEN THE HEPATITIS B VACCINATION? (Y/N) [')
    vc2 = input('(17) HAVE YOU EVER TAKEN THE HPV VACCINATION? (Y/N) [')

    dss = input('(18) DO YOU HAVE OBESITY? (Y/N) [')

    inf1 = input('(19) ARE YOU SUFFERING WITH A DISEASE CAUSED BY THE HPV VIRUS? (Y/N) [')
    inf2 = input('(20) DO YOU SUFFER A DISEASE CAUSED BY THE HEPATITIS B OR C VIRUS? (Y/N) [')
    inf3 = input('(21) ARE YOU SUFFERING WITH A DISEASE CAUSED BY THE EPSTEIN-BARR VIRUS? (Y/N)
[')
    inf4 = input('(22) ARE YOU SUFFERING WITH A DISEASE CAUSED BY H. PYLORI BACTERIA? (Y/N) [')



    #DSC
    if dsc == 'y' or dsc == 'Y':
        inp._val(inp,0.544782608695652)
        inp._dsc(inp,0.544782608695652)
    else:
        inp._val(inp,0)
        inp._dsc(inp,0)



    #LS
    if ls1 == 'y' or ls1 == 'Y':
        inp._val(inp,0.424782608695651)
        inp.xls.append(0.424782608695651)
    else:
        inp._val(inp,0)
        inp.xls.append(0)

    if ls2 == 'y' or ls2 == 'Y':
        inp._val(inp,0.414782608695651)
        inp.xls.append(0.414782608695651)
    else:
```

```
    inp._val(inp,0)

    inp.xls.append(0)


if ls3 == 'y' or ls3 == 'Y':

    inp._val(inp,0.404782608695651)

    inp.xls.append(0.404782608695651)

else:

    inp._val(inp,0)

    inp.xls.append(0)


if ls4 == 'n' or ls4 == 'N':

    inp._val(inp,0.394782608695651)

    inp.xls.append(0.394782608695651)

else:

    inp._val(inp,0)

    inp.xls.append(0)


if ls5 == 'n' or ls5 == 'N':

    inp._val(inp,0.384782608695651)

    inp.xls.append(0.384782608695651)

else:

    inp._val(inp,0)

    inp.xls.append(0)


if ls6 == 'n' or ls6 == 'N':

    inp._val(inp,0.374782608695651)

    inp.xls.append(0.374782608695651)

else:

    inp._val(inp,0)

    inp.xls.append(0)


if ls7 == 'n' or ls7 == 'N':

    inp._val(inp,0.364782608695651)

    inp.xls.append(0.364782608695651)

else:

    inp._val(inp,0)
    inp.xls.append(0)

if ls8 == 'n' or ls8 == 'N':
    inp._val(inp,0.354782608695651)
    inp.xls.append(0.354782608695651)
else:
    inp._val(inp,0)
    inp.xls.append(0)

if ls9 == 'n' or ls9 == 'N':
    inp._val(inp,0.344782608695651)
    inp.xls.append(0.344782608695651)
else:
    inp._val(inp,0)
    inp.xls.append(0)

xum = sum(inp.xls)
inp._ls(inp,xum)

#JOB

if job1 == 'y' or job1 == 'Y':
    inp._val(inp,0.524782608695651)
    inp.xjob.append(0.524782608695651)
else:
    inp._val(inp,0)
    inp.xjob.append(0)

if job2 == 'y' or job2 == 'Y':
    inp._val(inp,0.514782608695651)
    inp.xjob.append(0.514782608695651)
else:
    inp._val(inp,0)
    inp.xjob.append(0)

if job3 == 'y' or job3 == 'Y':
    inp._val(inp,0.504782608695651)

    inp.xjob.append(0.504782608695651)

else:

    inp._val(inp,0)

    inp.xjob.append(0)


if job4 == 'y' or job4 == 'Y':

    inp._val(inp,0.494782608695651)

    inp.xjob.append(0.494782608695651)

else:

    inp._val(inp,0)

    inp.xjob.append(0)


xum = sum(inp.xjob)

inp._job(inp,xum)


#ENV


if env == 'y' or env == 'Y':

    inp._val(inp,0.484782608695651)

    inp._env(inp,0.484782608695651)

else:

    inp._val(inp,0)

    inp._env(inp,0)


#VC


if vc1 == 'n' or vc1 == 'N':

    inp._val(inp,0.334782608695651)

    inp.xvc.append(0.334782608695651)

else:

    inp._val(inp,0)

    inp.xvc.append(0)


if vc2 == 'n' or vc2 == 'N':

    inp._val(inp,0.324782608695651)
    inp.xvc.append(0.324782608695651)
else:
    inp._val(inp,0)
    inp.xvc.append(0)

xum = sum(inp.xvc)
inp._vc(inp,xum)

#DSS

if dss == 'y' or dss == 'Y':
    inp._val(inp,0.434782608695651)
    inp._dss(inp,0.434782608695651)
else:
    inp._val(inp,0)
    inp._dss(inp,0)

