**Index**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Content** | **Pg. No.** |
| 1 | Acknowledgement | 2 |
| 2 | Overview | 3 |
| 3 | Modules | 4 |
| 4 | Functions | 5 |
| 5 | Files | 9 |
| 6 | SQL Table | 12 |
| 7 | SQL Connectivity with Python | 14 |
| 8 | Program | 16 |
| 9 | Output | 43 |
| 10 | Bibliography | 49 |

**Acknowledgement**

I would like to extend my sincere and heartfelt obligation towards all those who have helped me in making this project. Without their active guidance, help, cooperation, and encouragement, I would not have been able to present the project on time.

I am extremely thankful and pay my sincere gratitude to my teacher, Mrs. Vinita John, for her valuable guidance and support for completion of this project.

Of course, this project would not have been completed without the help, support and cooperation of my group members and friends.

I would also like to appreciate my friends and team members for their valuable suggestions given to me in completing the project.

I also acknowledge with a deep sense of reverence, my gratitude towards my parents and other faculty members of the school for their unconditional support and encouragement.

**Overview**

This project was made with the aim to help organize criminal database and make it more user friendly and accessible for future references.

The crime records are stored in a binary file and a module is created with relevant user-defined functions written for the authorized personnel to access the database and use them for comparing notes and identifying similarities etc.

The functions allow those with access to read, edit, add, remove, and find relevant data from records based on nature of crime, time/location of crime, victim details, criminal details, etc.

There are also functions to allow viewers (those not given authorized access) to search and view crime records.

**Modules**

A module is a file containing functions and variables defined into it. Modules contain functions that perform specific tasks and can be imported into programs when needed by the programmer. It allows for easy access of functions and efficient coding.

Modules imported in this project are – pickle, mysql.connector and time

User-defined module created is crime\_module

* PICKLE

Pickle is a Python module that is used to serialize (convert an object into a character stream or binary string that contains all the information to reconstruct the object) and deserialize (loading the character stream or binary string and creating the object) Python objects. It is imported into the main program and into the user-defined ‘crime\_module’ module to write functions that require access to the binary file in which the data is stored.

* MYSQL.CONNECTOR

MySQL Connector/Python enables Python programs to access MySQL databases, using an API that is compliant with the Python Database API Specification v2.0 (PEP 249). It is written in pure Python and does not have any dependencies except for the Python Standard Library.

* TIME

Python time module allows to work with time in Python. It allows functionality like getting the current time, pausing the Program from executing, etc. The time module in Python provides various time-related functions.

* CRIME\_MODULE

All user-defined functions and variables are created in this module. It contains functions to add, remove, edit and find details across various categories such as nature of crime, suspects, criminals, victims etc. It is imported into the main program.

**Functions**

Functions are a named group of instructions that accomplish specific tasks when invoked. Functions can be user-defined or built-in or defined in modules. Built-in functions are predefined functions that we can use as and when required. User-defined functions are define by the programmer to make tasks easier and make the program more efficient and are often imported from user-defined modules. Functions defined in modules are those that must be imported from user-defined/built-in modules.

Built-in functions used in this program are –

1. print() – It is used to display information to the user.
2. input() – It is used to accept variables from user.
3. eval() – It allows you to evaluate arbitrary Python expressions from a string-based or compiled-code-based input.
4. str() – It converts the specified value into a string.
5. int() – It converts the specified value into an integer number.
6. range() – It returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.
7. len() – It returns the number of items (length) in an object.
8. list.append() – It adds an element to the list at the end.
9. list.clear() – It clears the list and returns an empty list.
10. list.remove() – It takes a value as an argument and removes it from the list.
11. string.lower() – It returns the string in lowercase.
12. string.split() – It splits given text into a list of words
13. file.open() – It returns a file object, which is used to identify a file.
14. file.close() – It flushes the buffer and closes an open file.

Functions imported from modules used in this program are –

1. pickle.dump() – This function is imported from pickle module and is used to serialize (convert) python object into a byte stream.
2. pickle.load() – This function is imported from pickle module and is used to deserialize python objects convert byte stream to user-friendly python object
3. time.sleep(n) – This function gives a pause for ‘n’ seconds.
4. mysql.connector.connect() - It connects the python module with MySQL

User-defined Functions in this program are imported from user defined module ‘crime\_module’. They are –

1. userlogin() – It checks if the username and password if user is authorized
2. victim() – It is defined for the user to add in victim details in a new case. It is invoked in NewCase()
3. crime() – It is defined for the user to add in crime details in a new case. It is invoked in NewCase()
4. evidences() – It is defined for the user to add in evidence details in a new case. It is invoked in NewCase()
5. i\_suspects() – It is defined for the user to add in suspect details in a new case. It is invoked in NewCase()
6. accuse() – It is defined for the user to add in accused details in a new case. It is invoked in NewCase()
7. multi\_an() – It allows to enter multiple analysts and also checks if the entered analysts are valid. It is invoked in NewCase() and uxdets()
8. Valid\_IO() - It allows to enter the investigating officer and checks if the entered value is a valid one. It is invoked in NewCase() and uxdets()
9. file() – It is defined to enter file number and cross check if it is valid. It is invoked in status(), accused(), suspects(), evidence()
10. NewCase() – It is defined to add a new case to the database.
11. status() – It is defined to update status of the case in the database. It is invoked in update() function.
12. asu() – It is a variant of status() that is called in ar() that allows you to update status in the file if accused has been removed.
13. ar() – It is a function defined to remove accused from the database. It is called in accused().
14. uiea() – It is a function defined to update accused details parameter by parameter. It is evoked in accused().
15. accused() – It is defined to add or remove or update accused details in the database. It is invoked in update() function.
16. sur() – It is a function defined to remove a suspect detail and its called in suspects().
17. ueis() – It is a function defined to update suspect details parameter by parameter. It is evoked in suspects().
18. suspects() - It is defined to add or remove or update accused details in the database. It is invoked in update() function.
19. sr() – It is defined to remove samples. It is invoked in samples() function.
20. samples() - It is defined to add or remove comparison samples for a case in the database. It is invoked in evidence() function.
21. wr() - It is defined to remove witness details. It is invoked in witnesses() function.
22. ueiw() –It is a function defined to update witness details parameter by parameter. It is evoked in witnesses().
23. witnesses() - It is defined to add or remove witness details for a case in the database. It is invoked in evidence() function.
24. evidence() - It is defined to add or remove evidences (witness details and material samples) in the database. It is invoked in update() function.
25. uxdets() – It is defined to update the department which handles the case, the investigating officers and analysts. It is invoked in update() function.
26. update() – It is defined to update, add or remove, case details in the database across various parameters like status of crime, accused details, suspect details, evidences etc.
27. updateinfile() – It is defined to update the changes made in the dictionary ‘d’ defined in the module to the binary file ‘GBI\_Records.dat’
28. history() – It is defined to return criminal history of user-inputted criminal.
29. similar() – It is defined to return details of similar crimes based on user-inputted details like charges, evidences collected etc.
30. Fno() – It is defined to accept a file number and check if its there in the database and print the case details. It is called in display()
31. adisplaystatus() – It is defined to display case details based on status - ongoing, solved, unsolved. It is called in adisplay()
32. AllCaseDpt() – It is defined to display details of cases department wise. It displays the case details of the authorized person’s dept. or other dept. It is invoked in adisplaydpt()
33. adisplaydpt() – It further checks if authorized user wants to view all the cases in the department or cases handled by a particular investigating officer. It is invoked in adisplay()
34. adisplay() – It is defined to display details of case depending on parameters like status, department etc.
35. gdisplaystatus() – It is defined to display case details based on status - ongoing, solved, unsolved. It is called in gdisplay() It differs from adisplaystatus in the ‘Go back’ option
36. gdisplaydpt() – It further checks if user wants to view all the cases in the department or cases handled by a particular investigating officer. It is invoked in gdisplay()
37. gdisplay() – It is defined to display details of cases depending on what type of cases the user wants to see.
38. aoptions() – It shows the authorized user the options that the user can perform in the program. It invokes the updateinfile() function for when the user wants to update details for a case or add a case in the database. It invokes the update() function to update details as and when required by the user for option 1. It invokes the history() function to return the criminal history of criminal for option 2. It invokes similar() function to return details about similar cases for option 3. It invokes the NewCase() function when user wants add a case for option 4. It invokes display() function to display case details for option 5. And it breaks the loop and logs out and ends the program for option 6.
39. goptions() – It is like aoptions but is executed when user the user has not logged in with authorized access credentials. It has the same functions as aoptions() except those that allow user to make changes into the records i.e. it does not allow user to update or add a case. It invokes the history() function to return the criminal history of criminal for option 1. It invokes similar() function to return details about similar cases for option 2 It invokes display() function to display case details for option 3. And it breaks the loop and logs out and ends the program for option 4.

