Applied Databases

Higher Diploma in Science in Data Analytics

Innovation and extra functionality

Part 1:
A function to clear the screen for user after a period of time

```
# https://www.geeksforgeeks.org/clear-screen-python/
# import only system from os
# import sleep to show output for some time period
# define our clear function
def clear():
    # sleep(2)
    # for windows
    if name == 'nt':
        _ = system('cls')
    # for mac and linux(here, os.name is 'posix')
    else:
        _ = system('clear')

    ProjectApp.display_menu()
```

Part 2:
A countdown function to display to user when output from choice will clear and return to the main menu

```
def countdown():
    print("\nClearing Data on Screen and Returning to Main Menu in:")
    for i in range(9,0,-1):
        # print number to screen then remove
        print(f"{i}", end="\r", flush=True)
        # pause program for 1 second
        sleep(1)
```

Part 3:
A function to check if files are present to run the app

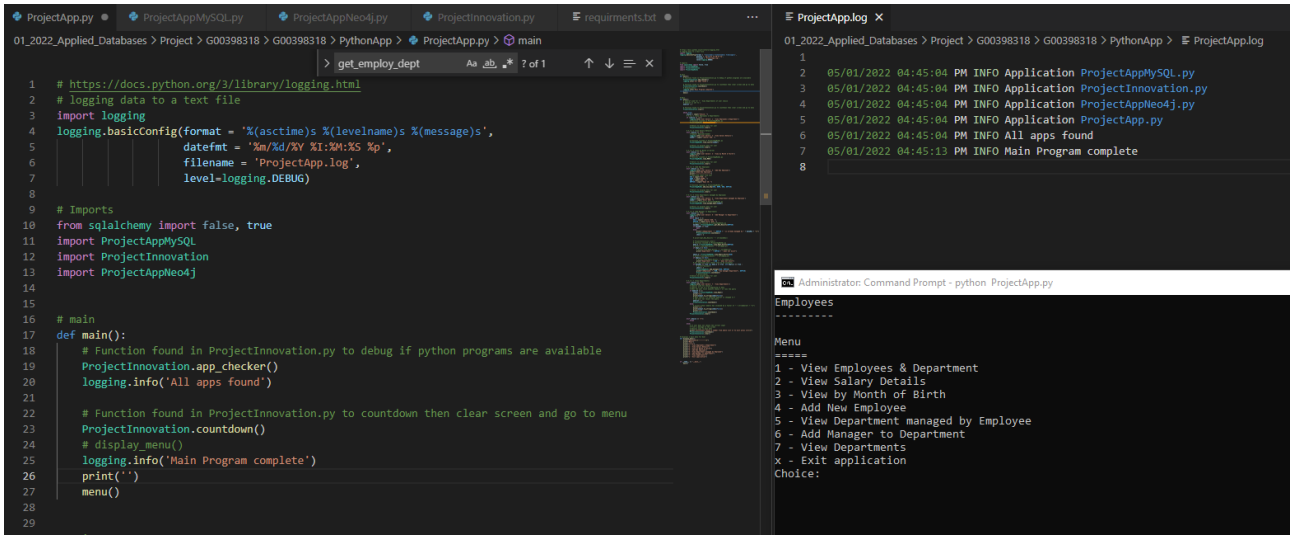
- *ProjectApp.py*
- *ProjectAppMySQL.py*
- *ProjectInnovation.py*
- *ProjectAppNeo4j.py*

```
# Function to check if all files required for the app are present in the same folder
# Assist in debugging for the app
def app_checker():
    # List of Files
```

```
appfiles = ["ProjectAppMySQL.py", "ProjectInnovation.py", "ProjectAppNeo4j.py"]
# Get currrent working directory
filelocation = os.path.dirname(__file__)
# Use list of file to check if that they are present
for file in appfiles:
    try:
        # try open file using path and filename
        # output to screen
        with open(filelocation + "\\" + file) as f:
            print("\nProject Apps File Found")
            print(file)
    except IOError:
        # Error if files not found outpur missing file name
        print("\nProject Apps File Not Found " + file)
        return false
return true
```

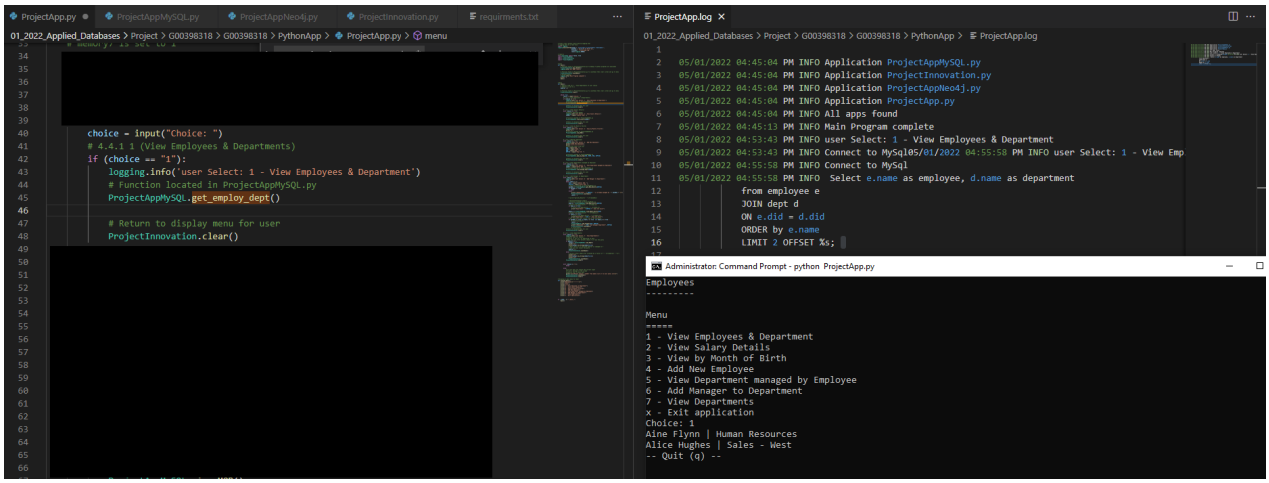
Part 4:
Logging the application to check if elements are working from the specification sheet (requires command prompt admin level privileges)
This is a great tool in python to log information for troubleshoot the application
All examples below show the output to the logs for each of the examples from the specification sheet