#INF

if inf1 == 'y' or inf1 == 'Y':
    inp._val(inp,0.474782608695651)
    inp.xinf.append(0.474782608695651)
else:
    inp._val(inp,0)
    inp.xinf.append(0)

if inf2 == 'y' or inf2 == 'Y':
    inp._val(inp,0.464782608695651)
    inp.xinf.append(0.464782608695651)
else:
    inp._val(inp,0)
    inp.xinf.append(0)

if inf3 == 'y' or inf3 == 'Y':
    inp._val(inp,0.454782608695651)
    inp.xinf.append(0.454782608695651)
```

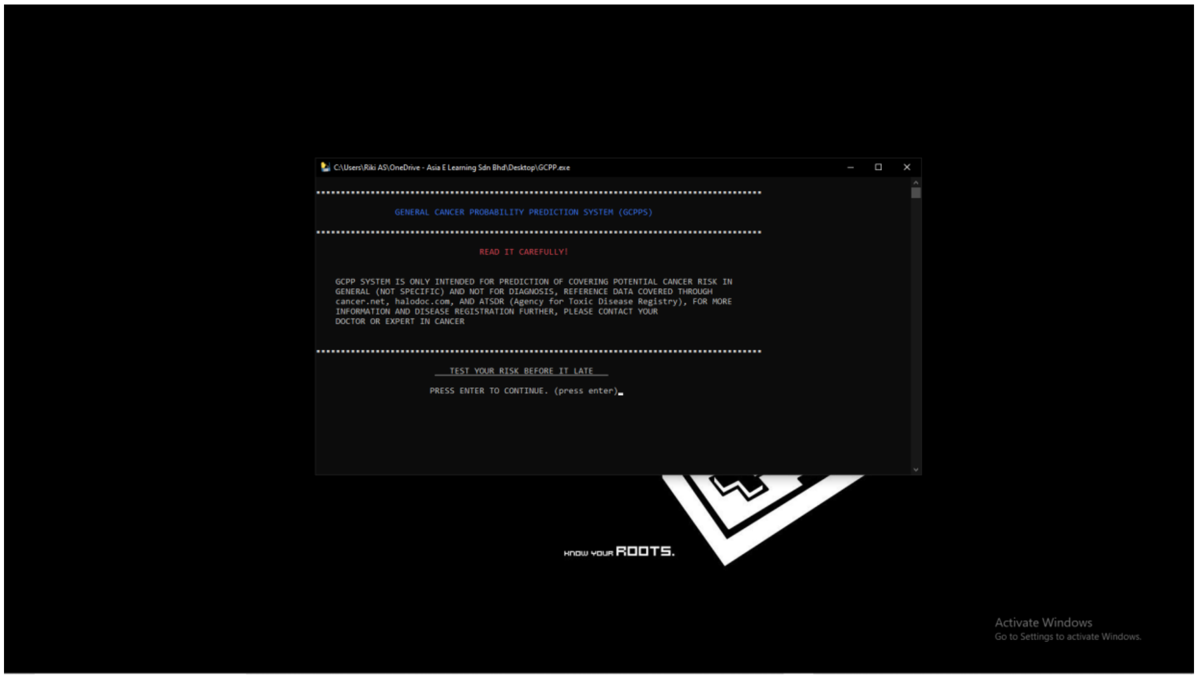
```
else:
    inp._val(inp,0)
    inp.xinf.append(0)

if inf4 == 'y' or inf4 == 'Y':
    inp._val(inp,0.444782608695651)
    inp.xinf.append(0.444782608695651)
else:
    inp._val(inp,0)
    inp.xinf.append(0)

xum = sum(inp.xinf)
inp._inf(inp,xum)

#PRINT FOR CEK UP-----------------------------------------------------
# print('\t')
# print('-'*90)
# print('\t')
# print('VALUE DATA       : ',inp.val)
# print('\t')
# print('-'*90)
# print('\t')
# zum = sum(inp.val)
# print('TOTAL VALUE      : ',zum)
# print('\t')
# print('-'*90)
# print('\t')
# inp._hv(inp,zum)
# print('HISTORICAL VALUE : ',inp.hv)
# print('\t')
# print('-'*90)
# print('\t')
# print('AGEC VALUE   : ',inp.agec)
# print('DSC VALUE    : ',inp.dsc)
# print('LS VALUE     : ',inp.ls)
# print('JOB VALUE    : ',inp.job)
```

```
    # print('ENV VALUE    : ',inp.env)
    # print('VC VALUE     : ',inp.vc)
    # print('DSS VALUE    : ',inp.dss)
    # print('INF VALUE    : ',inp.inf)
    # ---------------------------------------------------------
def res():

    print('\t')
    animate1("learning your life pattern..",70)

    print('\t')
    print('-'*90)
    print(color.BOLD+'\t\t\t\tANALYZING THE PATTERN'+ color.END)
    print('-'*90)

    print('\t')
    animate('\t☕ ACCESSING YOUR PROFILE', 200)
    print('\t')
    animate('\t☕ CALCULATING YOUR FBR SCORE', 70)
    print('\t')
    animate('\t☕ CALCULATING YOUR CRS SCORE', 250)
    print('\t')
    animate('\t☕ SAVING YOUR DATA TO DATABASE', 300)
    print('\t')
    animate('\t☕ ACCESSING YOUR DATA', 200)
    print('\t')