**Files**

The database is stored in a binary file. Information in binary files is stored in bytes. It is stored in a computer in a sequence of bytes. These files are not human readable, hence pickle module is used to serialize and deserialize python objects.



d={"#001-B":

{'Victim Details':["Arianna","25 yrs","F","Villa 42, South Street, Jumeirah","056-7894321"],

'Charges':"Burglary",

'Details of Crime':["23/01/23","17:00","Victim's Home","TV and Jewellery Stolen"],

'Evidences':{'Comparison Samples (Forensic Evidence)':["DNA","Fingerprints"],

'Witness Details':[["Nil"]]},

'Suspect Details':[['Nil']],

'Accused Details':[["John C","27 yrs","M","Flat#502, Black Building, Karama","065-9876356"]],

'Status':"Solved",

'Department':'Organized Crime Bureau',

'Investigating Officer':'Nancy Drew',

"Analyst(s)":["Andre Raines","Isobel Castille"]},

"#001-M":

{'Victim Details':["Jack","20 yrs","M","Flat#901, Pearl Oasis Complex, Creek","050-9230487"],

'Charges':"Murder",

'Details of Crime':["29/01/23","23:30","Victim's Home","Knife Wound & Gunshot to Head","Laptop Missing"],

'Evidences':{'Comparison Samples (Forensic Evidence)':["Blue Fiber","Bloody Footprint - Male Size 10"],

'Witness Details':["Nil"]},

'Suspect Details':[["Carlos","29 yrs","M","Flat#608, GreenView Apartments, Karama", "067-9872347"],

["Rihanna","27 yrs","F","Villa 52, Southbridge Community, DSO", "047-9823468"]],

'Accused Details':["Nil"],

'Status':"Ongoing",

'Department':'Homicide Bureau',

'Investigating Officer':'Rosa Diaz',

'Analyst(s)':["Kristen Vega","Ian Daniels"]},

"#001-C":

{'Victim Details':["Caren","17 yrs","F","Flat#105, Al Tala Apartments, Jadaf","092-8762341"],

'Charges':"Cyber Crime",

'Details of Crime':["01/02/23","Via Instagram","Cyber Bullying followed by Account Hack"],

'Evidences':{'Comparison Samples (Forensic Evidence)':["Hacker's Signature"],

'Witness Details':["Nil"]},

'Suspect Details':[["Rory\_23\_Rocking (Screen Name)","Unknown Age","Unknown Gender","Unknown Address", "Contact Info Unavailable"]],

'Accused Details':["Nil"],

'Status':"Unsolved",

'Department':'Cyber Crimes Bureau',

'Investigating Officer':'Erin Reagan',

'Analyst(s)':["Hana Gibson","Remy Scott","Katrin Jeager"]},

"#002-M":

{'Victim Details':["Jasper","25 yrs","M","Flat#301, Avenue Bridge Builidng, Deira","050-8872231"],

'Charges':"Murder",

'Details of Crime':["03/02/23","22:30","Victim's Home","Knife Wound & Gunshot to Head","Laptop Missing"],

'Evidences':{'Comparison Samples (Forensic Evidence)':["Blue Fiber","Bloody Footprint - Male Size 10"],

'Witness Details':["Nil"]},

'Suspect Details':[["Nil"]],

'Accused Details':["Nil"],

'Status':"Ongoing",

'Department':'Homicide Bureau',

'Investigating Officer':'Everly Kingston',

'Analyst(s)': ["Jamie Kellet"]},

"#001-A":

{'Victim Details':["Mairah","21 yrs","F","Villa 55, Stoneybridge Premium Villas, JVC","055-0900213"],

'Charges':"Assault",

'Details of Crime':["30/01/23","21:00","Neighbourhood Park","Blow to the Head","Stolen Purse and Diamond Bracelet"],

'Evidences':{'Comparison Samples (Forensic Evidence)':[""],

'Witness Details':[["Lana J","29 yrs","F","Room 2987, Green Time Hotel Apartments, Dubai Marina", "065-9903218"],

["Amira","21 yrs","F","Villa 57, JBR Beach, Jumeirah","076-9870273"]]},

'Suspect Details':[["Nil"]],

'Accused Details':["Nil"],

'Status':"Ongoing",

'Department':'Organized Crime Bureau',

'Investigating Officer':'Frank Hardy',

'Analyst(s)':["Jubal Valentine"]},

"#003-M":

{'Victim Details':["Evan Buckley","26 yrs","M","#2201, Starry Night Apartments, Broad Street, Elkfield Drive","058-0913883"],

'Charges':"Murder",

'Details of Crime':["23/01/2022","02:00","Elkfield Park","Gunshot to Chest","Missing Ring"],

'Evidences':{'Comparison Samples (Forensic Evidence)':["GSR on Victim's Clothing"],

'Witness Details':["Nil"]},

'Suspect Details':[["Nil"]],

'Accused Details':["Nil"],

'Status':"Ongoing",

'Department':'Homicide Bureau',

'Investigating Officer':'Eve Dallas',

'Analyst(s)':["Jamie Kellet","Ian Daniels"]}}

D={}

import pickle

f=open('GBI\_Records.dat','wb+')

for k in d:

x={}

x[k]=d[k]

pickle.dump(x,f)

f.seek(0)

f=open('GBI\_Records.dat','rb')

try:

while True:

x=pickle.load(f)

for k in x:

D[k]=x[k]

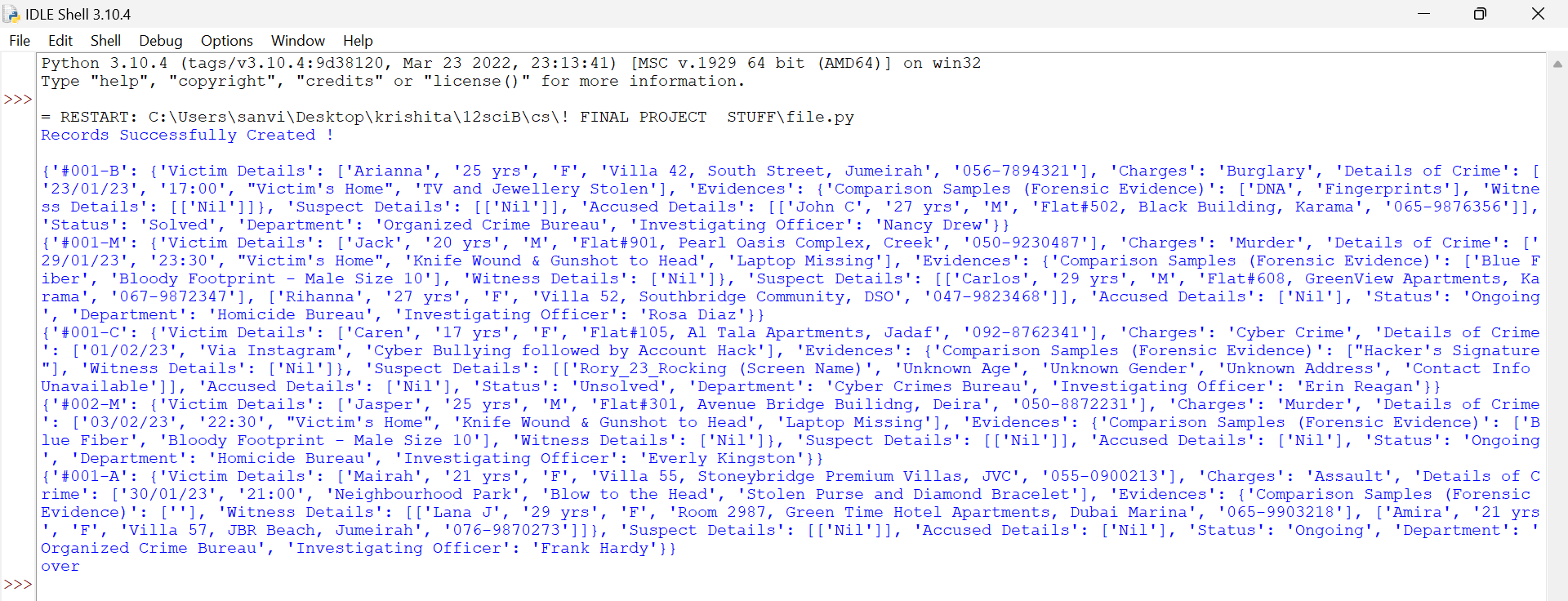
print(x)

except EOFError:

print('over')

f.close()

print(D)



**SQL Table**

SQL stands for Structured Query Language. SQL is a standard language for storing, manipulating and retrieving data in databases. The database created and used in this project is ‘GBI\_Records’ and it has a table titled ‘USER\_LOGIN’ which contains the user credentials of all authorized persons.