Program loads required programs. Outputs to logging file

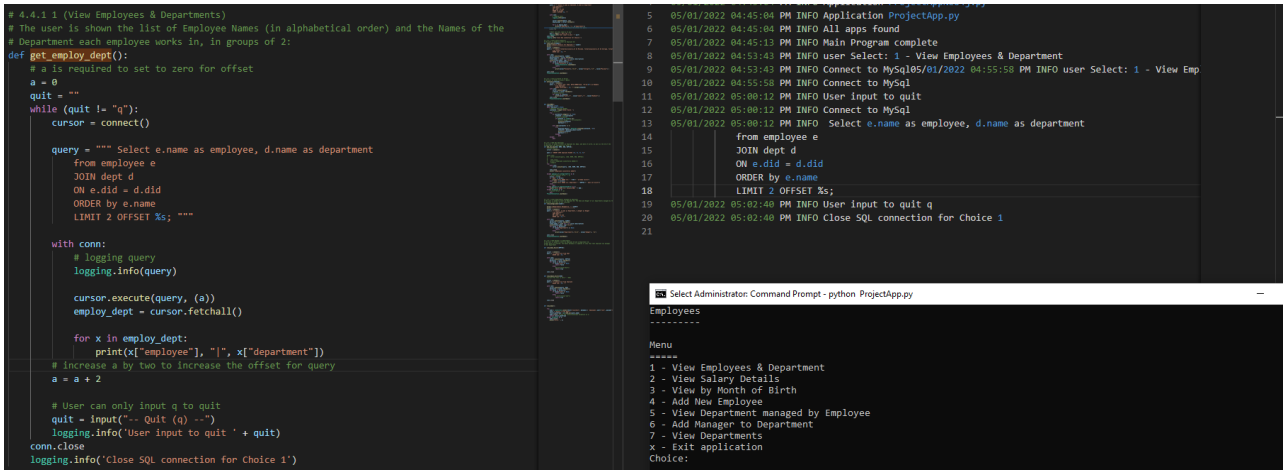


##

Select Choice 1

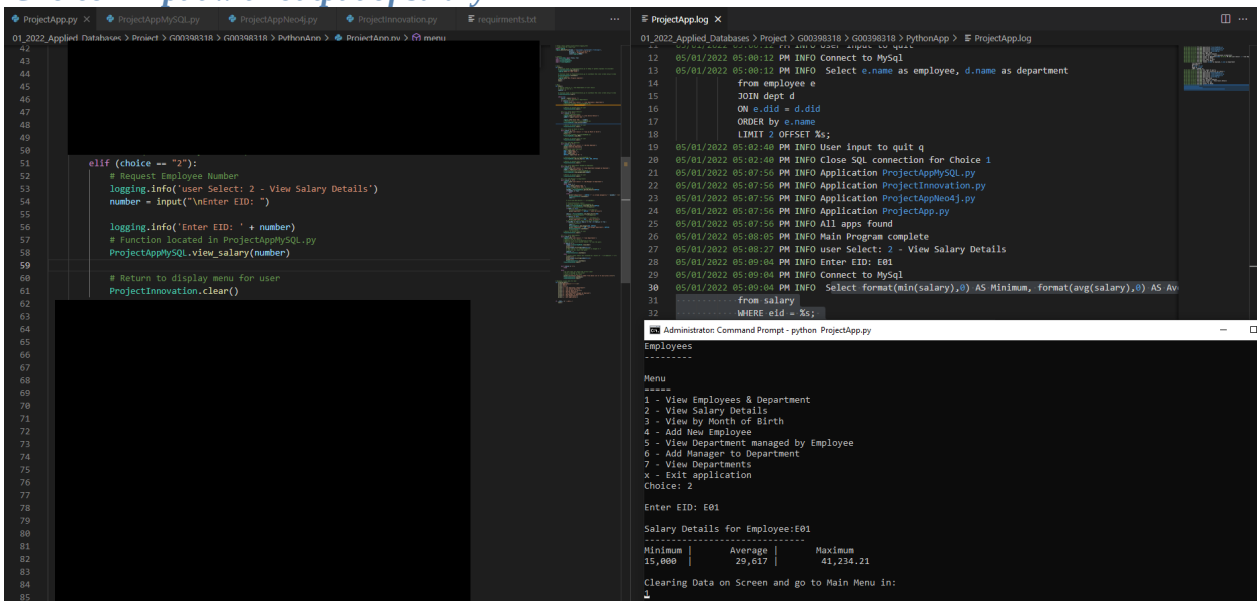


Query database and q to quit

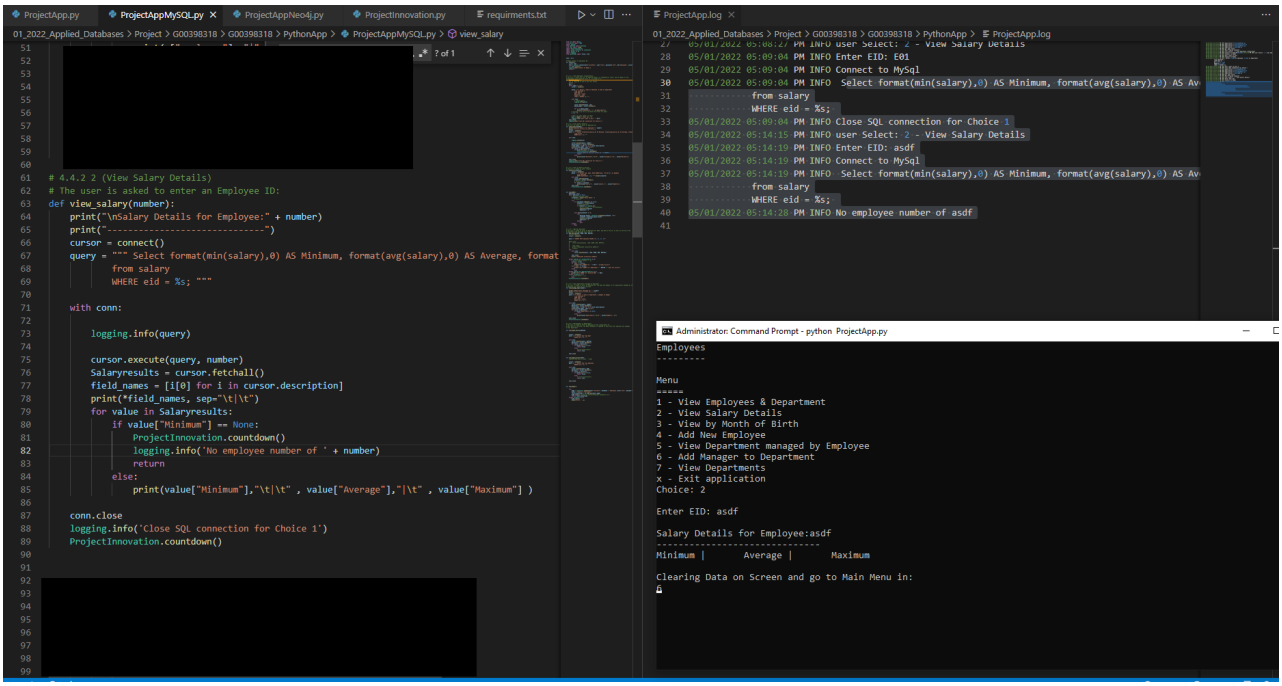


##

Choice 2 input with output of salary

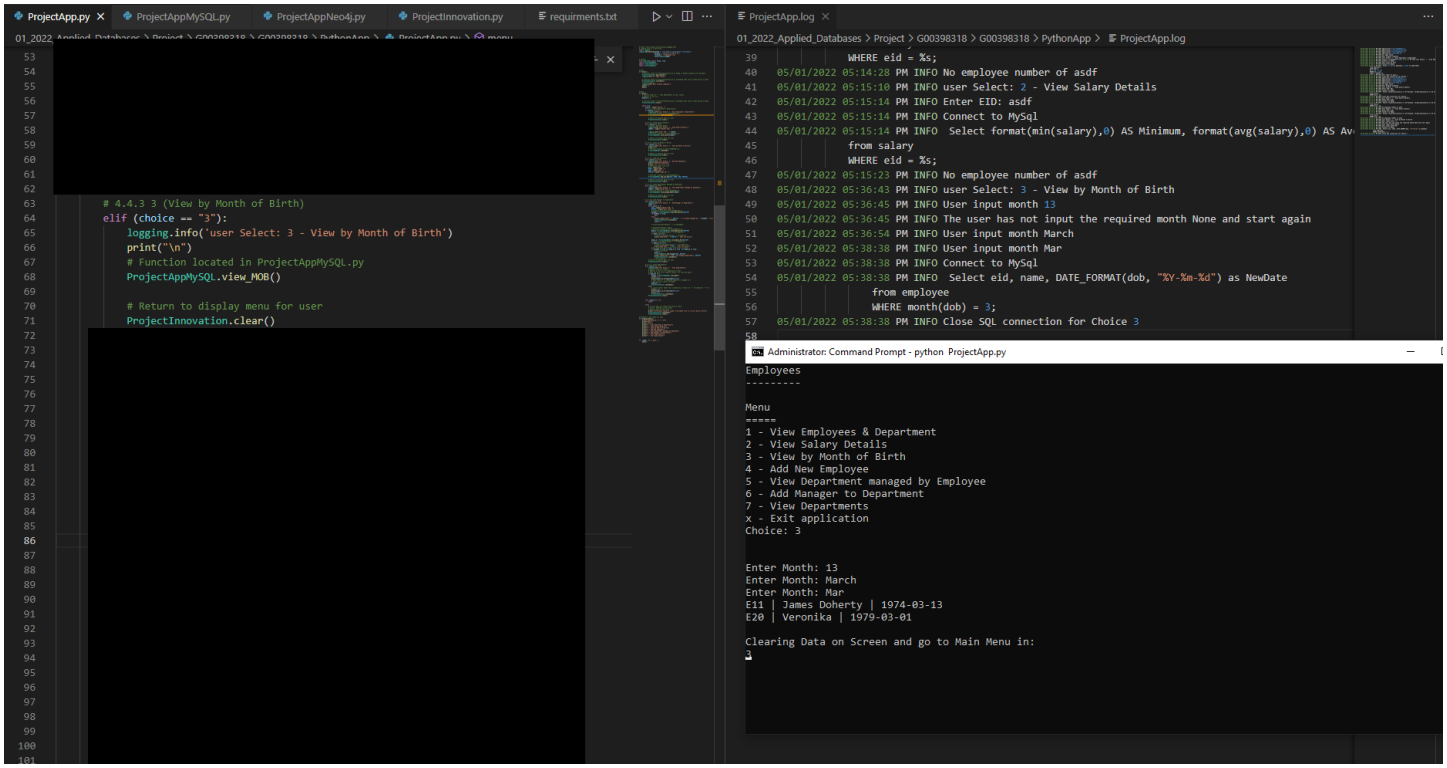


Records user input if non-existent Employee ID

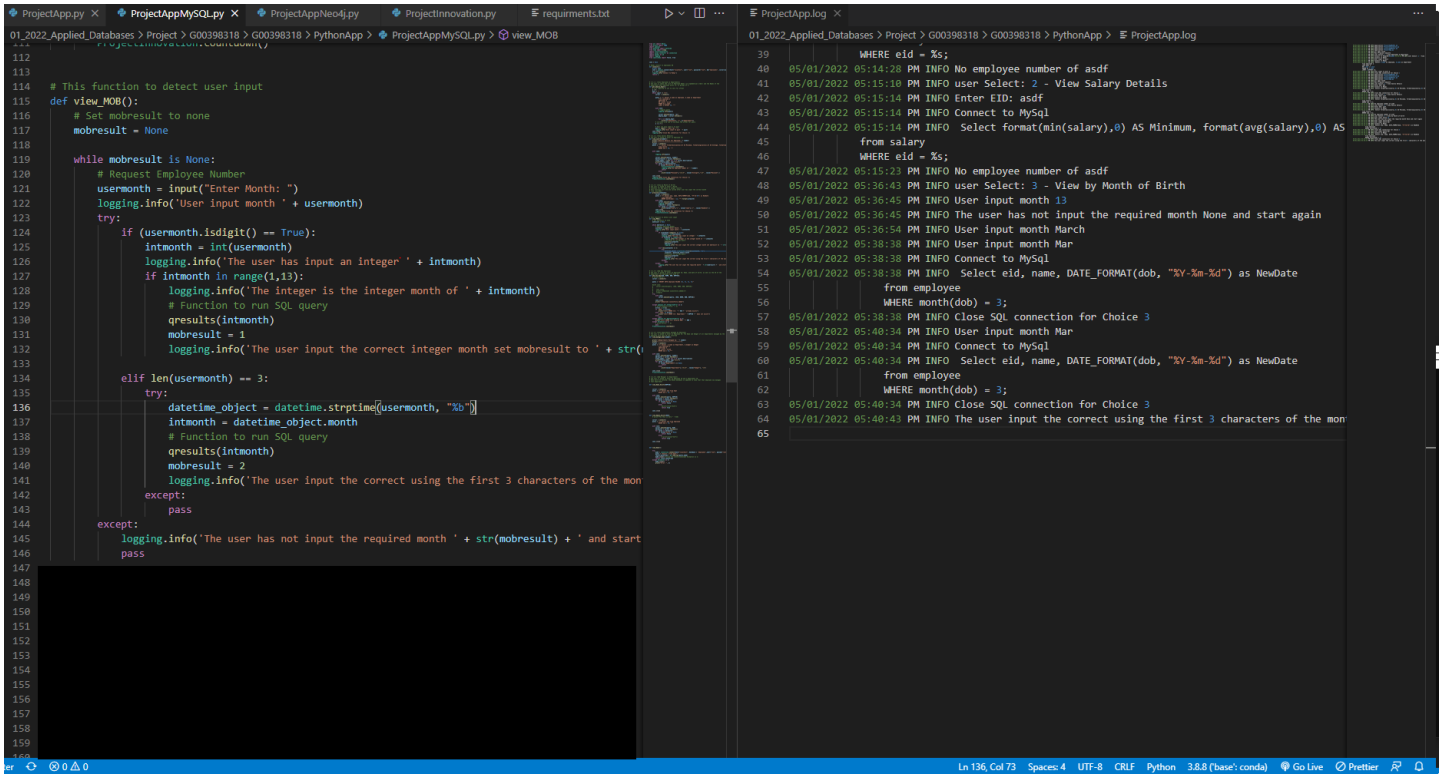


##

Choice 3
User input incorrect month format def view_MOB()