    zum = sum(inp.val)*10
    fres = '{}{}'.format(zum,'%')
    inp._hv(inp,fres)

    sleep(0.5)
    print('\t')
    print('-'*90)
    sleep(0.5)
    print(color.BOLD+'\t\t\t\t\tGCPP RESULT'+ color.END)
    print('-'*90)


    sleep(0.5)

    print('\t')

    print('[] FULL NAME    :', inp.nm[-1])

    sleep(0.4)

    print('[] SEX          :', inp.sex[-1])

    sleep(0.3)

    print('[] AGE          :', inp.age[-1])

    sleep(0.2)

    print('[] DOMICILE     :', inp.dm[-1])

    sleep(1)

    print('\t'*2)


    print(color.BOLD+color.UNDERLINE+'\t\t\t F A C T O R   B A S E D   R I S K '+color.END)

    print('\t')

    sleep(1)

    print('[] AGE          : ',str(inp.agec[-1]*10)+'%')

    sleep(0.5)

    print('[] GENETIC      : ',str(inp.dsc[-1]*10)+'%')

    sleep(0.5)

    print('[] LIFE STYLE   : ',str(inp.ls[-1]*10)+'%')

    sleep(0.5)

    print('[] JOB          : ',str(inp.job[-1]*10)+'%')

    sleep(0.5)

    print('[] ENVIRONMENT  : ',str(inp.env[-1]*10)+'%')

    sleep(0.5)

    print('[] VACCINATION  : ',str(inp.vc[-1]*10)+'%')

    sleep(0.5)

    print('[] DESEASE      : ',str(inp.dss[-1]*10)+'%')

    sleep(0.5)

    print('[] INFECTION    : ',str(inp.inf[-1]*10)+'%')

    sleep(1.5)

    print('\t')


    print(color.YELLOW+color.BOLD+color.UNDERLINE+'\t\t\t C A N C E R   R I S K   S C O R E
'+color.END)
```

```
    sleep(2)

    print('\t')

    print('\t\t\t\t',fres)

    print('\t')

    print('.'*90)


def dbs():


    # The data assigned to the list

    list_data=[inp.nm[-1], inp.sex[-1], inp.age[-1], inp.dm[-1], inp.agec[-1], inp.dsc[-1],
inp.ls[-1], inp.job[-1], inp.env[-1], inp.vc[-1], inp.dss[-1], inp.inf[-1], inp.agec[-1]*10,
inp.dsc[-1]*10, inp.ls[-1]*10, inp.job[-1]*10, inp.env[-1]*10, inp.vc[-1]*10, inp.dss[-1]*10,
inp.inf[-1]*10, inp.hv[-1]]

    header = ['Name', 'Branch', 'Year', 'CGPA']


    f = open('HV.csv', 'a')

    # xy = csv.DictWriter(f, fieldnames = header)

    # xy.writeheader()


    with f:

        f = open('HV.csv', 'a', newline='')

        writer_object = writer(f)

        writer_object.writerow(list_data)

        # Close the file object

        f.close()




def start():


    os.system('cls||clear')

    wellcome()

    os.system('cls||clear')

    regist()

    os.system('cls||clear')

    screen()

    os.system('cls||clear')

    res()

    dbs()
    print("\t\t         THIS IS THE END OF THE SYSTEM ")

    dcc = input("\t  SELECT YOUR DECISION [L : RE-START SYSTEM | LL : END SYSTEM] : [")

    if dcc == 'L' or dcc == 'l':

        start()

    if dcc == 'LL' or dcc == 'll':

        exit()


start()
```

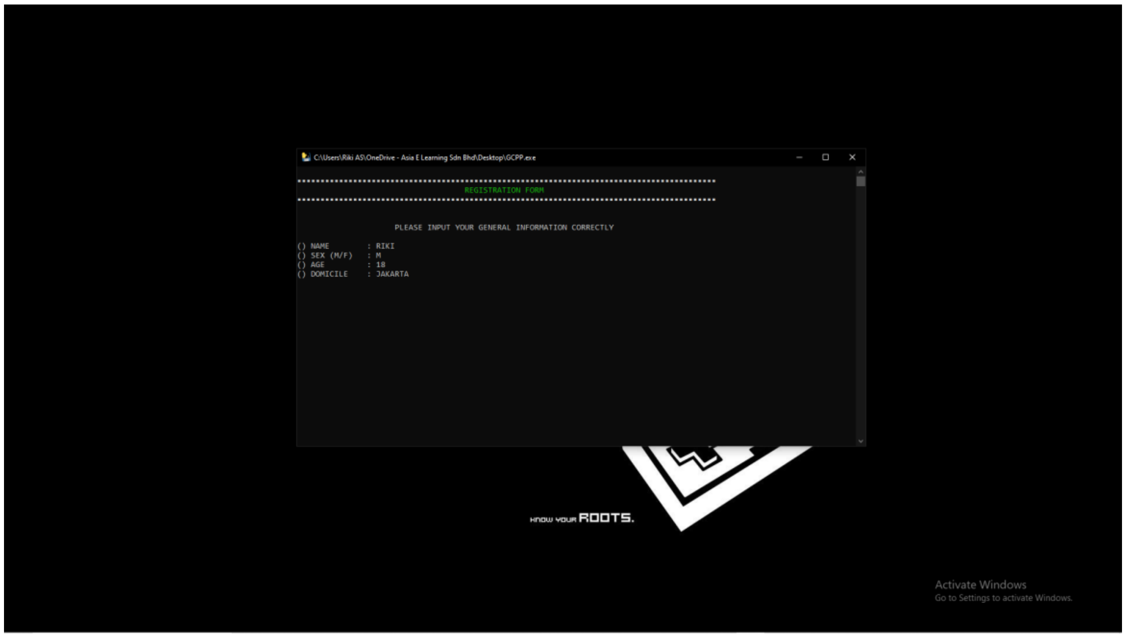
OUTPUT

A. MAIN MENU SCREEN


C:\Users\Riki AS\OneDrive - Asia E Learning Sdn Bhd\Desktop\GCPP.exe
KNOW YOUR ROOTS.
Activate Windows
Go to Settings to activate Windows.