Enter password: \*\*\*\*\*\*\*\*

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 72

Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its

affiliates. Other names may be trademarks of their respective

owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> DESC GBI\_RECORDS.USER\_LOGIN;

+----------------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+----------------+--------------+------+-----+---------+----------------+

| S\_NO | int | NO | PRI | NULL | auto\_increment |

| FULL\_NAME | varchar(100) | NO | | NULL | |

| EMAIL | varchar(50) | NO | UNI | NULL | |

| DESIGNATION | varchar(50) | NO | | NULL | |

| DEPARTMENT | varchar(100) | YES | | NULL | |

| USER\_NAME | varchar(50) | NO | UNI | NULL | |

| PASSWORD | varchar(200) | NO | | NULL | |

| LAST\_LOGGED\_IN | datetime | YES | | NULL | |

+----------------+--------------+------+-----+---------+----------------+

8 rows in set (0.02 sec)

mysql> SELECT \* FROM GBI\_RECORDS.USER\_LOGIN;

+------+----------------------+----------------------------+-------------------------------+-------------------------+------------------+-------------------------+---------------------+

|S\_NO| FULL\_NAME | EMAIL | DESIGNATION | DEPARTMENT | USER\_NAME | PASSWORD |LAST\_LOGGED\_IN|

+------+----------------------+----------------------------+-------------------------------+-------------------------+------------------+-------------------------+---------------------+

| 1 | ESHAL AJMAL | ESHAL36542@GMAIL.COM | DATABASE ADMINISTOR | N/A | ESHAL\_A | ESHAL@2006 | NULL |

| 2 | KRISHITA SUDHAKARAN | KRISHITA44867@GMAIL.COM | DATABASE ADMINISTOR | N/A | KRISH\_8 | KRISHITAS | 2023-10-24 07:44:44 |

| 3 | JOANN ELIZABETH BYJU | JOANNBYJU@GMAIL.COM | DATABASE ADMINISTOR | N/A | JOANN\_B | JOANNEB | NULL |

| 4 | ALIZA CARR | ALIZCAR2412@GMAIL.COM | DIRECTOR | N/A | CARR\_ALIZA | CARR\_DIRECTOR@GBI\_ALIZA | NULL |

| 5 | ALICE FRANK | ALICE803\_F@GMAIL.COM | DEPUTY DIRECTOR | N/A | FRANK\_ALICE | FRANK\_308\_ALICE | NULL |

| 6 | NICOLE SEQUIRA | NICOLESEQUIRA@GMAIL.COM | ASSISTANT DIRECTOR | N/A | SEQUIRA\_NICOLE | NS475BQ052 | NULL |

| 7 | IZMA MARWANI | IZMAWANI@GMAIL.COM | BUREAU DIRECTOR | HOMICIDE BUREAU | MARWANI\_IZMA | MAR@2002@WANI\_IZMA | NULL |

| 8 | BARBARA ROBERTS | BARBARAMROBERTS@GMAIL.COM | BUREAU DIRECTOR | FINANCIAL CRIMES BUREAU | ROBERTS\_BARBARA | KENSUCKS | NULL |

| 9 | GLORIA FERNANDEZ | GLORIAF9@GMAIL.COM | BUREAU DIRECTOR | ORGANIZED CRIME BUREAU | FERNANDEZ\_GLORIA | GL0RI@\_FE&@&DEZ | NULL |

| 10 | DAYANARA RUIZ | DAYARUIZNARA@GMAIL.COM | BUREAU DIRECTOR | CYBER CRIMES BUREAU | RUIZ\_DAYANARA | RUETODAYA | 2023-11-14 12:49:49 |

| 11 | EVE DALLAS | DALLAS.EVE@GMAIL.COM | SPECIAL AGENT IN CHARGE | HOMICIDE BUREAU | DALLAS\_EVE | EVE&ROARKE@NYSPD | NULL |

| 12 | JOHN WATSON | WATSONJ@GMAIL.COM | SPECIAL AGENT IN CHARGE | FINANCIAL CRIMES BUREAU | WATSON\_JOHN | JOHNTHESIDEKICK | NULL |

| 13 | SHERLOCK HOLMES | HOMESSHERLOCK@GMAIL.COM | SPECIAL AGENT IN CHARGE | ORGANIZED CRIME BUREAU | HOLMES\_SHERLOCK | ARTHURCONANDOYLE | NULL |

| 14 | IAN MERRICK | MERRICKIA@GMAIL.COM | SPECIAL AGENT IN CHARGE | CYBER CRIMES BUREAU | MERRICK\_IAN | EIANEMERRICK | NULL |

| 15 | ALEXANDER PERRO | ALEXPERRO@GMAIL.COM | ASST. SPECIAL AGENT IN CHARGE | HOMICIDE BUREAU | PERRO\_ALEXANDER | 123PERRO456ALEX | NULL |

| 16 | DELIA ROARKE | RODELIA@GMAIL.COM | ASST. SPECIAL AGENT IN CHARGE | FINANCIAL CRIMES BUREAU | ROARKE\_DELIA | DELIA\_IAN\_L | NULL |

| 17 | JANE RIZZOLI | RIZZOLIJ@GMAIL.COM | ASST. SPECIAL AGENT IN CHARGE | ORGANIZED CRIME BUREAU | RIZZOLI\_JANE | JANE@RIZZOLI@ISLES | NULL |

| 18 | MAURA ISLES | ISLESMAUR@GMAIL.COM | ASST. SPECIAL AGENT IN CHARGE | CYBER CRIMES BUREAU | ISLES\_MAURA | MAURAPEDIA | NULL |

| 19 | DIANA PRINCE | PRINCED@GMAIL.COM | SPECIAL AGENT | HOMICIDE BUREAU | PRINCE\_DIANA | THEMYSCIRA4EVER | NULL |

| 20 | KEVIN JACKSON | JACKEDKEVIN@GMAIL.COM | SPECIAL AGENT | HOMICIDE BUREAU | JACKSON\_KEVIN | STUFFOFIMAGINATION | NULL |

| 21 | EVERLY KINGSTON | KINGSEVERLY@GMAIL.COM | SPECIAL AGENT | HOMICIDE BUREAU | KINGSTON\_EVERLY | 13@ALPHA\_EVERLY | NULL |

| 22 | ROSA DIAZ | DIAZRORO@GMAIL.COM | SPECIAL AGENT | HOMICIDE BUREAU | DIAZ\_ROSA | AUNTRORODIAZ | 2023-11-14 11:47:33 |

| 23 | EVANGELINE BLACK | BLACKEVAN@GMAIL.COM | SPECIAL AGENT | HOMICIDE BUREAU | BLACK\_EVANGELINE | BLACKKNOWSNOBOUNDS | NULL |

| 24 | OLIVIA BENSON | BEN10LIVIE@GMAIL.COM | SPECIAL AGENT | FINANCIAL CRIMES BUREAU | BENSON\_OLIVIA | LIV@10BENSON | NULL |

| 25 | AMRITHA ANIL | AMRITHAANIL40305@GMAIL.COM | SPECIAL AGENT | FINANCIAL CRIMES BUREAU | ANIL\_AMRITHA | AMMU@134340 | 2023-11-14 16:29:42 |

| 26 | ELLIOT STABBLER | STABLERELLIOT@GMAIL.COM | SPECIAL AGENT | FINANCIAL CRIMES BUREAU | STABLER\_ELLIOT | STABBYELLI0T | NULL |

| 27 | VINCE KORSAK | VINNIE2345@GMAIL.COM | SPECIAL AGENT | FINANCIAL CRIMES BUREAU | KORSAK\_VINCE | WINVINNIE | NULL |

| 28 | BARRY FROST | FROSTYBARRY@GMAIL.COM | SPECIAL AGENT | FINANCIAL CRIMES BUREAU | FROST\_BARRY | FROSTYBARS224 | NULL |

| 29 | NANCY DREW | DREWNANCY@GMAIL.COM | SPECIAL AGENT | ORGANIZED CRIME BUREAU | DREW\_NANCY | TEEN@N@NCY | NULL |

| 30 | VERONICA MARS | VMARS@GMAIL.COM | SPECIAL AGENT | ORGANIZED CRIME BUREAU | MARS\_VERONICA | PI@NEPTUNE | NULL |

| 31 | FRANK HARDY | F.HARDY@GMAIL.COM | SPECIAL AGENT | ORGANIZED CRIME BUREAU | HARDY\_FRANK | FR@NK2433 | NULL |

| 32 | MELISSA REO | MELLIE.R@GMAIL.COM | SPECIAL AGENT | ORGANIZED CRIME BUREAU | REO\_MELISSA | LIZZ@RE0 | NULL |

| 33 | NADINE FERNANDEZ | NADINE.AARDRA.F@GMAIL.COM | SPECIAL AGENT | ORGANIZED CRIME BUREAU | FERNANDEZ\_NADINE | NADINE2209AF | NULL |

| 34 | JOE HARDY | J.HARDY@GMAIL.COM | SPECIAL AGENT | CYBER CRIMES BUREAU | HARDY\_JOE | JOEYHARDY | 2023-11-12 20:59:44 |

| 35 | MARIANNA SIDLEY | SIDLEY\_MARIANNA@GMAIL.COM | SPECIAL AGENT | CYBER CRIMES BUREAU | SIDLEY\_MARIANNA | M@R1N@S1D | NULL |

| 36 | VINCENT D'CRUZ | D'CRUZ.VINNIE@GMAIL.COM | SPECIAL AGENT | CYBER CRIMES BUREAU | D'CRUZ\_VINCENT | VINNIECRUZ | NULL |

| 37 | MAGGIE BELL | MAGSBELL@GMAIL.COM | SPECIAL AGENT | CYBER CRIMES BUREAU | BELL\_MAGGIE | MAGGS@BELLS | NULL |

| 38 | ERIN REAGAN | E.REAGAN@GMAIL.COM | SPECIAL AGENT | CYBER CRIMES BUREAU | REAGAN\_ERIN | HFJDEJ@REAGAN | 2023-11-14 12:27:17 |