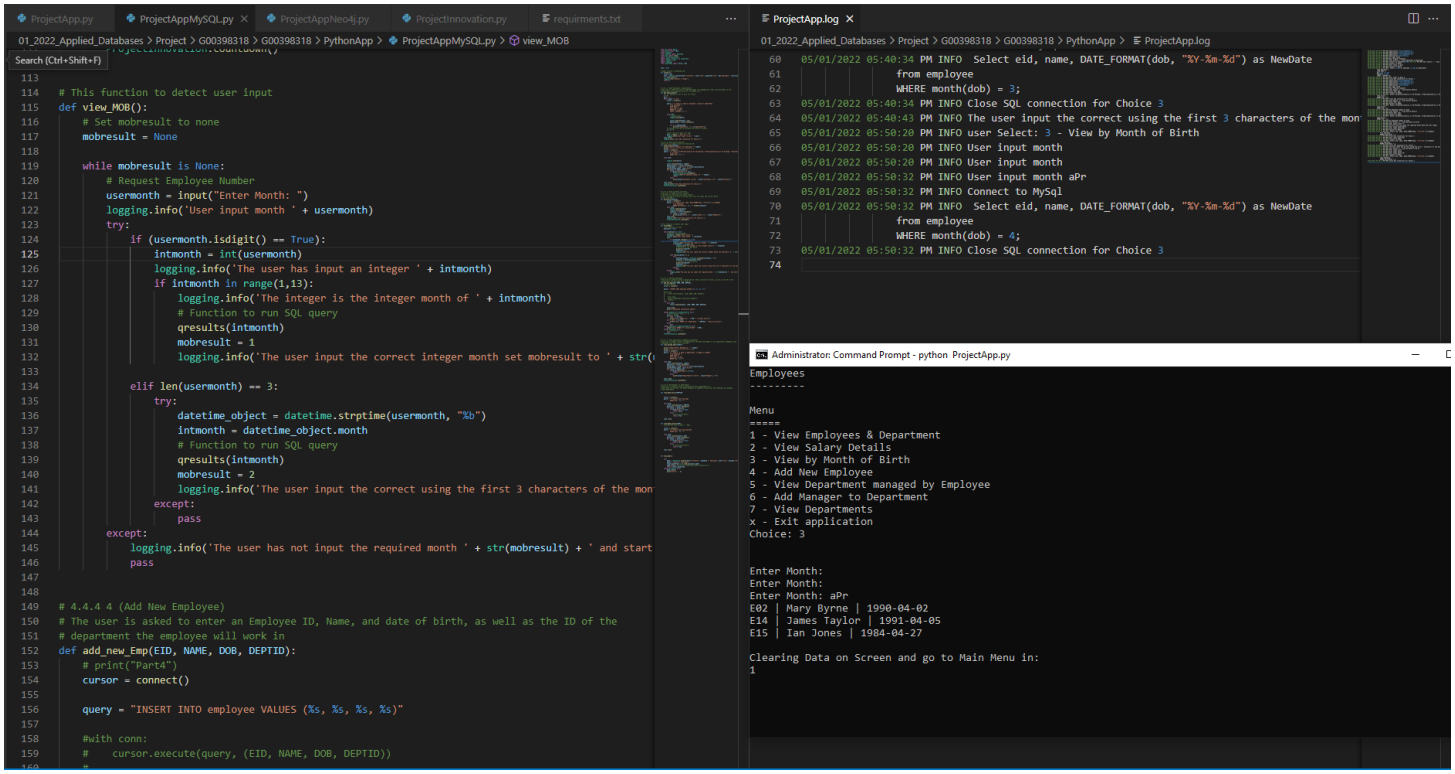
def view_MOB() test the user input is the correct format for month



```
# This function to detect user input
def view_MOB():
    # Set mobresult to none
    mobresult = None

    while mobresult is None:
        # Request Employee Number
        usermonth = input("Enter Month: ")
        logging.info('User input month ' + usermonth)
        try:
            if (usermonth.isdigit() == True):
                intmonth = int(usermonth)
                logging.info('The user has input an integer ' + intmonth)
                if intmonth in range(1,13):
                    logging.info('The integer is the integer month of ' + intmonth)
                    # Function to run SQL query
                    qresults(intmonth)
                    mobresult = 1
                    logging.info('The user input the correct integer month set mobresult to ' + str(

            elif len(usermonth) == 3:
                try:
                    datetime_object = datetime.strptime(usermonth, "%b")
                    intmonth = datetime_object.month
                    # Function to run SQL query
                    qresults(intmonth)
                    mobresult = 2
                    logging.info('The user input the correct using the first 3 characters of the mon
                except:
                    pass
        except:
            logging.info('The user has not input the required month ' + str(mobresult) + ' and start
            pass
```

```
                WHERE eid = %s;
05/01/2022 05:14:28 PM INFO No employee number of asdf
05/01/2022 05:15:10 PM INFO user Select: 2 - View Salary Details
05/01/2022 05:15:14 PM INFO Enter EID: asdf
05/01/2022 05:15:14 PM INFO Connect to MySql
05/01/2022 05:15:14 PM INFO  Select format(min(salary),0) AS Minimum, format(avg(salary),0) AS
                from salary
                WHERE eid = %s;
05/01/2022 05:15:23 PM INFO No employee number of asdf
05/01/2022 05:36:43 PM INFO user Select: 3 - View by Month of Birth
05/01/2022 05:36:45 PM INFO User input month 13
05/01/2022 05:36:45 PM INFO The user has not input the required month None and start again
05/01/2022 05:36:54 PM INFO User input month March
05/01/2022 05:38:38 PM INFO User input month Mar
05/01/2022 05:38:38 PM INFO Connect to MySql
05/01/2022 05:38:38 PM INFO  Select eid, name, DATE_FORMAT(dob, "%Y-%m-%d") as NewDate
                from employee
                WHERE month(dob) = 3;
05/01/2022 05:38:38 PM INFO Close SQL connection for Choice 3
05/01/2022 05:40:34 PM INFO User input month Mar
05/01/2022 05:40:34 PM INFO Connect to MySql
05/01/2022 05:40:34 PM INFO  Select eid, name, DATE_FORMAT(dob, "%Y-%m-%d") as NewDate
                from employee
                WHERE month(dob) = 3;
05/01/2022 05:40:34 PM INFO Close SQL connection for Choice 3
05/01/2022 05:40:43 PM INFO The user input the correct using the first 3 characters of the mon
```

Input month as aPr

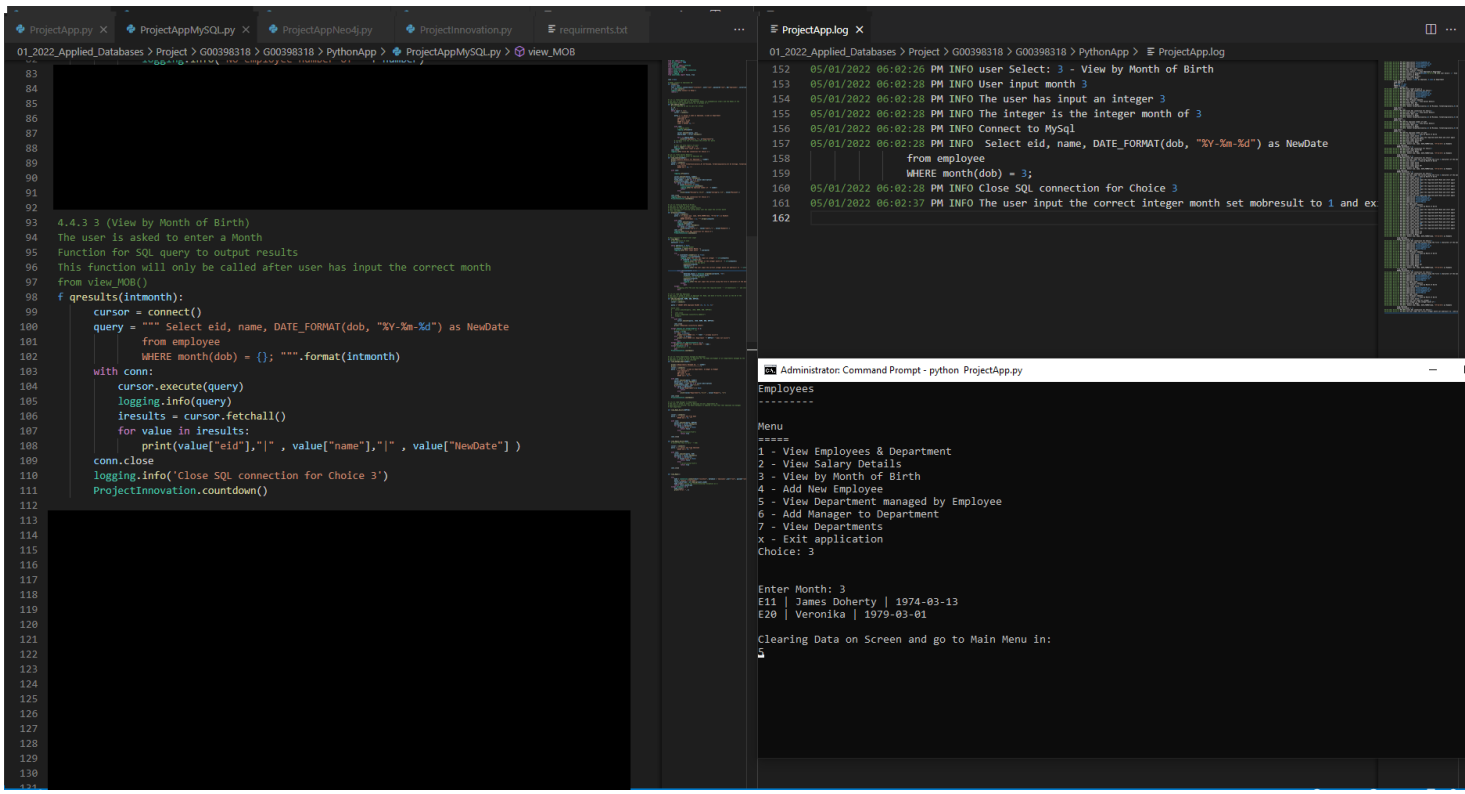


```
# This function to detect user input
def view_MOB():
    # Set mobresult to none
    mobresult = None

    while mobresult is None:
        # Request Employee Number
        usermonth = input("Enter Month: ")
        logging.info('User input month ' + usermonth)
        try:
            if (usermonth.isdigit() == True):
                intmonth = int(usermonth)
                logging.info('The user has input an integer ' + intmonth)
                if intmonth in range(1,13):
                    logging.info('The integer is the integer month of ' + intmonth)
                    # Function to run SQL query
                    qresults(intmonth)
                    mobresult = 1
                    logging.info('The user input the correct integer month set mobresult to ' + str(

            elif len(usermonth) == 3:
                try:
                    datetime_object = datetime.strptime(usermonth, "%b")
                    intmonth = datetime_object.month
                    # Function to run SQL query
                    qresults(intmonth)
                    mobresult = 2
                    logging.info('The user input the correct using the first 3 characters of the mon
                except:
                    pass
        except:
            logging.info('The user has not input the required month ' + str(mobresult) + ' and start
            pass


# 4.4.4 4 (Add New Employee)
# The user is asked to enter an Employee ID, Name, and date of birth, as well as the ID of the
# department the employee will work in
def add_new_Emp(EID, NAME, DOB, DEPTID):
    # print("Part4")
    cursor = connect()

    query = "INSERT INTO employee VALUES (%s, %s, %s, %s)"

    #with conn:
    #    cursor.execute(query, (EID, NAME, DOB, DEPTID))
```

```
05/01/2022 05:40:34 PM INFO  Select eid, name, DATE_FORMAT(dob, "%Y-%m-%d") as NewDate
                from employee
                WHERE month(dob) = 3;
05/01/2022 05:40:34 PM INFO Close SQL connection for Choice 3
05/01/2022 05:40:43 PM INFO The user input the correct using the first 3 characters of the mon
05/01/2022 05:50:20 PM INFO user Select: 3 - View by Month of Birth
05/01/2022 05:50:20 PM INFO User input month
05/01/2022 05:50:20 PM INFO User input month
05/01/2022 05:50:32 PM INFO User input month aPr
05/01/2022 05:50:32 PM INFO Connect to MySql
05/01/2022 05:50:32 PM INFO  Select eid, name, DATE_FORMAT(dob, "%Y-%m-%d") as NewDate
                from employee
                WHERE month(dob) = 4;
05/01/2022 05:50:32 PM INFO Close SQL connection for Choice 3
```

```
Administrator: Command Prompt - python  ProjectApp.py
Employees
---------

Menu
=====
1 - View Employees & Department
2 - View Salary Details
3 - View by Month of Birth
4 - Add New Employee
5 - View Department managed by Employee
6 - Add Manager to Department
7 - View Departments
x - Exit application
Choice: 3


Enter Month:
Enter Month:
Enter Month: aPr
E02 | Mary Byrne | 1990-04-02
E14 | James Taylor | 1991-04-05
E15 | Ian Jones | 1984-04-27

Clearing Data on Screen and go to Main Menu in:
1
```