```
********************************************************************************

              GENERAL CANCER PROBABILITY PREDICTION SYSTEM (GCPPS)

********************************************************************************

                               READ IT CAREFULLY!


   GCPP SYSTEM IS ONLY INTENDED FOR PREDICTION OF COVERING POTENTIAL CANCER RISK IN
   GENERAL (NOT SPECIFIC) AND NOT FOR DIAGNOSIS, REFERENCE DATA COVERED THROUGH
   cancer.net, halodoc.com, AND ATSDR (Agency for Toxic Disease Registry), FOR MORE
   INFORMATION AND DISEASE REGISTRATION FURTHER, PLEASE CONTACT YOUR
   DOCTOR OR EXPERT IN CANCER


********************************************************************************

                    ___TEST YOUR RISK BEFORE IT LATE___

                   PRESS ENTER TO CONTINUE. (press enter)_
```

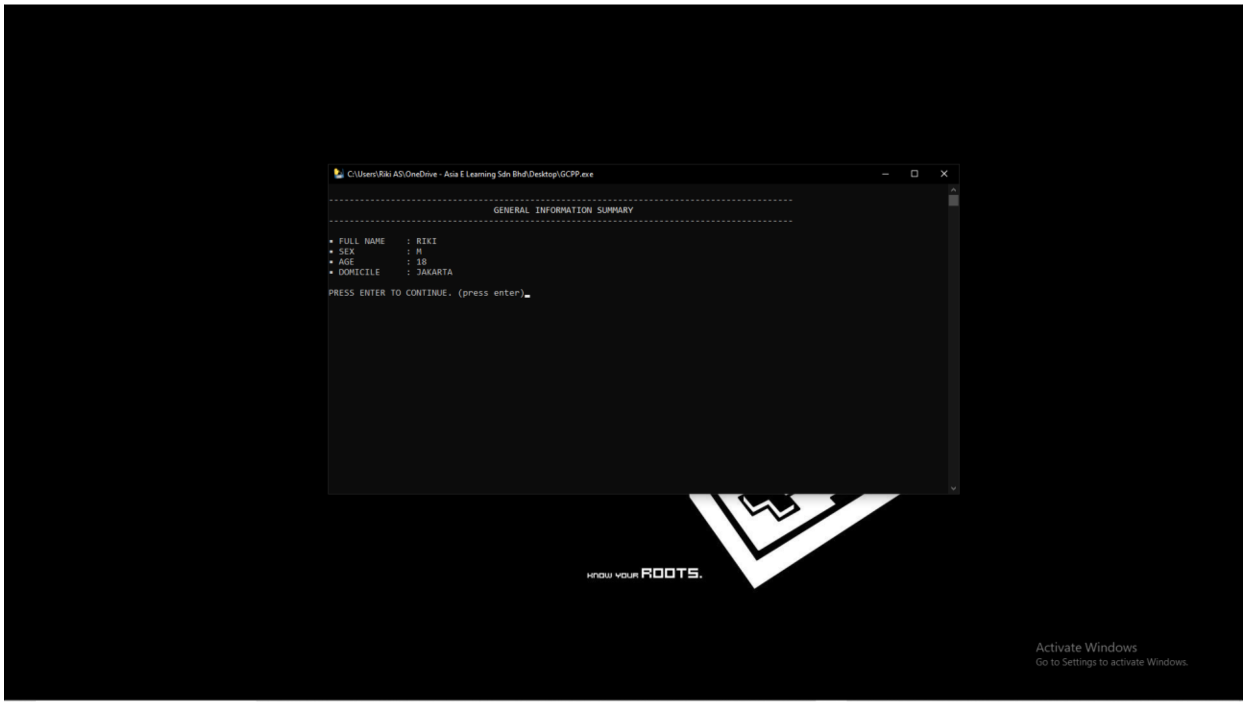
B. REGISTRATION SCREEN


C:\Users\Riki AS\OneDrive - Asia E Learning Sdn Bhd\Desktop\GCPP.exe
KNOW YOUR ROOTS.
Activate Windows
Go to Settings to activate Windows.