| 39 | KRISTEN VEGA | KKRIS@GMAIL.COM | SENIOR ANALYST | HOMICIDE BUREAU | VEGA\_KRISTEN | K\_VEGA\_KRISTEN | NULL |

| 40 | SCOTT FORRESTER | FORESTSCOTT@GMAIL.COM | SENIOR ANALYST | FINANCIAL CRIMES BUREAU | FORRESTER\_SCOTT | SCOOTYFOREST | NULL |

| 41 | ANDRE RAINES | RAINANDRE@GMAIL.COM | SENIOR ANALYST | ORGANIZED CRIME BUREAU | RAINES\_ANDRE | RAINY@NDRE | NULL |

| 42 | HANA GIBSON | GIBBSHANA@GMAIL.COM | SENIOR ANALYST | CYBER CRIMES BUREAU | GIBSON\_HANA | H@N@&G!BS0N | NULL |

| 43 | IAN DANIELS | DANNY.IAN@GMAIL.COM | ANALYST | HOMICIDE BUREAU | DANIELS\_IAN | DAN\_IAN | NULL |

| 44 | JAMIE KELLET | KELLET.JAMIE@GMAIL.COM | ANALYST | HOMICIDE BUREAU | KELLET\_JAMIE | JAMIEHSCOTT | NULL |

| 45 | CAMERON VO | VOCAM@GMAIL.COM | ANALYST | FINANCIAL CRIMES BUREAU | VO\_CAMERON | CAMMY!VO# | NULL |

| 46 | MEGAN GARRETSON | GARRETMEG@GMAIL.COM | ANALYST | FINANCIAL CRIMES BUREAU | GARRETSON\_MEGAN | MEGANGARRET | NULL |

| 47 | JUBAL VALENTINE | VALENTINE.J@GMAIL.COM | ANALYST | ORGANIZED CRIME BUREAU | VALENTINE\_JUBAL | VALENTIN!JUB@L | NULL |

| 48 | ISOBEL CASTILLE | BELLACASTILLE@GMAIL.COM | ANALYST | ORGANIZED CRIME BUREAU | CASTILLE\_ISOBEL | ISSYC@ST!LLE | NULL |

| 49 | REMY SCOTT | SCOTTYR@GMAIL.COM | ANALYST | CYBER CRIMES BUREAU | SCOTT\_REMY | REMITSC0TT | NULL |

| 50 | KATRIN JAEGER | J.KATRIN@GMAIL.COM | ANALYST | CYBER CRIMES BUREAU | JAEGER\_KATRIN | K@TTYJ@EGER | NULL |

+------+----------------------+----------------------------+-------------------------------+-------------------------+------------------+-------------------------+---------------------+

50 rows in set (0.00 sec)

**SQL Connectivity with Python**

SQL databases can be accessed via python to perform various functions. Python needs a MySQL driver to access the MySQL database. In this project, we need to access MySQL databases to verify the user access credentials. We further use the information retrieved from the SQL database to decide what functions apply to the user and restrict the user from perform functions for which they do not have the authority to perform.



import pickle

obj=mysql.connector.connect(\

host='localhost',

database='GBI\_RECORDS',

user='root',

password='SQL@1984')

c=obj.cursor()

q='SELECT FULL\_NAME, DESIGNATION, DEPARTMENT FROM USER\_LOGIN'

c.execute(q)

l=c.fetchall()

def userlogin():

global ut

global user

Q='SELECT \* FROM USER\_LOGIN'

c.execute(Q)

L=c.fetchall()

print ("Welcome to the GBI Crime Records Dept.")

print ("\nDo you wish to continue with authorized login?")

print ("1. Yes")

print ("2. No")

print()

x=str(input("Enter your choice: "))

if x=="2":

ut='View Only Access'

elif x=="1":

for i in range(3):

flag='not logged in'

flag2="valid username"

print('\nYou have',3-i,'attempt(s) to complete your login \n')

u=str(input("Username: "))

for i in L:

if u in i:

break

else:

flag2="invalid username"

print ("Invalid Username!")

if flag2=="valid username":

p=str(input("Password: "))

for i in L:

if u==i[5] and p==i[6]:

ut='Authorised Access'

user=i

flag='logged in'

break

else:

print('Incorrect Password!')

if flag=='logged in':

q='UPDATE USER\_LOGIN SET LAST\_LOGGED\_IN=CURRENT\_TIMESTAMP WHERE USER\_NAME=%s;'

c.execute(q,(u,))

obj.commit()

break

else:

print('\nYou have exceeded the number of login attempts \nRedirecting to View Only Access...\n')

ut='View Only Access'

else:

print("\nInvalid Option")

print("Auto-Redirecting to View Only Access...\n")

time.sleep(2)

ut='View Only Access'

**Program**



import mysql.connector

import Crime\_Module

import time

Crime\_Module.userlogin()

print(Crime\_Module.ut)

if Crime\_Module.ut=='Authorised Access':

x=Crime\_Module.user[1].title()+', '+Crime\_Module.user[3].title()

print('\nHi',x)

if Crime\_Module.user[7]!=None:

s=str(Crime\_Module.user[7])

l1=s.split()

print('Your last login was on ',l1[0],"at",l1[1],'\n')

else:

print ("This is your First Login")

time.sleep(1)

Crime\_Module.aoptions()

else:

time.sleep(1)

Crime\_Module.goptions()



import pickle

import mysql.connector

import time

obj=mysql.connector.connect(\

host='localhost',

database='GBI\_RECORDS',

user='root',

password='SQL@1984')

c=obj.cursor()

q='SELECT FULL\_NAME, DESIGNATION, DEPARTMENT FROM USER\_LOGIN'

c.execute(q)

l=c.fetchall()

d={}

flag="red"

f=open('GBI\_Records.dat','rb')

try:

while True:

x=pickle.load(f)

for k in x:

d[k]=x[k]

except EOFError:

f.close()

def userlogin():

global ut

global user

Q='SELECT \* FROM USER\_LOGIN'

c.execute(Q)

L=c.fetchall()

print ("Welcome to the GBI Crime Records Dept.")

print ("\nDo you wish to continue with authorized login?")

print ("1. Yes")

print ("2. No")

print()

x=str(input("Enter your choice: "))

if x=="2":

ut='View Only Access'

elif x=="1":

for i in range(3):

flag='not logged in'

flag2="valid username"

print('\nYou have',3-i,'attempt(s) to complete your login \n')

u=str(input("Username: "))

for i in L:

if u in i:

break

else:

flag2="invalid username"

print ("Invalid Username!")

if flag2=="valid username":

p=str(input("Password: "))

for i in L:

if u==i[5] and p==i[6]:

ut='Authorised Access'

user=i

flag='logged in'

break

else:

print('Incorrect Password!')

if flag=='logged in':

q='UPDATE USER\_LOGIN SET LAST\_LOGGED\_IN=CURRENT\_TIMESTAMP WHERE USER\_NAME=%s;'

c.execute(q,(u,))

obj.commit()

break

else:

print('\nYou have exceeded the number of login attempts \nRedirecting to View Only Access...\n')

ut='View Only Access'

else:

print("\nInvalid Option")

print("Auto-Redirecting to View Only Access...\n")

time.sleep(2)

ut='View Only Access'

def victim():

global lv

lv=[]

print()

print('Victim Details')

lv.append(str(input('Name: ')))

lv.append(str(input('Age: ')))

lv.append(str(input('Gender: ')))

lv.append(str(input('Address: ')))

lv.append(str(input('Contact: ')))

def crime():

global ch

global dets

print()

print('Crime Details')

ch=str(input('Charges Pressed against the Accused: '))

dets=str(input('Enter the Details of the Crime (Date, Time, Location, Important Facts) briefly: ')).split(',')

def evidences():

global de

de={}

lwn=[]

print()

print('Evidences')

el=input('Enter the List of Comparison Samples: ')

ec=el.split(',')

if len(ec)==0:

ec=["Nil"]

nw=int(input('Enter the no. of Witnesses: '))

print()

for i in range(nw):

lw=[]

lw.append(str(input('Name: ')))

lw.append(str(input('Age: ')))

lw.append(str(input('Gender: ')))

lw.append(str(input('Address: ')))

lw.append(str(input('Contact: ')))

lwn.append(lw)

if nw==0:

lwn+=['Nil']

de['Comparison Samples (Forensic Evidence)']=ec

de['Witness Details']=lwn

def i\_suspects():

global lsn

lsn=[]

print()

print ("Suspect Details")

ns=int(input('Enter the no. of Suspects: '))

for i in range(ns):

print()

ls=[]

ls.append(str(input('Name: ')))

ls.append(str(input('Age: ')))

ls.append(str(input('Gender: ')))

ls.append(str(input('Address: ')))

ls.append(str(input('Contact: ')))

lsn.append(ls)

if ns==0:

lsn+=['Nil']

def accuse():

global la

la=[]

print()

print('Accused Details')

j=input("Has the Accused been Identified? ")

if "y" in j.lower():

print()

x=str(input("Are there Multiple Accused? "))

if "y" in x.lower():

n=eval(input("Enter the no. Accused: "))

for i in range(n):

print()

a=[]

print ("Enter the Details of Accused no.",i,":")

a.append(str(input('Name: ')))

a.append(str(input('Age: ')))

a.append(str(input('Gender: ')))