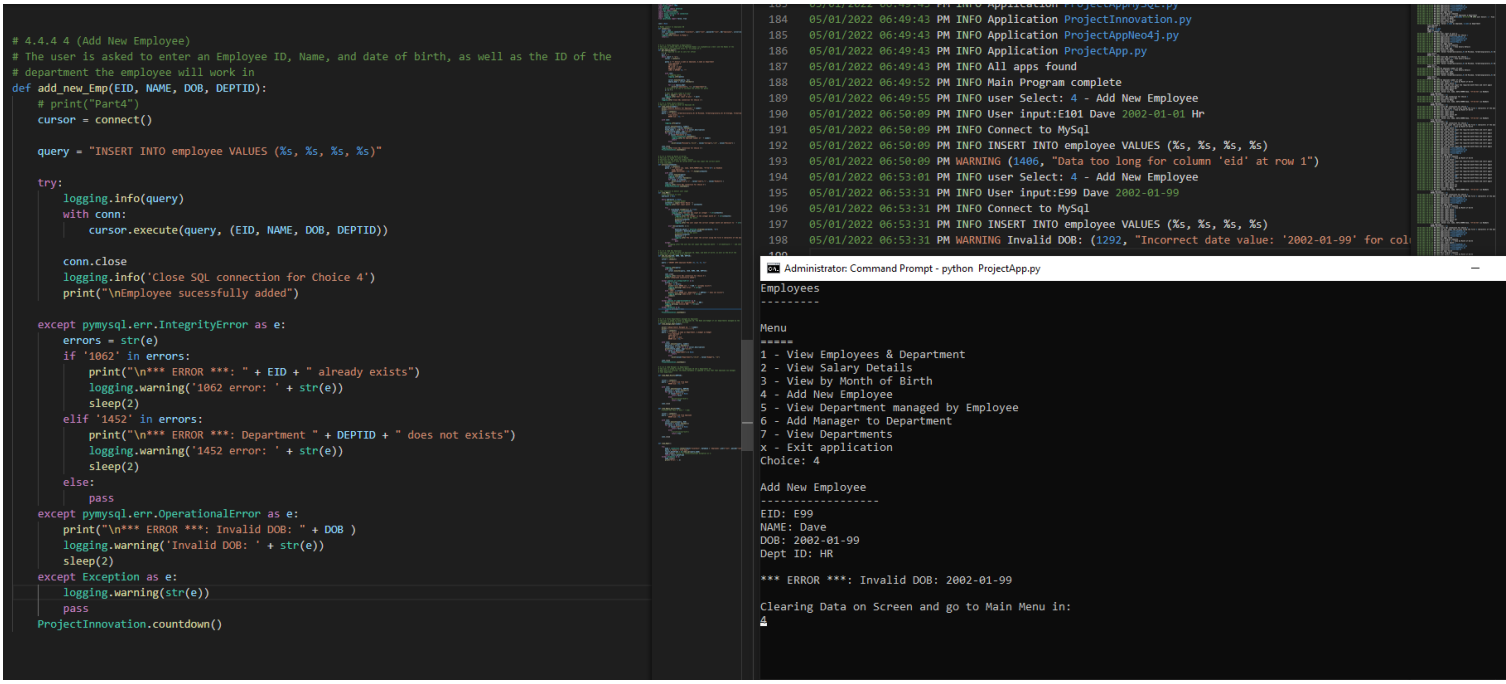
qresults(intmonth) function is run when user inputs the correct format.


ProjectApp.py
ProjectAppMySQL.py
ProjectAppNeo4j.py
ProjectInnovation.py
requirments.txt
01_2022_Applied_Databases > Project > G00398318 > G00398318 > PythonApp > ProjectAppMySQL.py > view_MOB
4.4.3 3 (View by Month of Birth)
The user is asked to enter a Month
Function for SQL query to output results
This function will only be called after user has input the correct month
from view_MOB()
f qresults(intmonth):
cursor = connect()
query = """ Select eid, name, DATE_FORMAT(dob, "%Y-%m-%d") as NewDate
from employee
WHERE month(dob) = {}; """.format(intmonth)
with conn:
cursor.execute(query)
logging.info(query)
iresults = cursor.fetchall()
for value in iresults:
print(value["eid"],"|" , value["name"],"|" , value["NewDate"])
conn.close
logging.info('Close SQL connection for Choice 3')
ProjectInnovation.countdown()
ProjectApp.log
01_2022_Applied_Databases > Project > G00398318 > G00398318 > PythonApp > ProjectApp.log
05/01/2022 06:02:26 PM INFO user Select: 3 - View by Month of Birth
05/01/2022 06:02:28 PM INFO User input month 3
05/01/2022 06:02:28 PM INFO The user has input an integer 3
05/01/2022 06:02:28 PM INFO The integer is the integer month of 3
05/01/2022 06:02:28 PM INFO Connect to MySql
05/01/2022 06:02:28 PM INFO Select eid, name, DATE_FORMAT(dob, "%Y-%m-%d") as NewDate
from employee
WHERE month(dob) = 3;
05/01/2022 06:02:28 PM INFO Close SQL connection for Choice 3
05/01/2022 06:02:37 PM INFO The user input the correct integer month set mobresult to 1 and ex
Administrator: Command Prompt - python ProjectApp.py
Employees
Menu
1 - View Employees & Department
2 - View Salary Details
3 - View by Month of Birth
4 - Add New Employee
5 - View Department managed by Employee
6 - Add Manager to Department
7 - View Departments
x - Exit application
Choice: 3
Enter Month: 3
E11 | James Doherty | 1974-03-13
E20 | Veronika | 1979-03-01
Clearing Data on Screen and go to Main Menu in:
5

##

Choice 4

Error on Date

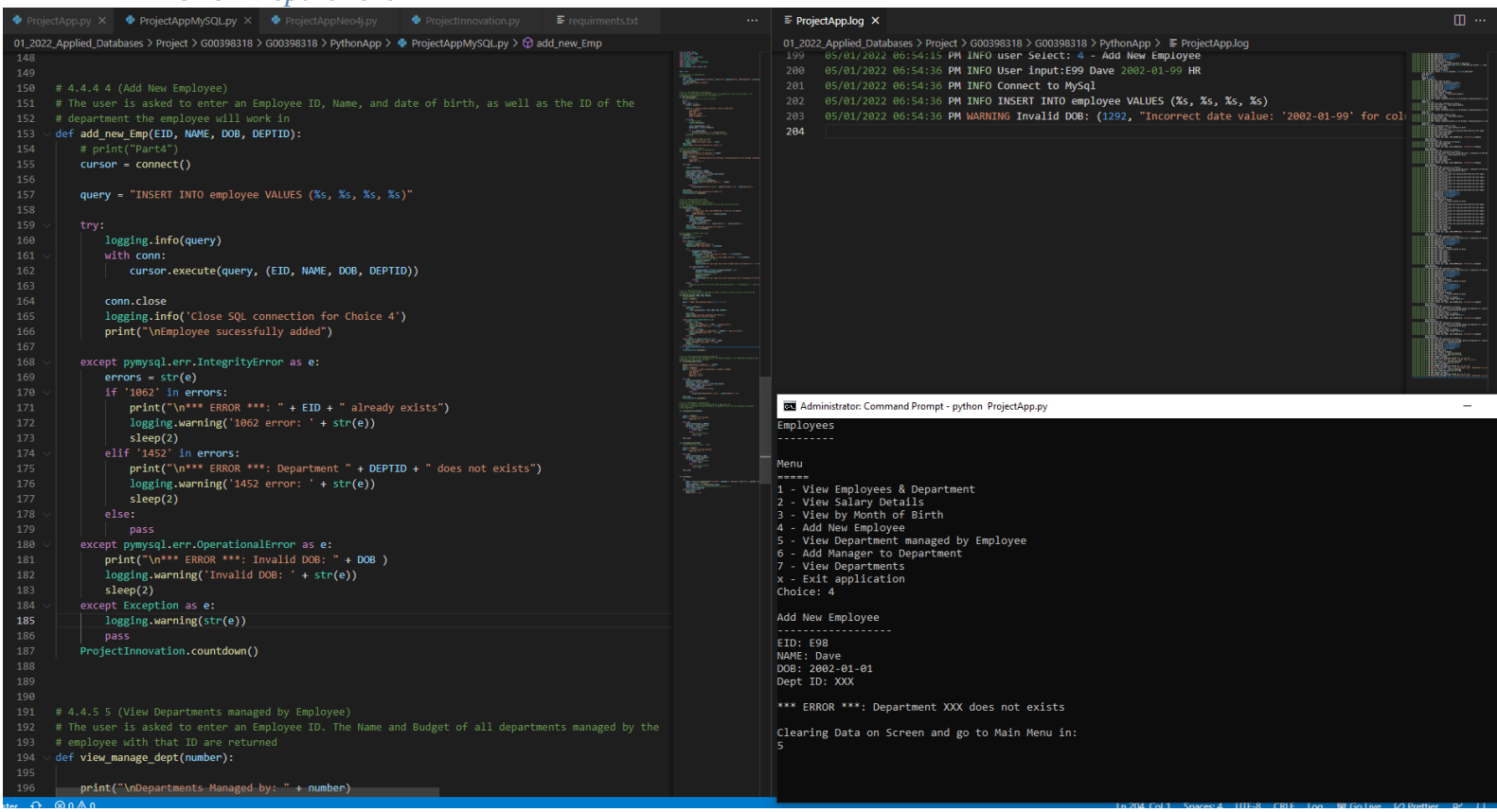


```
# 4.4.4 4 (Add New Employee)
# The user is asked to enter an Employee ID, Name, and date of birth, as well as the ID of the
# department the employee will work in
def add_new_Emp(EID, NAME, DOB, DEPTID):
    # print("Part4")
    cursor = connect()

    query = "INSERT INTO employee VALUES (%s, %s, %s, %s)"

    try:
        logging.info(query)
        with conn:
            cursor.execute(query, (EID, NAME, DOB, DEPTID))

        conn.close
        logging.info('Close SQL connection for Choice 4')
        print("\nEmployee sucessfully added")

    except pymysql.err.IntegrityError as e:
        errors = str(e)
        if '1062' in errors:
            print("\n*** ERROR ***: " + EID + " already exists")
            logging.warning('1062 error: ' + str(e))
            sleep(2)
        elif '1452' in errors:
            print("\n*** ERROR ***: Department " + DEPTID + " does not exists")
            logging.warning('1452 error: ' + str(e))
            sleep(2)
        else:
            pass
    except pymysql.err.OperationalError as e:
        print("\n*** ERROR ***: Invalid DOB: " + DOB )
        logging.warning('Invalid DOB: ' + str(e))
        sleep(2)
    except Exception as e:
        logging.warning(str(e))
        pass
    ProjectInnovation.countdown()
```

```
05/01/2022 06:49:43 PM INFO Application ProjectInnovation.py
05/01/2022 06:49:43 PM INFO Application ProjectAppNeo4j.py
05/01/2022 06:49:43 PM INFO Application ProjectApp.py
05/01/2022 06:49:43 PM INFO All apps found
05/01/2022 06:49:52 PM INFO Main Program complete
05/01/2022 06:49:55 PM INFO user Select: 4 - Add New Employee
05/01/2022 06:50:09 PM INFO User input:E101 Dave 2002-01-01 Hr
05/01/2022 06:50:09 PM INFO Connect to MySql
05/01/2022 06:50:09 PM INFO INSERT INTO employee VALUES (%s, %s, %s, %s)
05/01/2022 06:50:09 PM WARNING (1406, "Data too long for column 'eid' at row 1")
05/01/2022 06:53:01 PM INFO user Select: 4 - Add New Employee
05/01/2022 06:53:31 PM INFO User input:E99 Dave 2002-01-99
05/01/2022 06:53:31 PM INFO Connect to MySql
05/01/2022 06:53:31 PM INFO INSERT INTO employee VALUES (%s, %s, %s, %s)
05/01/2022 06:53:31 PM WARNING Invalid DOB: (1292, "Incorrect date value: '2002-01-99' for col
```

```
Administrator: Command Prompt - python  ProjectApp.py

Employees
---------

Menu
=====
1 - View Employees & Department
2 - View Salary Details
3 - View by Month of Birth
4 - Add New Employee
5 - View Department managed by Employee
6 - Add Manager to Department
7 - View Departments
x - Exit application
Choice: 4

Add New Employee
-----------------
EID: E99
NAME: Dave
DOB: 2002-01-99
Dept ID: HR

*** ERROR ***: Invalid DOB: 2002-01-99

Clearing Data on Screen and go to Main Menu in:
4
```