```
********************************************************************************
                              REGISTRATION FORM
********************************************************************************


                  PLEASE INPUT YOUR GENERAL INFORMATION CORRECTLY

() NAME        : RIKI
() SEX (M/F)   : M
() AGE         : 18
() DOMICILE    : JAKARTA
```

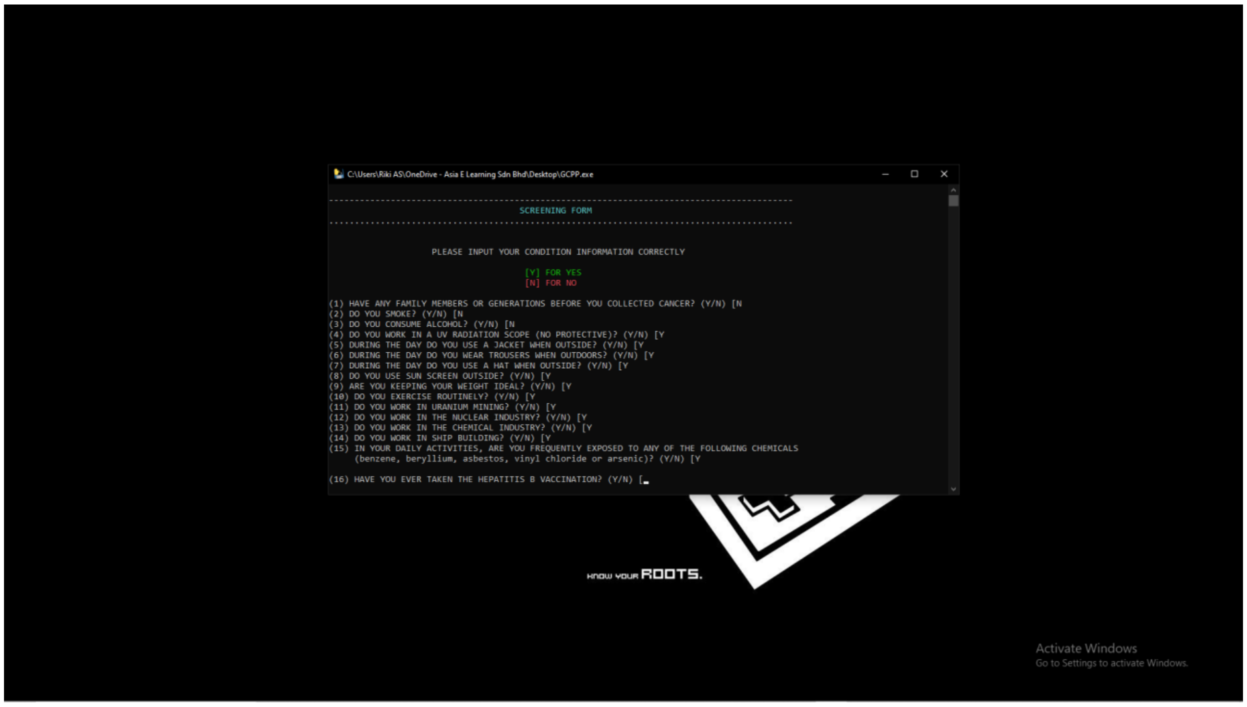
OUTPUT

C. VIEW AFTER REGISTRATION



```
C:\Users\Riki AS\OneDrive - Asia E Learning Sdn Bhd\Desktop\GCPP.exe

-----------------------------------------------------------------------------------------
                              GENERAL INFORMATION SUMMARY
-----------------------------------------------------------------------------------------

▪ FULL NAME     : RIKI
▪ SEX           : M
▪ AGE           : 18
▪ DOMICILE      : JAKARTA

PRESS ENTER TO CONTINUE. (press enter)_
```

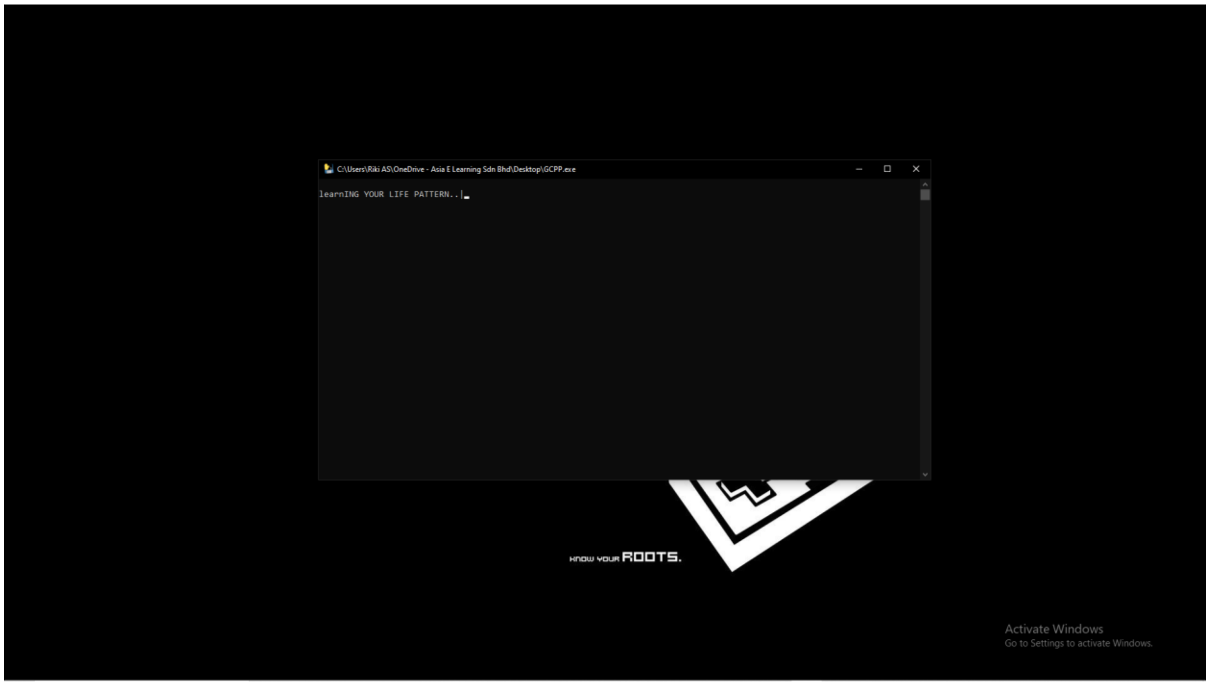
D. COMPLAINT CHECKING PROCESS