a.append(str(input('Address: ')))

a.append(str(input('Contact: ')))

la.apend(a)

elif "n" in x.lower():

print()

print ("Enter the Details")

la.append(str(input('Name: ')))

la.append(str(input('Age: ')))

la.append(str(input('Gender: ')))

la.append(str(input('Address: ')))

la.append(str(input('Contact: ')))

else:

print ("Enter Yes or No")

accuse()

elif "n" in j.lower():

la=["Nil"]

else:

print ("Enter Yes or No")

accuse()

def multi\_an():

global an

global dpt

an=[]

dal=[]

for i in l:

if i[2]==dpt.upper() and "ANALYST" in i[1]:

dal.append(i[0].title())

cp=input("Has an Analyst been Assigned? ")

if "y" in cp.lower():

print()

x=str(input("Are there Multiple Analysts? "))

if "y" in x.lower():

n=eval(input("Enter the no. analysts: "))

print()

i=1

while i<=n:

aa= input ("Enter the Name of the Analyst: ")

if aa in dal:

an.append(aa)

i+=1

else:

print('Invalid Analyst')

print('Select an analyst from the below analyst:')

print(dal)

elif "n" in x.lower():

aaa=input ("Enter the Name of Ananlyst : ")

an.append(aaa)

else:

print ("Enter Yes or No")

multi\_an()

elif "n" in cp.lower():

an=["Nil"]

else:

print ("Enter Yes or No")

multi\_an()

def Valid\_IO():

global dpt

global ins

for i in l:

dl=[]

if i[2]==dpt and i[1] in ['SPECIAL AGENT, SPECIAL AGENT IN CHARGE, ASST. SPECIAL AGENT IN CHARGE']:

dl+=[i[0]]

ins=str(input('Please Enter the Investigating Officer: '))

if ins not in dl:

print('Invalid Investigating Officer')

print('Select an investigating officer from the below officers:')

for k in dl:

print(k)

Valid\_IO()

def File():

global file

global fileee

global flag

print()

file=input("Enter the File No.: ")

if file not in d:

print()

print ("Invalid File No. ")

File()

fileee='valid'

if user[3] in ['SPECIAL AGENT IN CHARGE','ASST. SPECIAL AGENT IN CHARGE','SENIOR ANALYST']:

if d[file]['Department']!=user[4].title():

print("Access to file denied")

print("ERROR 404 - Cross Department Access")

fileee='invalid'

flag="green"

aoptions()

elif user[3] in ['SPECIAL AGENT','ANALYST']:

if user[1].title() not in d[file]['Investigating Officer'] or user[1].title not in d[file]['Analyst(s)']:

print("Unauthorized Access to File")

print('Access Denied')

fileee='invalid'

flag="green"

aoptions()

def NewCase():

global d

global dpt

global ins

global flag

global an

flag="red"

print()

print('Options: ')

print('i. Continue')

print ('ii. Go Back')

print()

cp=input('Enter your Choice: ')

if cp=="i":

file\_id=input("Enter File ID: ")

if file\_id in d:

print ("File Exists")

print()

print ("Do you wish to Update the Details in the File?")

print('Options: ')

print('i. Update')

print ('ii. Go Back')

print()

cn=input('Enter your Choice: ')

if cn=="i":

update()

elif cn=="ii":

flag="green"

aoptions()

else:

NewCase()

else:

dpt=str(input("Please Enter the Department Investigating the Case: "))

if dpt.upper()==user[4] or user[4]=='N/A':

victim()

crime()

evidences()

i\_suspects()

accuse()

print()

stat=str(input('Status (Ongoing/Solved/Unsolved): '))

if user[4]=="N/A" or user[3] in ["BUREAU DIRECTOR", "SPECIAL AGENT IN CHARGE", "ASST. SPECIAL AGENT IN CHARGE"]:

Valid\_IO()

multi\_an()

elif user[3]=="SPECIAL AGENT":

ins=user[1].title()

multi\_an()

elif user[3]=='SENIOR ANALYST':

Valid\_IO()

multi\_an()

elif i[1]=='ANALYST':

Valid\_IO()

x=str(input("Are there Multiple Analysts Working with You? "))

if "y" in x.lower():

n=eval(input("Enter the no. analysts apart from yourself: "))

print()

i=1

while i<=n:

aa= input ("Enter the Name of the Analyst: ")

if aa in dal:

an.apend(aa)

i+=1

else:

print('Invalid Analyst')

print('Select from the below analysts:')

print(dal)

an.append(user[1])

elif 'n' in x.lower():

an.append(i[0])

else:

print ("Enter a valid option")

print ("Re-routing to Start")

NewCase()

d1={'Victim Details':lv,'Charges':ch,'Details of Crime':dets,'Evidences':de,'Suspect Details':lsn,'Accused Details':la,'Status':stat,"Department":dpt,"Investigating Officer":ins,"Analyst(s)":an}

d[file\_id]=d1

print()

print("Case Added")

else:

print("Access to file denied")

print("ERROR 404 - Cross Department Access")

print ("Do you wish to add another case?")

NewCase()

elif cp=="ii":

flag="green"

aoptions()

else:

print ("Invalid Option")

NewCase()

#Status

def status():

File()

if fileee=='valid':

a=input("Enter the New Status: ")

d[file]["Status"]=a

print ("Status Updated")

def asu():

q=input("Enter your Choice: ")

if "y" in q.lower():

a=input("Enter the New Status: ")

d[file]["Status"]=a

print ("Status Updated")

elif "n" in q.lower():

pass

else:

print ("Invalid Choice")

print()

asu()

def ar():

n=input("Enter the Name: ")

for i in d[file]["Accused Details"]:

if n in i:

d[file]["Accused Details"].remove(i)

print()

print ("Accused Removed")

if len(d[file]["Accused Details"])==0:

d[file]["Accused Details"]+=["Nil"]

if d[file]["Status"]=="Solved":

print ()

print ("You have 0 accused, would you like to change the status? ")

asu()

break

else:

print ("Invalid Name")

ar()

def ueia():

x=input("Enter the Name of Accused: ").lower()

print()

print('Parameters:')

print('i. Name')

print('ii. Age')

print('iii. Gender')

print('iv. Address')

print('v. Contact')

print()

for i in d[file]["Accused Details"]:

if i[0].lower()==x:

c=str(input('Enter your Choice: '))

if c=="i":

u=input("Enter the Updated Name: ")

i[0]=u

print ("Details Updated")

break

elif c=="ii":

u=input("Enter the Updated Age: ")

i[1]=u

print ("Details Updated")

break

elif c=="iii":

u=input("Enter the Updated Gender: ")

i[2]=u

print ("Details Updated")

break

elif c=="iv":

u=input("Enter the Updated Address: ")

i[3]=u

print ("Details Updated")

break

elif c=="v":

u=input("Enter the Updated Contact: ")

i[4]=u

print ("Details Updated")

break

else:

print ("Invalid Choice")

ueia()

else:

print ("Invalid Name")

ueia()

#Accused

def accused():

print()

print('Parameters: ')

print('i. Add')

print('ii. Remove')

print ('iii. Update Existing Info')

print('iv. Go Back')

print()

c=str(input('Enter your Choice: '))

File()

if fileee=='valid':

l=d[file]["Accused Details"]

else:

c='iv'

if c=="i":

if l==["Nil"]:

l.clear()

print()

print ("New Accused Details")

las=[]

las.append(str(input('Name: ')))

las.append(str(input('Age: ')))

las.append(str(input('Gender: ')))

las.append(str(input('Address: ')))

las.append(str(input('Contact: ')))

l.append(las)

d[file]["Accused Details"]=l

print()

print ("New Accused Added")

elif c=="ii":

ar()

elif c=='iii':

ueia()

elif c=="iv":

update()

else:

print ("Invalid Choice")

accused()

def sur():

n=input("Enter the Name: ")

for i in d[file]["Suspect Details"]:

if n in i:

d[file]["Suspect Details"].remove(i)

print()

print ("Suspect Removed")

if len(d[file]["Suspect Details"])==0:

d[file]["Suspect Details"]+=["Nil"]

break

else:

print ("Invalid Name")

sur()

def ueis():

x=input("Enter the name of suspect: ").lower()

print()

print('Parameters:')

print('i. Name')

print('ii. Age')

print('iii. Gender')

print('iv. Address')

print('v. Contact')

print()

c=str(input('Enter your Choice: '))

for i in d[file]["Suspect Details"]:

if i[0].lower()==x:

if c=="i":

u=input("Enter the Updated Name: ")

i[0]=u

break

elif c=="ii":

u=input("Enter the Updated Age: ")

i[1]=u

break

elif c=="iii":

u=input("Enter the Updated Gender: ")

i[2]=u

break

elif c=="iv":

u=input("Enter the Updated Address: ")

i[3]=u

break

elif c=="v":

u=input("Enter the Updated Contact: ")

i[4]=u

break

else:

print ("Invalid Choice")

ueis()

else:

print ("Invalid Name")

ueis()

#Suspects

def suspects():

print()

print('Parameters: ')

print('i. Add')

print('ii. Remove')

print ('iii. Update Existing Info')

print ('iv. Go Back')

print()

c=str(input('Enter your Choice: '))