Error on Department



```
# 4.4.4 4 (Add New Employee)
# The user is asked to enter an Employee ID, Name, and date of birth, as well as the ID of the
# department the employee will work in
def add_new_Emp(EID, NAME, DOB, DEPTID):
    # print("Part4")
    cursor = connect()

    query = "INSERT INTO employee VALUES (%s, %s, %s, %s)"

    try:
        logging.info(query)
        with conn:
            cursor.execute(query, (EID, NAME, DOB, DEPTID))

        conn.close
        logging.info('Close SQL connection for Choice 4')
        print("\nEmployee sucessfully added")

    except pymysql.err.IntegrityError as e:
        errors = str(e)
        if '1062' in errors:
            print("\n*** ERROR ***: " + EID + " already exists")
            logging.warning('1062 error: ' + str(e))
            sleep(2)
        elif '1452' in errors:
            print("\n*** ERROR ***: Department " + DEPTID + " does not exists")
            logging.warning('1452 error: ' + str(e))
            sleep(2)
        else:
            pass
    except pymysql.err.OperationalError as e:
        print("\n*** ERROR ***: Invalid DOB: " + DOB )
        logging.warning('Invalid DOB: ' + str(e))
        sleep(2)
    except Exception as e:
        logging.warning(str(e))
        pass
    ProjectInnovation.countdown()


# 4.4.5 5 (View Departments managed by Employee)
# The user is asked to enter an Employee ID. The Name and Budget of all departments managed by the
# employee with that ID are returned
def view_manage_dept(number):

    print("\nDepartments Managed by: " + number)
```

```
05/01/2022 06:54:15 PM INFO user Select: 4 - Add New Employee
05/01/2022 06:54:36 PM INFO User input:E99 Dave 2002-01-99 HR
05/01/2022 06:54:36 PM INFO Connect to MySql
05/01/2022 06:54:36 PM INFO INSERT INTO employee VALUES (%s, %s, %s, %s)
05/01/2022 06:54:36 PM WARNING Invalid DOB: (1292, "Incorrect date value: '2002-01-99' for col
```

```
Administrator: Command Prompt - python  ProjectApp.py

Employees
---------

Menu
=====
1 - View Employees & Department
2 - View Salary Details
3 - View by Month of Birth
4 - Add New Employee
5 - View Department managed by Employee
6 - Add Manager to Department
7 - View Departments
x - Exit application
Choice: 4

Add New Employee
-----------------
EID: E98
NAME: Dave
DOB: 2002-01-01
Dept ID: XXX

*** ERROR ***: Department XXX does not exists

Clearing Data on Screen and go to Main Menu in:
5
```

Error Duplicate Employee

ProjectApp.py | ProjectAppMySQL.py | ProjectAppNeo4j.py | ProjectInnovation.py | requirments.txt

01_2022_Applied_Databases > Project > G00398318 > G00398318 > PythonApp > ProjectAppMySQL.py > add_new_Emp

```
148
149
150 # 4.4.4 4 (Add New Employee)
151 # The user is asked to enter an Employee ID, Name, and date of birth, as well as the ID of the
152 # department the employee will work in
153 def add_new_Emp(EID, NAME, DOB, DEPTID):
154     # print("Part4")
155     cursor = connect()
156
157     query = "INSERT INTO employee VALUES (%s, %s, %s, %s)"
158
159     try:
160         logging.info(query)
161         with conn:
162             cursor.execute(query, (EID, NAME, DOB, DEPTID))
163
164         conn.close
165         logging.info('Close SQL connection for Choice 4')
166         print("\nEmployee sucessfully added")
167
168     except pymysql.err.IntegrityError as e:
169         errors = str(e)
170         if '1062' in errors:
171             print("\n*** ERROR ***: " + EID + " already exists")
172             logging.warning('1062 error: ' + str(e))
173             sleep(2)
174         elif '1452' in errors:
175             print("\n*** ERROR ***: Department " + DEPTID + " does not exists")
176             logging.warning('1452 error: ' + str(e))
177             sleep(2)
178         else:
179             pass
180     except pymysql.err.OperationalError as e:
181         print("\n*** ERROR ***: Invalid DOB: " + DOB )
182         logging.warning('Invalid DOB: ' + str(e))
183         sleep(2)
184     except Exception as e:
185         logging.warning(str(e))
186         pass
187     ProjectInnovation.countdown()
188
189
190
191 # 4.4.5 5 (View Departments managed by Employee)
192 # The user is asked to enter an Employee ID. The Name and Budget of all departments managed by the
193 # employee with that ID are returned
194 def view_manage_dept(number):
```

ProjectApp.log

01_2022_Applied_Databases > Project > G00398318 > G00398318 > PythonApp > ProjectApp.log

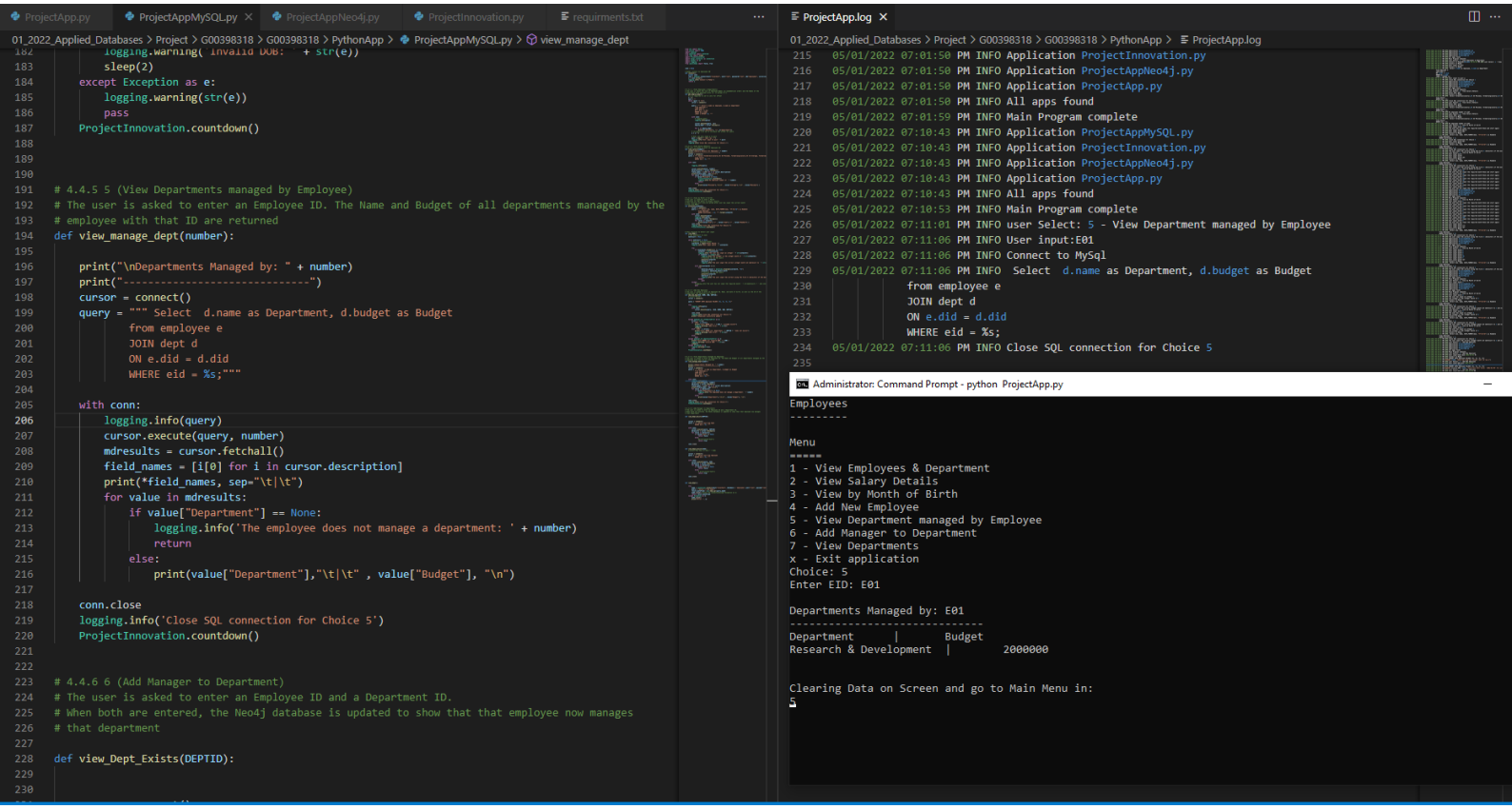
```
199 05/01/2022 06:54:15 PM INFO user Select: 4 - Add New Employee
200 05/01/2022 06:54:36 PM INFO User input:E99 Dave 2002-01-99 HR
201 05/01/2022 06:54:36 PM INFO Connect to MySql
202 05/01/2022 06:54:36 PM INFO INSERT INTO employee VALUES (%s, %s, %s, %s)
203 05/01/2022 06:54:36 PM WARNING Invalid DOB: (1292, "Incorrect date value: '2002-01-99' for col
204
```