```
C:\Users\Riki AS\OneDrive - Asia E Learning Sdn Bhd\Desktop\GCPP.exe

-----------------------------------------------------------------------------------------
                                      SCREENING FORM
.........................................................................................


                    PLEASE INPUT YOUR CONDITION INFORMATION CORRECTLY

                                      [Y] FOR YES
                                      [N] FOR NO

(1) HAVE ANY FAMILY MEMBERS OR GENERATIONS BEFORE YOU COLLECTED CANCER? (Y/N) [N
(2) DO YOU SMOKE? (Y/N) [N
(3) DO YOU CONSUME ALCOHOL? (Y/N) [N
(4) DO YOU WORK IN A UV RADIATION SCOPE (NO PROTECTIVE)? (Y/N) [Y
(5) DURING THE DAY DO YOU USE A JACKET WHEN OUTSIDE? (Y/N) [Y
(6) DURING THE DAY DO YOU WEAR TROUSERS WHEN OUTDOORS? (Y/N) [Y
(7) DURING THE DAY DO YOU USE A HAT WHEN OUTSIDE? (Y/N) [Y
(8) DO YOU USE SUN SCREEN OUTSIDE? (Y/N) [Y
(9) ARE YOU KEEPING YOUR WEIGHT IDEAL? (Y/N) [Y
(10) DO YOU EXERCISE ROUTINELY? (Y/N) [Y
(11) DO YOU WORK IN URANIUM MINING? (Y/N) [Y
(12) DO YOU WORK IN THE NUCLEAR INDUSTRY? (Y/N) [Y
(13) DO YOU WORK IN THE CHEMICAL INDUSTRY? (Y/N) [Y
(14) DO YOU WORK IN SHIP BUILDING? (Y/N) [Y
(15) IN YOUR DAILY ACTIVITIES, ARE YOU FREQUENTLY EXPOSED TO ANY OF THE FOLLOWING CHEMICALS
     (benzene, beryllium, asbestos, vinyl chloride or arsenic)? (Y/N) [Y

(16) HAVE YOU EVER TAKEN THE HEPATITIS B VACCINATION? (Y/N) [_
```

OUTPUT

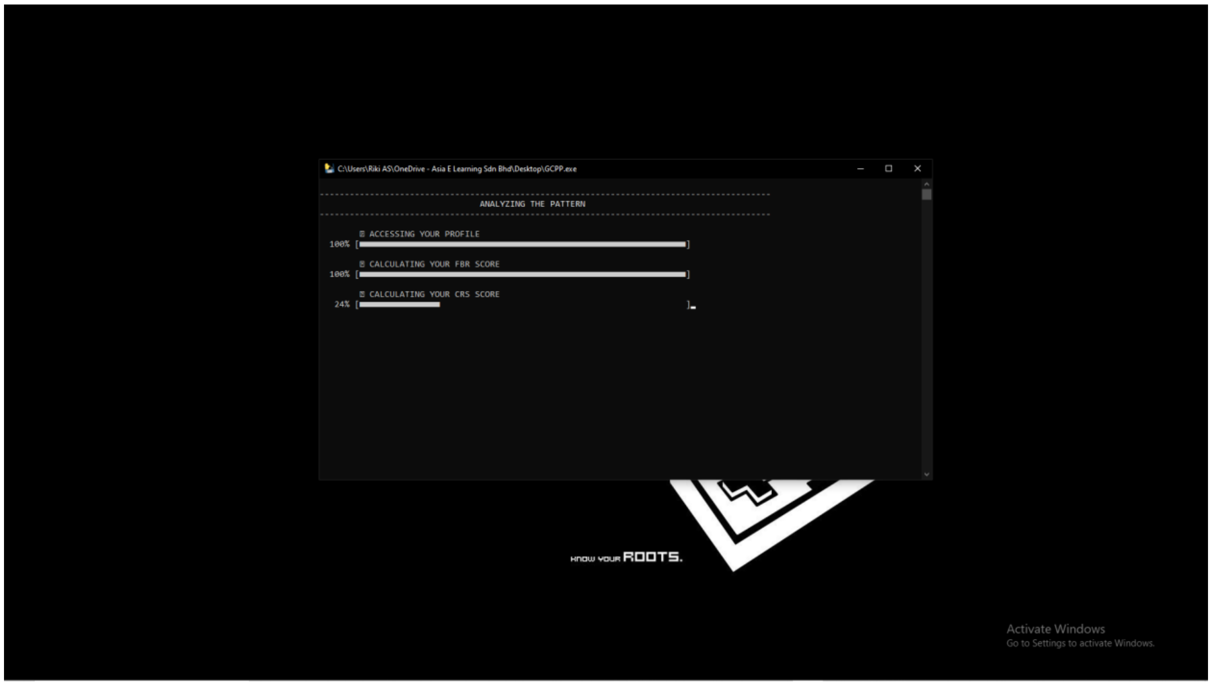
E. CHECKING PROCESS



C:\Users\Riki AS\OneDrive - Asia E Learning Sdn Bhd\Desktop\GCPP.exe