File()

if fileee=='valid':

l=d[file]["Suspect Details"]

else:

c='iv'

if c=="i":

if l==[["Nil"]]:

l.clear()

print()

print ("New Suspect Details")

ls=[]

ls.append(str(input('Name: ')))

ls.append(str(input('Age: ')))

ls.append(str(input('Gender: ')))

ls.append(str(input('Address: ')))

ls.append(str(input('Contact: ')))

l.append(ls)

d[file]["Suspect Details"]=l

print()

print ("New Suspect Added")

elif c=="ii":

sur()

elif c=="iii":

ueis()

elif c=="iv":

update()

else:

print ("Invalid Choice")

suspects()

def sr():

k=str(input("What sample would you like to remove? "))

for i in d[file]["Evidences"]["Comparison Samples (Forensic Evidence)"]:

if k in i:

d[file]["Evidences"]["Comparison Samples (Forensic Evidence)"].remove(i)

print()

print ("Sample Removed")

if len(d[file]["Evidences"]["Comparison Samples (Forensic Evidence)"])==0:

d[file]["Evidences"]['Comparison Samples (Forensic Evidence)']+=["Nil"]

break

else:

print("Invalid Sample")

sr()

def samples():

print()

print('Parameters: ')

print('i. Add Sample')

print('ii. Remove Sample')

print('iii. Go Back')

print()

cp=str(input('Enter your Choice: '))

l=d[file]["Evidences"]['Comparison Samples (Forensic Evidence)']

if cp=="i":

if l==["Nil"]:

l.clear()

e=input("Enter the Sample: ")

l.append(e)

d[file]["Evidences"]["Comparison Samples (Forensic Evidence)"]=l

print ("New Sample Added")

elif cp=="ii":

sr()

elif c=="iii":

evidence()

else:

print ("Invalid Choice")

samples()

def wr():

n=input("Enter the Name: ")

for i in d[file]["Evidences"]["Witness Details"]:

if n in i:

d[file]["Evidences"]["Witness Details"].remove(i)

print()

print ("Witness Removed")

if len(d[file]["Evidences"]["Witness Details"])==0:

d[file]["Evidences"]["Witness Details"]+=["Nil"]

break

else:

print ("Invalid Name")

wr()

def ueiw():

x=input("Enter the Name of Witness: ").lower()

print()

print('Parameters:')

print('i. Name')

print('ii. Age')

print('iii. Gender')

print('iv. Address')

print('v. Contact')

print()

c=str(input('Enter your Choice: '))

for i in d[file]["Evidences"]["Witness Details"]:

if i[0].lower()==x:

if c=="i":

u=input("Enter the Updated Name: ")

i[0]=u

break

elif c=="ii":

u=input("Enter the Updated Age: ")

i[1]=u

break

elif c=="iii":

u=input("Enter the Updated Gender: ")

i[2]=u

break

elif c=="iv":

u=input("Enter the Updated Address: ")

i[3]=u

break

elif c=="v":

u=input("Enter the Updated Contact: ")

i[4]=u

break

else:

print ("Invalid Choice")

ueiw()

else:

print ("Invalid Name")

ueiw()

def witnesses():

print()

print('Parameters:')

print('i. Add Witness')

print('ii. Remove Witness')

print ("iii. Update Existing Info")

print('iv. Go Back')

print()

cp=str(input('Enter your Choice: '))

l=d[file]["Evidences"]["Witness Details"]

if cp=="i":

if l==[["Nil"]]:

l.clear()

print()

print ("New Witness Details")

ls=[]

ls.append(str(input('Name: ')))

ls.append(str(input('Age: ')))

ls.append(str(input('Gender: ')))

ls.append(str(input('Address: ')))

ls.append(str(input('Contact: ')))

l.append(ls)

print()

print ("New Witness Added")

elif cp=="ii":

wr()

elif cp=="iii":

ueiw()

elif cp=="iv":

evidence()

else:

print ("Invalid Choice")

witnesses()

#Evidence

def evidence():

print()

print('Parameters: ')

print('i. Comparison Samples')

print('ii. Witness')

print('iii. Go Back')

print()

c=str(input('Enter your Choice: '))

File()

if fileee=='valid':

if c=="i":

samples()

elif c=="ii":

witnesses()

elif c=="iii":

update()

else:

print ("Invalid Choice")

evidence()

def uxdets():

global flg

global dpt

File()

if fileee=='valid':

print()

print('Transfer Ownership: ')

print('i. Department')

print('ii. Investigating Officer')

print('iii. Analyst(s)')

print('iv. Go Back')

print()

if flg==0:

c=str(input('Enter your Choice: '))

if c=='i':

dpt=str(input("Please Enter the Department: "))

Valid\_IO()

d[file]['Investigating Officer']=ins

multi\_an()

d[file]['Analyst(s)']=an

print ("Case Transferred to Another Department")

elif c=='ii' or flg==2:

flg=2

dl=[]

for i in l:

if i[2]==d[file]["Department"].upper() and "AGENT" in i[1]:

print (i)

dl.append(i[0].title())

nio=str(input('Enter the New Investigating Officer: '))

if nio in dl:

d[file]['Investigating Officer']=nio

else:

print('Invalid Officer')

print('Select Investigating Officer from the Available List of Officers - ',dl)

uxdets()

print ("Investigating Officer Changed")

elif c=='iii' or flg==3:

flg=3

dal=[]

for i in l:

if i[2]==d[file]['Department'].upper() and "ANALYST" in i[1]:

dal.append(i[0].title())

print('All Current Analysts removed \n Enter new analyst(s) details: \n')

cp=input("Has a new analyst been Assigned? ")

if "y" in cp.lower():

an=[]

print()

x=str(input("Are there Multiple Analysts? "))

if "y" in x.lower():

n=eval(input("Enter the no. analysts: "))

print()

i=1

while i<=n:

aa= input ("Enter the Name of Analyst: ")

if aa in dal:

an.append(aa)

i+=1

else:

print('Invalid Analyst')

d[file]['Analyst(s)']=an

elif "n" in x.lower():

aaa=input ("Enter the Name of Analyst : ")

an.append(aaa)

d[file]['Analyst(s)']=an

else:

print ("Enter Yes or No")

uxdets()

elif "n" in j.lower():

an=["Nil"]

print ("Analyst Details Changed")

else:

print ("Enter Yes or No")

uxdets()

elif c=="iv":

update()

else:

print("Invalid Choice")

uxdets()

def update():

global flag

flag="red"

print()

print('Parameters :')

print('i. Status')

print('ii. Accused')

print('iii. Suspects')

print('iv. Evidence')

print('v. Transfer Case')

print('vi. Go Back')

print()

c=str(input('Enter your Choice: '))

if c=="i":

status()

elif c=="ii":

accused()

elif c=="iii":

suspects()

elif c=="iv":

evidence()

elif c=='v':

global flg

flg=0

uxdets()

elif c=='vi':

flag="green"

aoptions()

else:

print ("Invalid Choice")

update()

#History

def history():

global ut

global flag

flag="red"

print()

print('Options: ')

print('i. Continue')

print ('ii. Go Back')

print()

cp=input('Enter your Choice: ')

if cp=="i":

criminal=str(input('Enter Name of Criminal: ')).title()

c=0

for k in d:

for w in d[k]['Accused Details']:

if criminal in w[0].title():

print()

print("File No.:",k)

for j in d[k]:

print(j,":",d[k][j],end="\n")

c+=1

else:

if c==0:

print('All Clear')

elif cp=="ii":

if ut=="Authorised Access":

aoptions()

flag="green"

else:

goptions()

flag="green"

else:

print ("Invalid Option")

history()

#Similar

def similar():

global flag

flag="red"

simc=0

print()

print('Parameters:')

print('i. Charges')

print('ii. Details')

print('iii. Evidence')

print ('iv. Go Back')

print()

cp=input('Enter your choice - ')

if cp=="i":

c=str(input('Enter charges - '))

p=c.lower()

for k in d:

if d[k]['Charges'].lower()==p:

print()

simc+=1

print("File No.:",k)

for j in d[k]:

print(j,":",d[k][j],end="\n")

else:

if simc==0:

print('No existing cases matches')

elif cp=="ii":

dt=str(input('Enter the Details of Crime (Date, Time, Location, Important Facts) briefly: ')).lower()

x=dt.split(",")

for k in d:

for y in d[k]['Details of Crime']:

if y.lower() in x:

simc+=1

print()

print("File No.:",k)

for j in d[k]:

print(j,":",d[k][j],end="\n")

break

else:

if simc==0:

print('No existing cases matches')

elif cp=="iii":

print()

print('Parameters - ')

print('i. Comparison Samples')

print('ii. Witness')

print('iii. Go Back')

print()

c=str(input('Enter your Choice: '))

if c=="i":

el\_=str(input('Enter Comparison Sample: '))

for k in d:

for y in d[k]['Evidences']['Comparison Samples (Forensic Evidence)']:

if el\_==y:

print()

simc+=1

print("File No.:",k)

for j in d[k]:

print(j,":",d[k][j],end="\n")

break

else:

if simc==0:

print('No existing cases matches')

elif c=="ii":

wl\_=str(input('Enter Witness Name: '))

for k in d:

for y in d[k]['Evidences']['Witness Details']:

for l in y:

if wl\_==y[0]:

print()

print("File No.:",k)

for j in d[k]:

print(j,":",d[k][j],end="\n")

simc+=1

break

else:

if simc==0:

print('No existing cases matches')

elif c=="iii":

similar()

else:

print ("Invalid Choice")

similar()

elif cp=='iv':

if ut=="Authorised Access":

aoptions()

flag="green"

else:

goptions()

flag="green"

else:

print ("Invalid Choice")

print()

similar()

def Fno():

f=str(input("Enter the file no.: "))

for i in d:

if i==f:

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

break

else:

print("Invalid File No.")