Administrator: Command Prompt - python ProjectApp.py

```
Employees
---------

Menu
=====
1 - View Employees & Department
2 - View Salary Details
3 - View by Month of Birth
4 - Add New Employee
5 - View Department managed by Employee
6 - Add Manager to Department
7 - View Departments
x - Exit application
Choice: 4

Add New Employee
----------------
EID: E01
NAME: Dave
DOB: 2002-01-01
Dept ID: Hr

*** ERROR ***: E01 already exists

Clearing Data on Screen and go to Main Menu in:
5
```

##

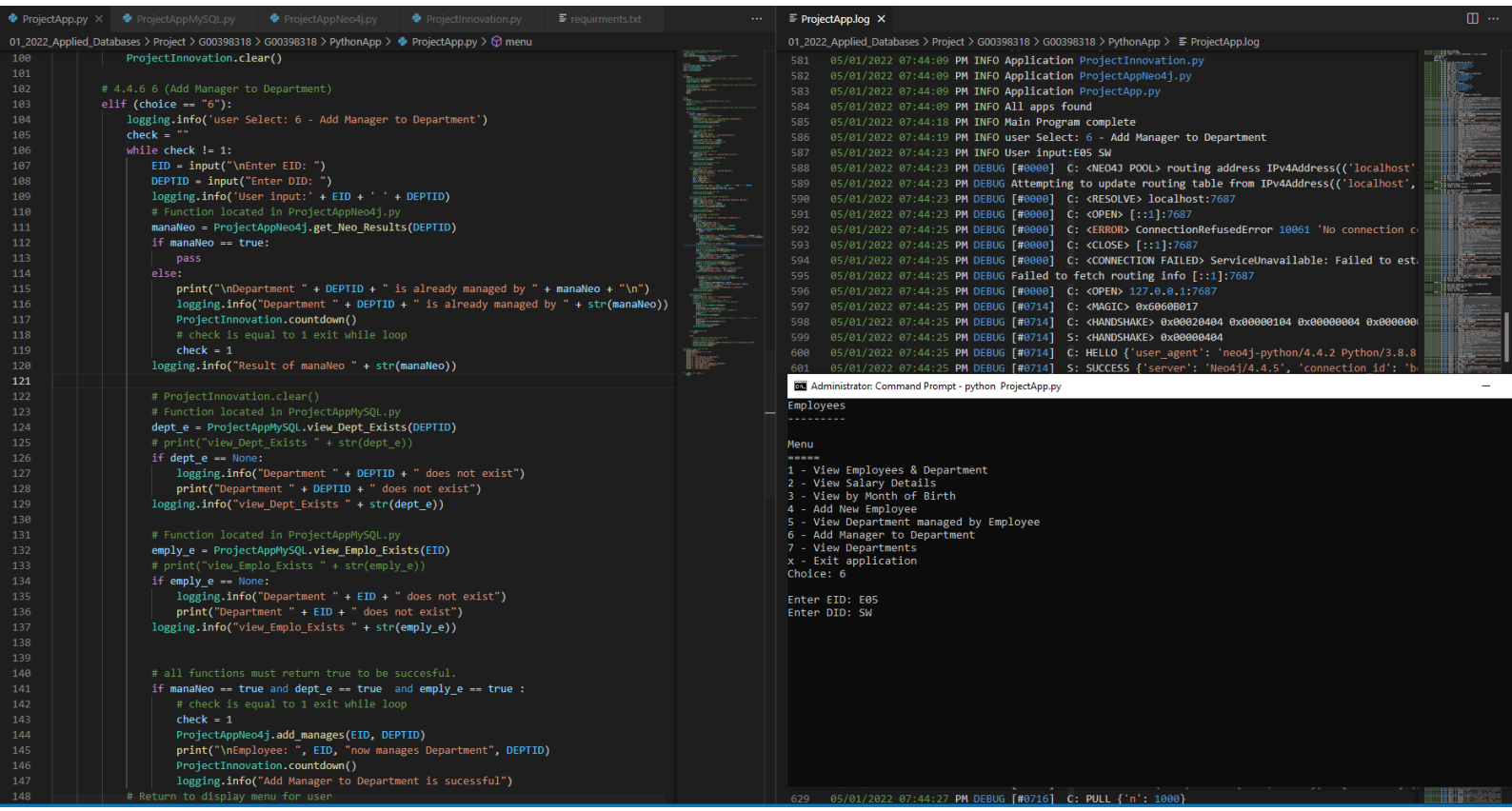
Choice 5



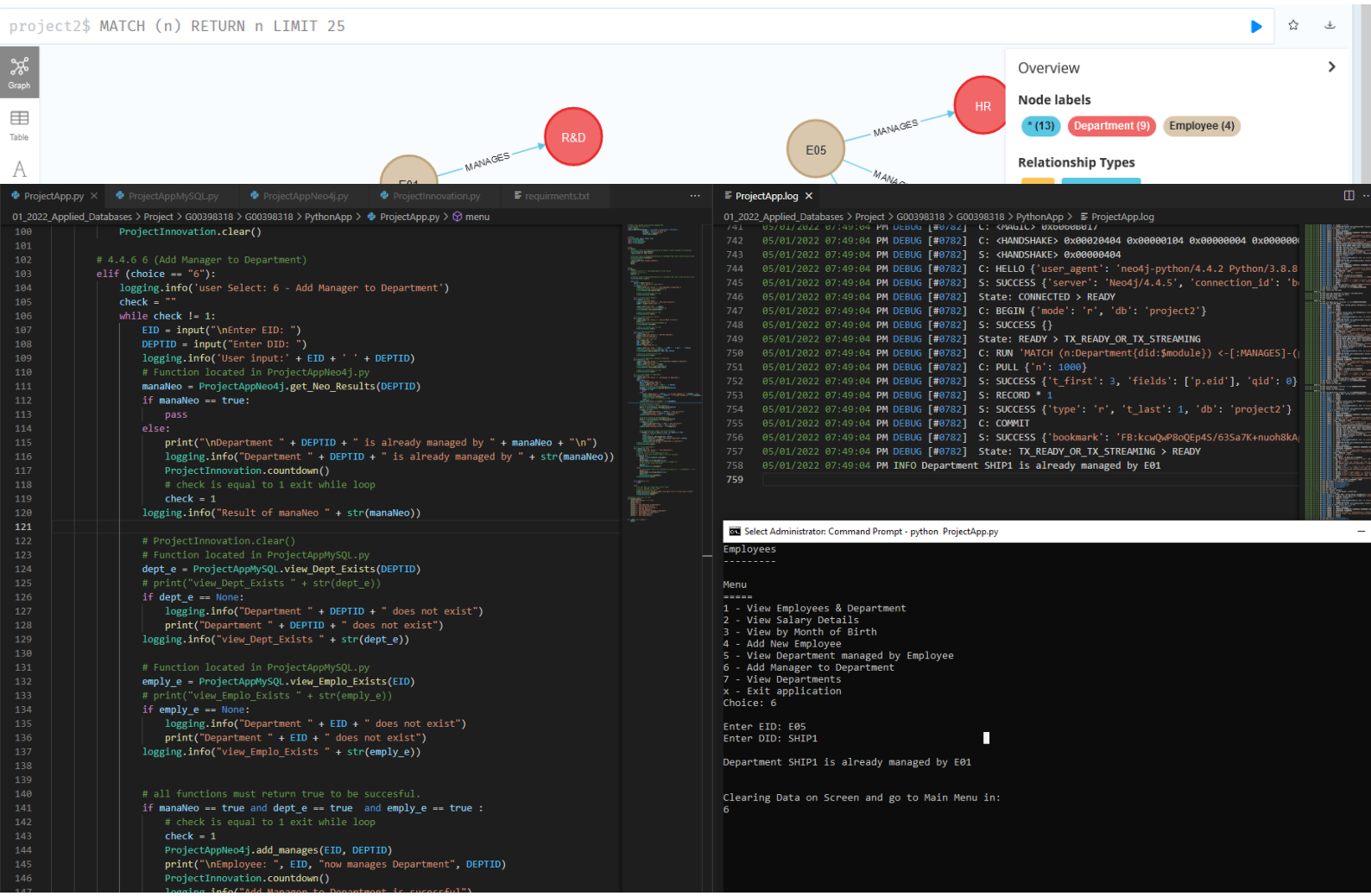
##

Choice 6

Adding


ProjectApp.py
ProjectAppMySQL.py
ProjectAppNeo4j.py
ProjectInnovation.py
requirments.txt
01_2022_Applied_Databases > Project > G00398318 > G00398318 > PythonApp > ProjectApp.py > menu
ProjectInnovation.clear()
# 4.4.6 6 (Add Manager to Department)
elif (choice == "6"):
logging.info('user Select: 6 - Add Manager to Department')
check = ""
while check != 1:
EID = input("\nEnter EID: ")
DEPTID = input("Enter DID: ")
logging.info('User input:' + EID + ' ' + DEPTID)
# Function located in ProjectAppNeo4j.py
manaNeo = ProjectAppNeo4j.get_Neo_Results(DEPTID)
if manaNeo == true:
pass
else:
print("\nDepartment " + DEPTID + " is already managed by " + manaNeo + "\n")
logging.info("Department " + DEPTID + " is already managed by " + str(manaNeo))
ProjectInnovation.countdown()
# check is equal to 1 exit while loop
check = 1
logging.info("Result of manaNeo " + str(manaNeo))
# ProjectInnovation.clear()
# Function located in ProjectAppMySQL.py
dept_e = ProjectAppMySQL.view_Dept_Exists(DEPTID)
# print("view_Dept_Exists " + str(dept_e))
if dept_e == None:
logging.info("Department " + DEPTID + " does not exist")
print("Department " + DEPTID + " does not exist")
logging.info("view_Dept_Exists " + str(dept_e))
# Function located in ProjectAppMySQL.py
emply_e = ProjectAppMySQL.view_Emplo_Exists(EID)
# print("view_Emplo_Exists " + str(emply_e))
if emply_e == None:
logging.info("Department " + EID + " does not exist")
print("Department " + EID + " does not exist")
logging.info("view_Emplo_Exists " + str(emply_e))
# all functions must return true to be succesful.
if manaNeo == true and dept_e == true and emply_e == true :
# check is equal to 1 exit while loop
check = 1
ProjectAppNeo4j.add_manages(EID, DEPTID)
print("\nEmployee: ", EID, "now manages Department", DEPTID)
ProjectInnovation.countdown()
logging.info("Add Manager to Department is sucessful")
# Return to display menu for user
ProjectApp.log
01_2022_Applied_Databases > Project > G00398318 > G00398318 > PythonApp > ProjectApp.log
581 05/01/2022 07:44:09 PM INFO Application ProjectInnovation.py
582 05/01/2022 07:44:09 PM INFO Application ProjectAppNeo4j.py
583 05/01/2022 07:44:09 PM INFO Application ProjectApp.py
584 05/01/2022 07:44:09 PM INFO All apps found
585 05/01/2022 07:44:18 PM INFO Main Program complete
586 05/01/2022 07:44:19 PM INFO user Select: 6 - Add Manager to Department
587 05/01/2022 07:44:23 PM INFO User input:E05 SW
588 05/01/2022 07:44:23 PM DEBUG [#0000] C: <NEO4J POOL> routing address IPv4Address(('localhost'
589 05/01/2022 07:44:23 PM DEBUG Attempting to update routing table from IPv4Address(('localhost',
590 05/01/2022 07:44:23 PM DEBUG [#0000] C: <RESOLVE> localhost:7687
591 05/01/2022 07:44:23 PM DEBUG [#0000] C: <OPEN> [::1]:7687
592 05/01/2022 07:44:25 PM DEBUG [#0000] C: <ERROR> ConnectionRefusedError 10061 'No connection c
593 05/01/2022 07:44:25 PM DEBUG [#0000] C: <CLOSE> [::1]:7687
594 05/01/2022 07:44:25 PM DEBUG [#0000] C: <CONNECTION FAILED> ServiceUnavailable: Failed to est
595 05/01/2022 07:44:25 PM DEBUG Failed to fetch routing info [::1]:7687
596 05/01/2022 07:44:25 PM DEBUG [#0000] C: <OPEN> 127.0.0.1:7687
597 05/01/2022 07:44:25 PM DEBUG [#0714] C: <MAGIC> 0x6060B017
598 05/01/2022 07:44:25 PM DEBUG [#0714] C: <HANDSHAKE> 0x00020404 0x00000104 0x00000004 0x000000
599 05/01/2022 07:44:25 PM DEBUG [#0714] S: <HANDSHAKE> 0x00000404
600 05/01/2022 07:44:25 PM DEBUG [#0714] C: HELLO {'user_agent': 'neo4j-python/4.4.2 Python/3.8.8