```
learnING YOUR LIFE PATTERN..|
```

F.

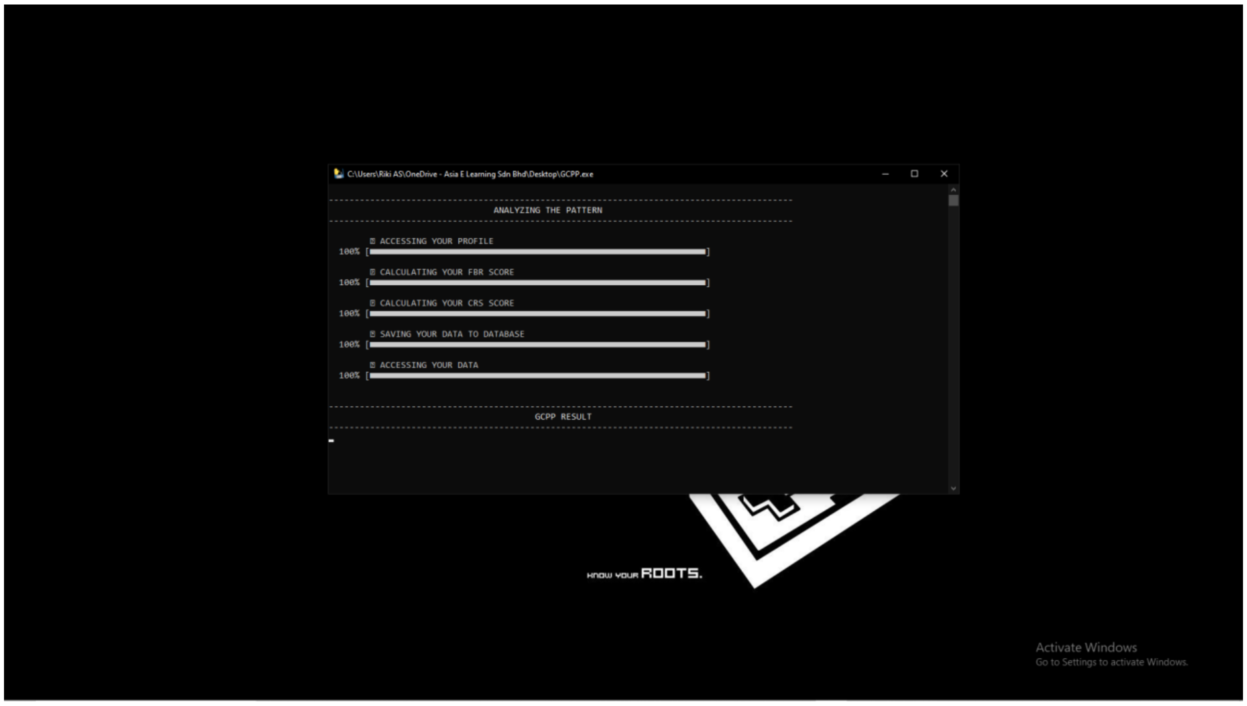


C:\Users\Riki AS\OneDrive - Asia E Learning Sdn Bhd\Desktop\GCPP.exe

```
-----------------------------------------------------------------------------
                              ANALYZING THE PATTERN
-----------------------------------------------------------------------------

       ▦ ACCESSING YOUR PROFILE
 100% [████████████████████████████████████████████████████████████████]

       ▦ CALCULATING YOUR FBR SCORE
 100% [████████████████████████████████████████████████████████████████]