Fno()

#Display()

def adispalystatus():

global flag

flag="red"

print()

print('Parameters: ')

print('i. Ongoing')

print('ii. Solved')

print('iii. Unsolved')

print ('iv. Go Back')

print()

cp=str(input('Enter your Choice: '))

if cp=='i':

c=0

for i in d:

if d[i]["Status"].lower()=="ongoing":

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print ("No Ongoing Cases")

elif cp=='ii':

c=0

for i in d:

if d[i]["Status"].lower()=="solved":

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print ("No Solved Cases")

elif cp=="iii":

c=0

for i in d:

if d[i]["Status"].lower()=="unsolved":

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print ("No Unsolved Cases")

elif cp=="iv":

flag="green"

adisplay()

else:

print ("Invalid Option")

adispalystatus()

def AllCaseDpt():

global flag

global user

flag="red"

print()

print('Parameters: ')

print('i. Your Department')

print('ii. Other Department')

print ('iii. Go Back')

print()

ch=str(input('Enter your Choice: '))

if ch=="i":

c=0

for i in d:

if d[i]["Department"].upper()==user[4]:

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print ("No Cases Exist in Your Deaprtment")

elif ch=="ii":

print()

dpt=str(input("Enter the Department: "))

c=0

for i in d:

if d[i]["Department"].upper()==dpt.upper():

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print ("No Cases Exist in the",dpt)

elif ch=="iii":

flag='green'

adisplaydpt()

else:

AllCaseDpt()

def adisplaydpt():

global flag

global user

flag="red"

print()

print('Parameters: ')

print('i. All Cases in A Department')

print('ii. Specifc Investigating Officer')

print ('iii. Go Back')

print()

cp=str(input('Enter your Choice: '))

if cp=="i":

AllCaseDpt()

elif cp=="ii":

print()

ins=str(input("Enter the Investigating Offcier: "))

c=0

for i in d:

if d[i]["Investigating Officer"].upper()==ins.upper():

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print (ins,"is not the investigating officer of any case")

elif cp=="iii":

flag='green'

adisplay()

else:

print ("Invalid Option")

adisplaydpt()

def adisplay():

global flag

flag="red"

print()

print('Parameters: ')

print('i. File No.')

print('ii. Status')

print('iii. Department')

print ('iv. All Cases')

print ('v. My Cases')

print ('vi. Go Back')

print()

cp=str(input('Enter your Choice: '))

if cp=='i':

Fno()

elif cp=="ii":

adispalystatus()

elif cp=="iii":

adisplaydpt()

elif cp=="iv":

for i in d:

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

elif cp=="v":

c=0

if 'ANALYST' not in user[3]:

for i in d:

if d[i]["Investigating Officer"].upper()==user[1]:

c+=1

print()

print("File No.:",i)

for j in d[i]:

print (j,":",d[i][j], end="\n")

else:

if c==0:

print ("You currently have 0 cases registered.")

elif 'ANALYST' in user[3]:

for i in d:

for j in d[i]["Analyst(s)"]:

if j==user[1]:

c+=1

print()

print("File No.:",i)

for k in d[i]:

print (k,":",d[i][k], end="\n")

else:

if c==0:

print ("You currently have 0 cases registered.")

elif cp=="vi":

aoptions()

else:

print ("Invalid Choice")

adisplay()

def gdispalystatus():

global flag

flag="red"

print()

print('Parameters: ')

print('i. Ongoing')

print('ii. Solved')

print('iii. Unsolved')

print ('iv. Go Back')

print()

cp=str(input('Enter your Choice: '))

if cp=='i':

c=0

for i in d:

if d[i]["Status"].lower()=="ongoing":

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print ("No Ongoing Cases")

elif cp=='ii':

c=0

for i in d:

if d[i]["Status"].lower()=="solved":

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print ("No Solved Cases")

elif cp=="iii":

c=0

for i in d:

if d[i]["Status"].lower()=="unsolved":

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print ("No Unsolved Cases")

elif cp=="iv":

flag="green"

gdisplay()

else:

print ("Invalid Option")

gdispalystatus()

def gdisplaydpt():

global flag

global user

flag="red"

print()

print('Parameters: ')

print('i. All Cases in A Department')

print('ii. Specifc Investigating Officer')

print ('iii. Go Back')

print()

cp=str(input('Enter your Choice: '))

if cp=="i":

print()

dpt=str(input("Enter the Department: "))

c=0

for i in d:

if d[i]["Department"].upper()==dpt.upper():

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print (ins,"is not investigating any case")

elif cp=="ii":

print()

ins=str(input("Enter the Investigating Offcier: "))

c=0

for i in d:

if d[i]["Investigating Officer"].upper()==ins.upper():

c+=1

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

else:

if c==0:

print (ins,"is not the investigating officer of any case")

elif cp=="iii":

flag='green'

gdisplay()

else:

print ("Invalid Option")

gdisplaydpt()

def gdisplay():

global flag

flag="red"

print()

print('Parameters: ')

print('i. File No.')

print('ii. Status')

print('iii. Department')

print ('iv. All Cases')

print ('v. Go Back')

print()

cp=str(input('Enter your Choice: '))

if cp=='i':

Fno()

elif cp=="ii":

gdispalystatus()

elif cp=="iii":

gdisplaydpt()

elif cp=="iv":

for i in d:

print()

print("File No.:",i)

for j in d[i]:

print(j,":",d[i][j],end="\n")

elif cp=="v":

goptions()

else:

print ("Invalid Choice")

gdisplay()

def updateinfile():

f=open('GBI\_Records.dat','wb+')

for k in d:

x={}

x[k]=d[k]

pickle.dump(x,f)

f.close()

def aoptions():

global flag

while True:

print()

print ("Please select the option you wish to execute")

print()

print('1. Update - Status | Accused | Suspects | Evidence | Case Handlers')

print('2. Criminal History')

print('3. Details of Similar Crimes - Charges | Details of Crime | Evidence')

print('4. Add New Case')

print('5. Display Details - File No. | Status | Department | All Cases | My Cases')

print('6. Exit')

print()

c=str(input('Enter your Choice: '))

if c=="1":

update()

updateinfile()

if flag=="green":

break

elif c=="2":

history()

if flag=="green":

break

elif c=="3":

similar()

if flag=="green":

break

elif c=="4":

NewCase()

updateinfile()

if flag=="green":

break

elif c=="5":

adisplay()

if flag=="green":

break

elif c=="6":

print ("Logging Out...")

print("Exiting Database")

break

else:

print("Invalid Option")

aoptions()

def goptions():

global flag

flag="red"

while True:

print()

print ("Please select the option you wish to execute")

print()

print('1. Criminal History')

print('2. Details of Similar Crimes - Charges | Details of Crime | Evidence')

print('3. Display Details - File No. | Status | Derpartment | All Cases')

print('4. Exit')

print()

c=str(input('Enter your Choice: '))

if c=="1":

history()

if flag=="green":

break

elif c=="2":

similar()

if flag=="green":

break

elif c=="3":

gdisplay()

if flag=="green":

break

elif c=="4":

print ("Exiting Database")

break

else:

print("Invalid Option")

goptions()