601 05/01/2022 07:44:25 PM DEBUG [#0714] S: SUCCESS {'server': 'Neo4j/4.4.5', 'connection_id': 'b
Administrator: Command Prompt - python ProjectApp.py
Employees

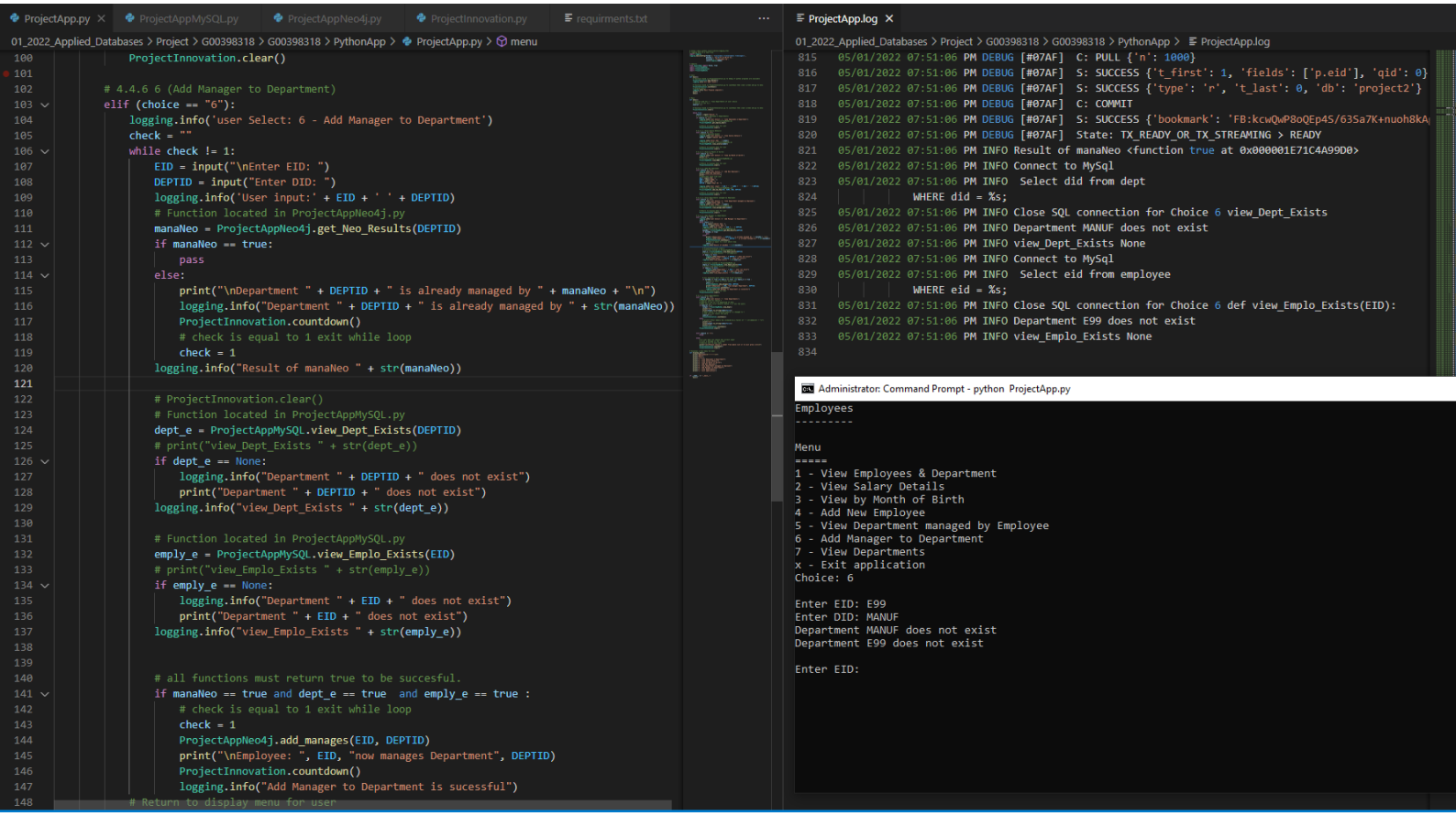
Menu
=====
1 - View Employees & Department
2 - View Salary Details
3 - View by Month of Birth
4 - Add New Employee
5 - View Department managed by Employee
6 - Add Manager to Department
7 - View Departments
x - Exit application
Choice: 6
Enter EID: E05
Enter DID: SW
629 05/01/2022 07:44:27 PM DEBUG [#0716] C: PULL {'n': 1000}


project2$ MATCH (n) RETURN n LIMIT 25
Graph
Table
R&D
MANAGES
E05
HR
Overview
Node labels
* (13)
Department (9)
Employee (4)
Relationship Types
ProjectApp.py
ProjectAppMySQL.py
ProjectAppNeo4j.py
ProjectInnovation.py
requirments.txt
01_2022_Applied_Databases > Project > G00398318 > G00398318 > PythonApp > ProjectApp.py > menu
ProjectInnovation.clear()
# 4.4.6 6 (Add Manager to Department)
elif (choice == "6"):
logging.info('user Select: 6 - Add Manager to Department')
check = ""
while check != 1:
EID = input("\nEnter EID: ")
DEPTID = input("Enter DID: ")
logging.info('User input:' + EID + ' ' + DEPTID)
# Function located in ProjectAppNeo4j.py
manaNeo = ProjectAppNeo4j.get_Neo_Results(DEPTID)
if manaNeo == true:
pass
else:
print("\nDepartment " + DEPTID + " is already managed by " + manaNeo + "\n")
logging.info("Department " + DEPTID + " is already managed by " + str(manaNeo))
ProjectInnovation.countdown()
# check is equal to 1 exit while loop
check = 1
logging.info("Result of manaNeo " + str(manaNeo))
# ProjectInnovation.clear()
# Function located in ProjectAppMySQL.py
dept_e = ProjectAppMySQL.view_Dept_Exists(DEPTID)
# print("view_Dept_Exists " + str(dept_e))
if dept_e == None:
logging.info("Department " + DEPTID + " does not exist")
print("Department " + DEPTID + " does not exist")
logging.info("view_Dept_Exists " + str(dept_e))
# Function located in ProjectAppMySQL.py
emply_e = ProjectAppMySQL.view_Emplo_Exists(EID)
# print("view_Emplo_Exists " + str(emply_e))
if emply_e == None:
logging.info("Department " + EID + " does not exist")
print("Department " + EID + " does not exist")
logging.info("view_Emplo_Exists " + str(emply_e))
# all functions must return true to be succesful.
if manaNeo == true and dept_e == true and emply_e == true :
# check is equal to 1 exit while loop
check = 1
ProjectAppNeo4j.add_manages(EID, DEPTID)
print("\nEmployee: ", EID, "now manages Department", DEPTID)
ProjectInnovation.countdown()
ProjectApp.log
01_2022_Applied_Databases > Project > G00398318 > G00398318 > PythonApp > ProjectApp.log
742 05/01/2022 07:49:04 PM DEBUG [#0782] C: <HANDSHAKE> 0x00020404 0x00000104 0x00000004 0x000000
743 05/01/2022 07:49:04 PM DEBUG [#0782] S: <HANDSHAKE> 0x00000404
744 05/01/2022 07:49:04 PM DEBUG [#0782] C: HELLO {'user_agent': 'neo4j-python/4.4.2 Python/3.8.8
745 05/01/2022 07:49:04 PM DEBUG [#0782] S: SUCCESS {'server': 'Neo4j/4.4.5', 'connection_id': 'b
746 05/01/2022 07:49:04 PM DEBUG [#0782] State: CONNECTED > READY
747 05/01/2022 07:49:04 PM DEBUG [#0782] C: BEGIN {'mode': 'r', 'db': 'project2'}
748 05/01/2022 07:49:04 PM DEBUG [#0782] S: SUCCESS {}
749 05/01/2022 07:49:04 PM DEBUG [#0782] State: READY > TX_READY_OR_TX_STREAMING
750 05/01/2022 07:49:04 PM DEBUG [#0782] C: RUN 'MATCH (n:Department{did:$module}) <-[:MANAGES]-(
751 05/01/2022 07:49:04 PM DEBUG [#0782] C: PULL {'n': 1000}
752 05/01/2022 07:49:04 PM DEBUG [#0782] S: SUCCESS {'t_first': 3, 'fields': ['p.eid'], 'qid': 0}
753 05/01/2022 07:49:04 PM DEBUG [#0782] S: RECORD * 1
754 05/01/2022 07:49:04 PM DEBUG [#0782] S: SUCCESS {'type': 'r', 't_last': 1, 'db': 'project2'}
755 05/01/2022 07:49:04 PM DEBUG [#0782] C: COMMIT
756 05/01/2022 07:49:04 PM DEBUG [#0782] S: SUCCESS {'bookmark': 'FB:kcwQwP8oQEp4S/63Sa7K+nuoh8kA
757 05/01/2022 07:49:04 PM DEBUG [#0782] State: TX_READY_OR_TX_STREAMING > READY
758 05/01/2022 07:49:04 PM INFO Department SHIP1 is already managed by E01
759
Select Administrator: Command Prompt - python ProjectApp.py
Employees