       ▦ CALCULATING YOUR CRS SCORE
  24% [████████████████                                                ]
```

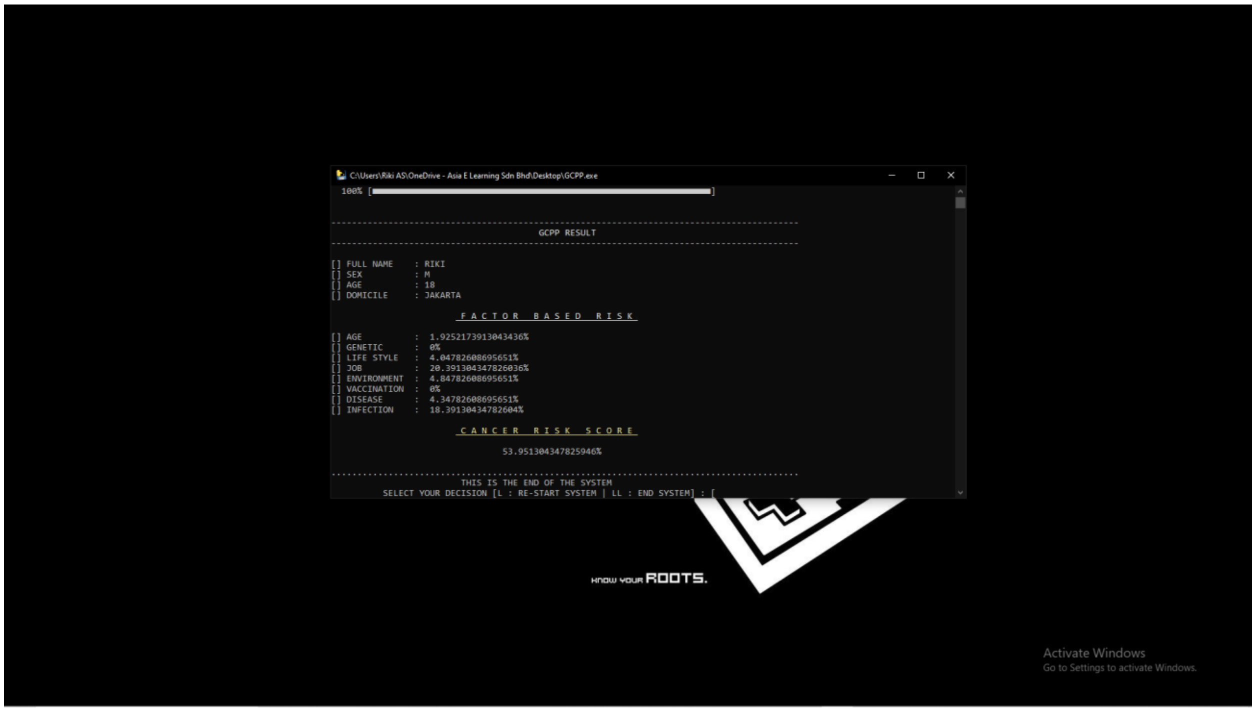
OUTPUT

G.



```
C:\Users\Riki AS\OneDrive - Asia E Learning Sdn Bhd\Desktop\GCPP.exe
--------------------------------------------------------------------------------
                         ANALYZING THE PATTERN
--------------------------------------------------------------------------------

      ⊠ ACCESSING YOUR PROFILE
 100% [████████████████████████████████████████████████████████]

      ⊠ CALCULATING YOUR FBR SCORE
 100% [████████████████████████████████████████████████████████]

      ⊠ CALCULATING YOUR CRS SCORE
 100% [████████████████████████████████████████████████████████]

      ⊠ SAVING YOUR DATA TO DATABASE
 100% [████████████████████████████████████████████████████████]

      ⊠ ACCESSING YOUR DATA
 100% [████████████████████████████████████████████████████████]

--------------------------------------------------------------------------------
                              GCPP RESULT
--------------------------------------------------------------------------------
_
```

H. OVERALL DIAGNOSIS



```
C:\Users\Riki AS\OneDrive - Asia E Learning Sdn Bhd\Desktop\GCPP.exe
 100% [████████████████████████████████████████████████████████]

--------------------------------------------------------------------------------
                              GCPP RESULT
--------------------------------------------------------------------------------

[] FULL NAME    : RIKI
[] SEX          : M
[] AGE          : 18
[] DOMICILE     : JAKARTA

                  F A C T O R   B A S E D   R I S K

[] AGE          :  1.9252173913043436%
[] GENETIC      :  0%
[] LIFE STYLE   :  4.04782608695651%
[] JOB          :  20.391304347826036%
[] ENVIRONMENT  :  4.84782608695651%
[] VACCINATION  :  0%
[] DISEASE      :  4.34782608695651%
[] INFECTION    :  18.39130434782604%

                  C A N C E R   R I S K   S C O R E

                         53.951304347825946%

................................................................................
                  THIS IS THE END OF THE SYSTEM
        SELECT YOUR DECISION [L : RE-START SYSTEM | LL : END SYSTEM] : [
```

CONFIGURATION

Hardware : Intel I7 gen 11, RAM 16 GB HDD 100 GB or 1 TB, Video graphicNVidia GeForce RTX 3080

VGA Resolution : 7680x4320

Operating System : Windows 11

Software : Command Prompt

Microsoft Word 2021

Microsoft Excel 2021

PROBLEM ENCOUNTRED

Our team is facing a time managements challenge in this project which is caused by the re-scheduling of the education schedule, the developer also faces a lack of team communication and meeting also supported by our own busyness, but with this problem, the developer doesn’t stop the movement of this project, and successfully to fix this problem as our challenge.