**Output**



A screenshot of a computer

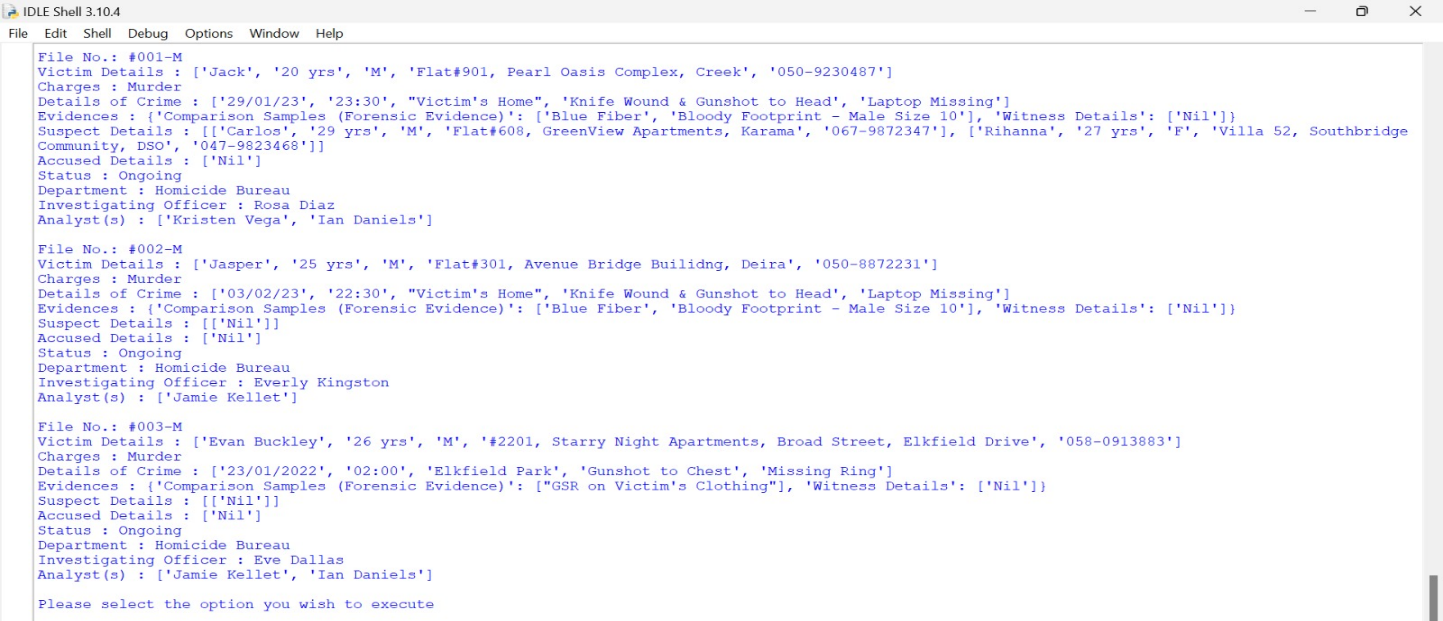
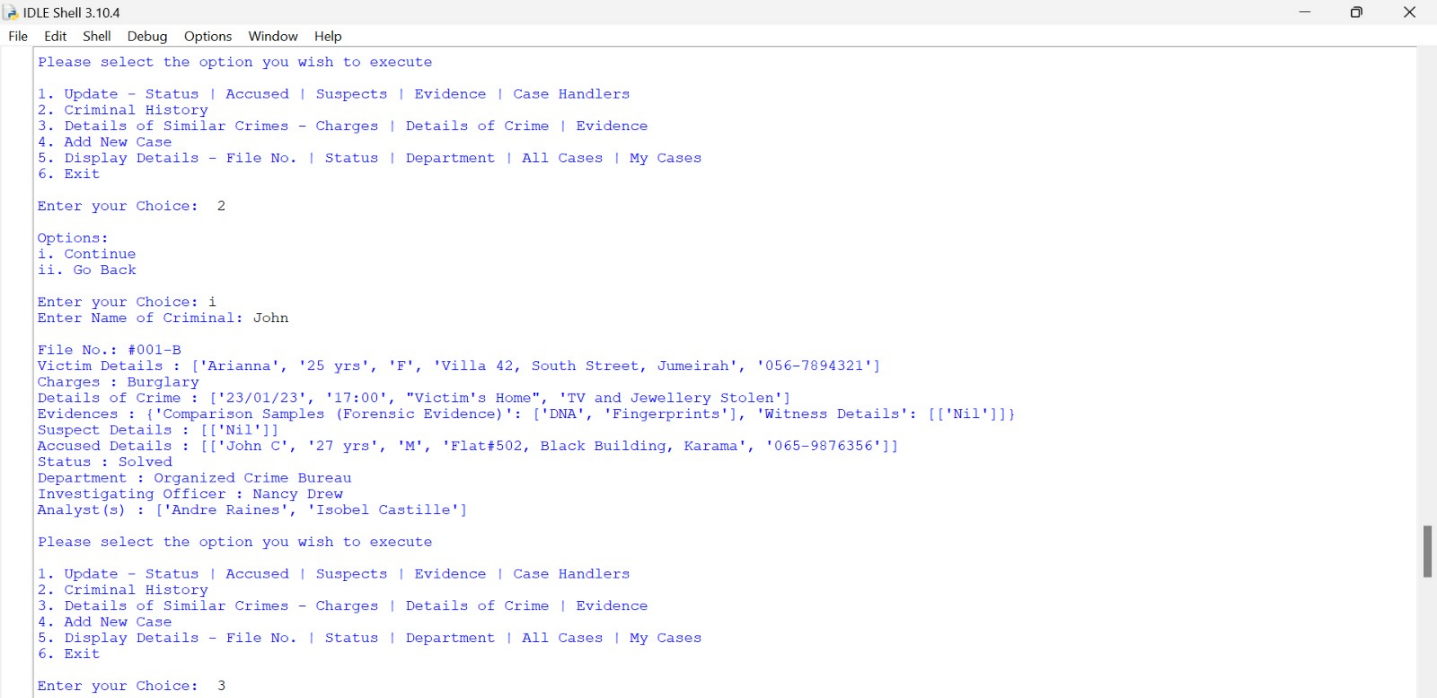
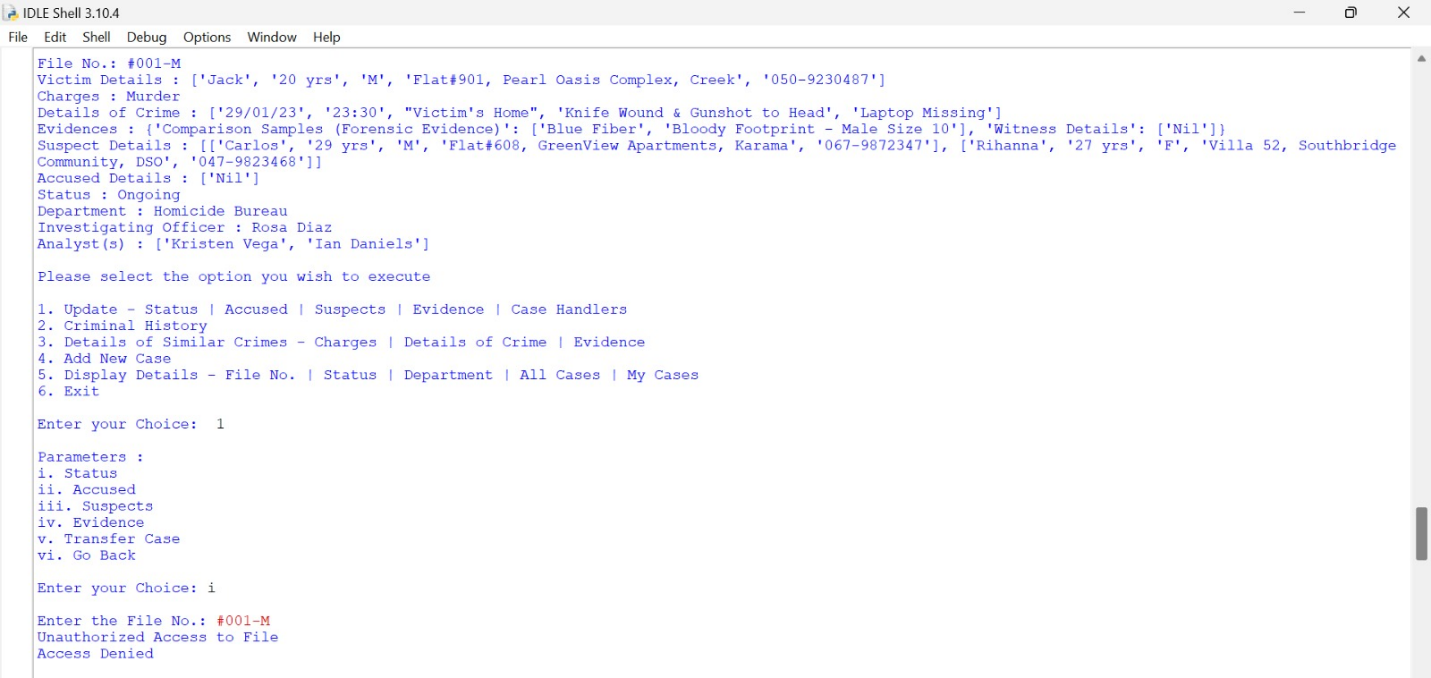
Description automatically generated A white background with blue text

Description automatically generated

A close-up of a white background

Description automatically generated A screenshot of a computer

Description automatically generated





**Bibliography**

1. <https://docs.python.org/3/library/time.html>
2. <https://www.geeksforgeeks.org/mysql-connector-python-module-in-python/>
3. <https://www.w3schools.com/python/python_mysql_getstarted.asp>
4. <https://www.w3schools.com/sql/sql_intro.asp>
5. <https://www.w3schools.com/sql/>
6. <https://www.w3schools.com/sql/sql_constraints.asp>
7. <https://www.w3schools.com/sql/sql_create_table.asp>
8. <https://stackoverflow.com/questions/174582/how-do-i-rename-a-column-in-a-database-table-using-sql>
9. <https://stackoverflow.com/questions/5938523/check-constraint-for-date>
10. <https://www.w3schools.com/sql/sql_default.asp>
11. <https://www.w3schools.com/SQl/sql_autoincrement.asp>
12. <https://database.guide/list-of-date-and-time-functions-in-sql-server-t-sql/>
13. <https://www.w3schools.com/SQL/sql_ref_drop_column.asp>
14. <https://www.w3schools.com/sqL/func_sqlserver_dateadd.asp>
15. <https://stackoverflow.com/questions/37441744/sql-rowcount-on-select-statement>