Menu
=====
1 - View Employees & Department
2 - View Salary Details
3 - View by Month of Birth
4 - Add New Employee
5 - View Department managed by Employee
6 - Add Manager to Department
7 - View Departments
x - Exit application
Choice: 6
Enter EID: E05
Enter DID: SHIP1
Department SHIP1 is already managed by E01
Clearing Data on Screen and go to Main Menu in:
6

Department already managed by

Employee and department does not exist



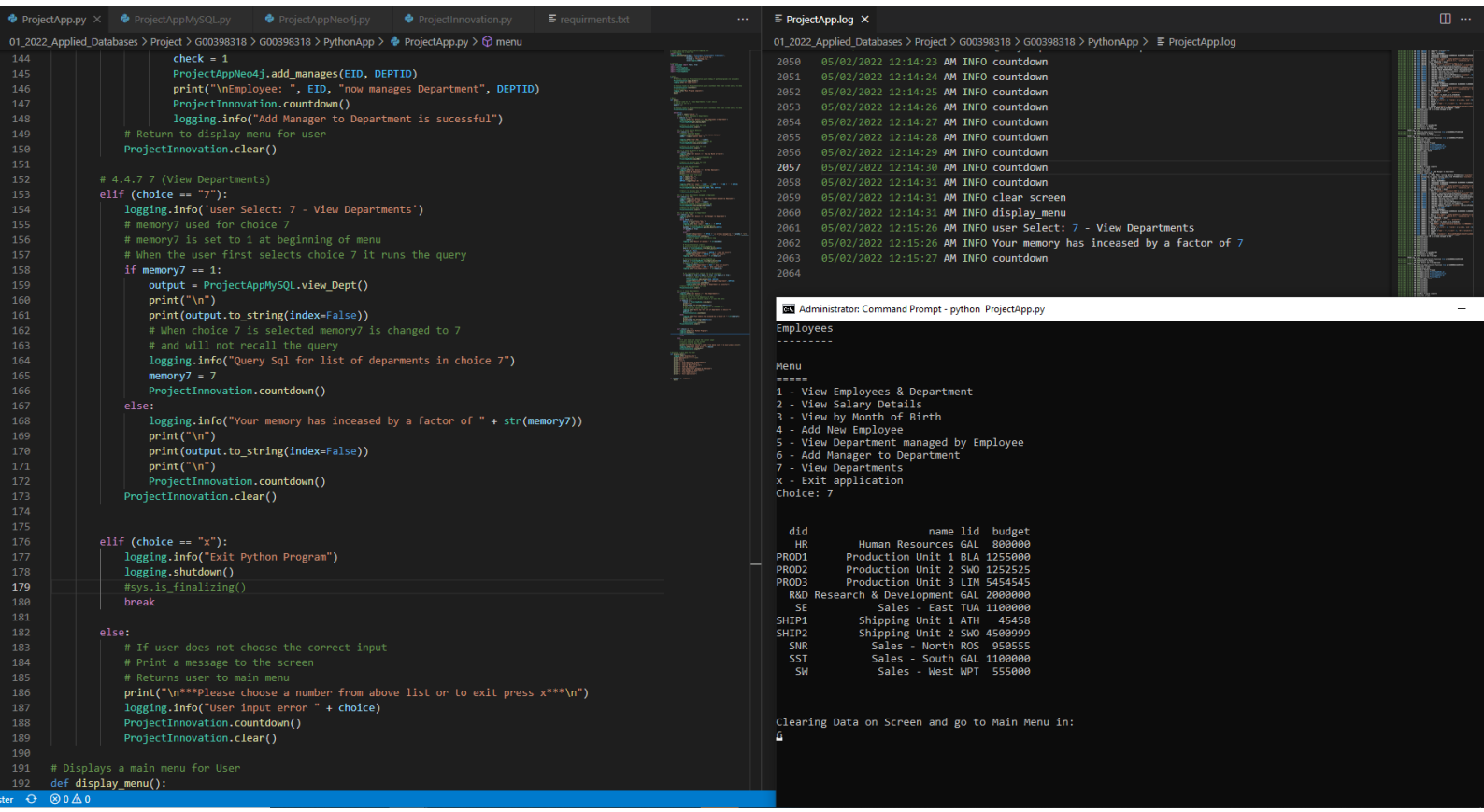
```
ProjectInnovation.clear()

# 4.4.6 6 (Add Manager to Department)
elif (choice == "6"):
    logging.info('user Select: 6 - Add Manager to Department')
    check = ""
    while check != 1:
        EID = input("\nEnter EID: ")
        DEPTID = input("Enter DID: ")
        logging.info('User input:' + EID + ' ' + DEPTID)
        # Function located in ProjectAppNeo4j.py
        manaNeo = ProjectAppNeo4j.get_Neo_Results(DEPTID)
        if manaNeo == true:
            pass
        else:
            print("\nDepartment " + DEPTID + " is already managed by " + manaNeo + "\n")
            logging.info("Department " + DEPTID + " is already managed by " + str(manaNeo))
            ProjectInnovation.countdown()
            # check is equal to 1 exit while loop
            check = 1
        logging.info("Result of manaNeo " + str(manaNeo))

        # ProjectInnovation.clear()
        # Function located in ProjectAppMySQL.py
        dept_e = ProjectAppMySQL.view_Dept_Exists(DEPTID)
        # print("view_Dept_Exists " + str(dept_e))
        if dept_e == None:
            logging.info("Department " + DEPTID + " does not exist")
            print("Department " + DEPTID + " does not exist")
        logging.info("view_Dept_Exists " + str(dept_e))

        # Function located in ProjectAppMySQL.py
        emply_e = ProjectAppMySQL.view_Emplo_Exists(EID)
        # print("view_Emplo_Exists " + str(emply_e))
        if emply_e == None:
            logging.info("Department " + EID + " does not exist")
            print("Department " + EID + " does not exist")
        logging.info("view_Emplo_Exists " + str(emply_e))


        # all functions must return true to be succesful.
        if manaNeo == true and dept_e == true  and emply_e == true :
            # check is equal to 1 exit while loop
            check = 1
            ProjectAppNeo4j.add_manages(EID, DEPTID)
            print("\nEmployee: ", EID, "now manages Department", DEPTID)
            ProjectInnovation.countdown()
            logging.info("Add Manager to Department is sucessful")
    # Return to display menu for user
```

```
05/01/2022 07:51:06 PM DEBUG [#07AF]  C: PULL {'n': 1000}
05/01/2022 07:51:06 PM DEBUG [#07AF]  S: SUCCESS {'t_first': 1, 'fields': ['p.eid'], 'qid': 0}
05/01/2022 07:51:06 PM DEBUG [#07AF]  S: SUCCESS {'type': 'r', 't_last': 0, 'db': 'project2'}
05/01/2022 07:51:06 PM DEBUG [#07AF]  C: COMMIT
05/01/2022 07:51:06 PM DEBUG [#07AF]  S: SUCCESS {'bookmark': 'FB:kcwQwP8oQEp4S/63Sa7K+nuoh8kA
05/01/2022 07:51:06 PM DEBUG [#07AF]  State: TX_READY_OR_TX_STREAMING > READY
05/01/2022 07:51:06 PM INFO Result of manaNeo <function true at 0x000001E71C4A99D0>
05/01/2022 07:51:06 PM INFO Connect to MySql
05/01/2022 07:51:06 PM INFO  Select did from dept
                WHERE did = %s;
05/01/2022 07:51:06 PM INFO Close SQL connection for Choice 6 view_Dept_Exists
05/01/2022 07:51:06 PM INFO Department MANUF does not exist
05/01/2022 07:51:06 PM INFO view_Dept_Exists None
05/01/2022 07:51:06 PM INFO Connect to MySql
05/01/2022 07:51:06 PM INFO  Select eid from employee
                WHERE eid = %s;
05/01/2022 07:51:06 PM INFO Close SQL connection for Choice 6 def view_Emplo_Exists(EID):
05/01/2022 07:51:06 PM INFO Department E99 does not exist
05/01/2022 07:51:06 PM INFO view_Emplo_Exists None
```

```
Administrator: Command Prompt - python  ProjectApp.py
Employees
---------

Menu
=====
1 - View Employees & Department
2 - View Salary Details
3 - View by Month of Birth
4 - Add New Employee
5 - View Department managed by Employee
6 - Add Manager to Department
7 - View Departments
x - Exit application
Choice: 6

Enter EID: E99
Enter DID: MANUF
Department MANUF does not exist
Department E99 does not exist

Enter EID:
```

##

Choice 7
Table called from memory



##

Does not recreate on every time program is run
Possible error if exit program after choice 6
Error on exit for logging for a known issue with Python finalisation
"NameError: name 'open' is not defined" when logging really late during the Python finalization

https://bugs.python.org/issue26789
https://pythondev.readthedocs.io/finalization.html

```
Exception ignored in: <function Driver.__del__ at 0x000002528EC634C0>
Traceback (most recent call last):
  File "C:\ProgramData\Anaconda3\lib\site-packages\neo4j\__init__.py", line 281, in __del__
  File "C:\ProgramData\Anaconda3\lib\site-packages\neo4j\__init__.py", line 311, in close
  File "C:\ProgramData\Anaconda3\lib\site-packages\neo4j\io\__init__.py", line 806, in close
  File "C:\ProgramData\Anaconda3\lib\site-packages\neo4j\io\__init__.py", line 795, in remove
  File "C:\ProgramData\Anaconda3\lib\site-packages\neo4j\io\_bolt4.py", line 333, in close
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1430, in debug
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1585, in _log
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1595, in handle
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1657, in callHandlers
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 950, in handle
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1182, in emit
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1172, in _open
NameError: name 'open' is not defined
Exception ignored in: <function Bolt.__del__ at 0x000002528EC48EE0>
Traceback (most recent call last):
  File "C:\ProgramData\Anaconda3\lib\site-packages\neo4j\io\__init__.py", line 390, in __del__
  File "C:\ProgramData\Anaconda3\lib\site-packages\neo4j\io\_bolt4.py", line 333, in close
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1430, in debug
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1585, in _log
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1595, in handle
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1657, in callHandlers
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 950, in handle
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1182, in emit
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1172, in _open
NameError: name 'open' is not defined
Exception ignored in: <function Bolt.__del__ at 0x000002528EC48EE0>
Traceback (most recent call last):
  File "C:\ProgramData\Anaconda3\lib\site-packages\neo4j\io\__init__.py", line 390, in __del__
  File "C:\ProgramData\Anaconda3\lib\site-packages\neo4j\io\_bolt4.py", line 333, in close
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1430, in debug
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1585, in _log
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1595, in handle
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1657, in callHandlers
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 950, in handle
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1182, in emit
  File "C:\ProgramData\Anaconda3\lib\logging\__init__.py", line 1172, in _open
NameError: name 'open' is not defined
```