Abir Razzak, INFO365 – Database Administration I – PA8

(View in “Web Layout” for best reading experience)

Part 1

Code (Python)

import mysql.connector

mydb = mysql.connector.connect(

  host="localhost",

  user="dba",

  password="",

  database="CollegeScorecard"

)

print("Connected to DB successfully.\n\n")

input\_name = input("Please enter a college name to get information:")

mycursor = mydb.cursor()

sql = "SELECT s.OPEID, s.INSTNM, s.CITY, s.STABBR FROM school s WHERE INSTNM LIKE %s LIMIT 10"

instnm\_name = ("%"+input\_name+"%", )  # Prevent SQL Injection

mycursor.execute(sql, instnm\_name)

myresult = mycursor.fetchall()

print("\nSchools Found:")

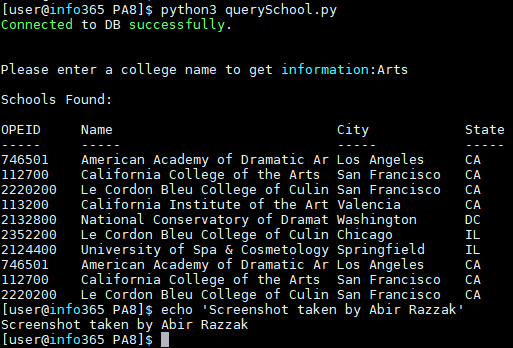
print('\n{:10}{:32}{:16}{:2}'.format('OPEID', 'Name', 'City', 'State'))

print('{:10}{:32}{:16}{:2}'.format('-----', '-----', '-----', '-----'))

for x in myresult:

    print('{:10}{:32.31}{:16.15}{:2}'.format(str(x[0]),x[1],x[2],x[3]))

Screenshot



Part 2

SQL Script

**CREATE** **OR** **REPLACE** **VIEW** SchoolData **AS**

**SELECT**

s.OPEID **as** 'Institution ID',

s.INSTNM **as** 'Institution Name',

s.ACCREDAGENCY **as** 'Crediting Agency',

s.INSTURL **as** 'Institution Website',

r.RELAFFIL\_VALUE **as** 'Religious Affiliation',

l.LOCALE\_VALUE **as** 'Institution Region',

c.TUITIONFEE\_IN **as** 'Tuition In-State',

c.TUITIONFEE\_OUT **as** 'Tuition Out-State',

a.ADM\_RATE **as** 'Administration Rate',

a.SAT\_AVG **as** 'SAT Average Score'

**FROM** school s

**LEFT** **JOIN** relaffil r

**ON** s.RELAFFIL = r.RELAFFIL\_ID

**LEFT** **JOIN** locale l

**ON** s.LOCALE = l.LOCALE\_ID

**LEFT** **JOIN** cost c

**ON** s.OPEID = c.OPEID

**LEFT** **JOIN** admissions a

**ON** s.OPEID = a.OPEID;

**SELECT** \* **FROM** SchoolData **WHERE** `Institution ID` = 337100;

Code (Python)

import mysql.connector

import json

mydb = mysql.connector.connect(

*host*="localhost",

*user*="dba",

*password*="",

*database*="CollegeScorecard"

)

print("Connected to DB successfully.\n\n")

input\_name = input("Please enter a college name to get information: ")

cursor = mydb.cursor()

query1 = "SELECT s.OPEID, s.INSTNM, s.CITY, s.STABBR FROM school s WHERE INSTNM LIKE %s LIMIT 10"

param1 = ("%"+input\_name+"%", ) # Prevent SQL Injection

cursor.execute(query1, param1)

result1 = cursor.fetchall()

columns1 = cursor.column\_names

print("\nSchools Found:")

print('\n{:10}{:32}{:16}{:2}'.format(columns1[0], columns1[1], columns1[2], columns1[3]))

print('{:10}{:32}{:16}{:2}'.format('-----', '-----', '-----', '-----'))

for x in result1:

print('{:10}{:32.31}{:16.15}{:2}'.format(str(x[0]),x[1],x[2],x[3]))

input\_opeid = input('\n\nPlease provide an OPEID for the desired school: ')

query2 = "SELECT \* FROM SchoolData WHERE `Institution ID` = %s LIMIT 1"

param2 = (input\_opeid, ) # Prevent SQL Injection

cursor.execute(query2, param2)

result2 = cursor.fetchall()

columns2 = cursor.column\_names

# Code from: https://stackoverflow.com/questions/43796423/python-converting-mysql-query-result-to-json

json\_data = []

for r in result2:

json\_data.append(dict(zip(columns2,r)))

print(json.dumps(json\_data))

Screenshot

Graphical user interface, application

Description automatically generated

Part 3

Code – User Authentication (Python)

def authenticate\_user():  
 input\_user = input('UserID: ')  
 input\_pass = input('Password:')  
  
 query = "SELECT UserPassPhrase, UserName FROM APIUser a WHERE UserID = %s;"  
 param = (input\_user, ) # Prevent SQL Injection  
  
 cursor.execute(query, param)  
  
 result = cursor.fetchone()  
  
 if result is not None and result[0] == input\_pass:  
 print('Password Correct')  
 print('--WELCOME BACK {}--'.format(result[1]))  
 else:  
 print('Password Incorrect')  
 exit(-1)

Full Code (Python)

import mysql.connector  
import json  
  
  
def get\_basic\_college\_data():  
 input\_name = input("Please enter a college name to get information: ")  
  
 query = "SELECT s.OPEID, s.INSTNM, s.CITY, s.STABBR FROM school s WHERE INSTNM LIKE %s LIMIT 10"  
 param = ("%" + input\_name + "%",) # Prevent SQL Injection  
  
 cursor.execute(query, param)  
  
 result1 = cursor.fetchall()  
 columns1 = cursor.column\_names  
  
 print("\nSchools Found:")  
 print('\n{:10}{:32}{:16}{:2}'.format(columns1[0], columns1[1], columns1[2], columns1[3]))  
 print('{:10}{:32}{:16}{:2}'.format('-----', '-----', '-----', '-----'))  
  
 for x in result1:  
 print('{:10}{:32.31}{:16.15}{:2}'.format(str(x[0]), x[1], x[2], x[3]))  
  
  
def get\_advanced\_college\_data():  
 input\_opeid = input('\n\nPlease provide an OPEID for the desired school: ')  
  
 query2 = "SELECT \* FROM SchoolData WHERE `Institution ID` = %s"  
 param2 = (input\_opeid,) # Prevent SQL Injection  
  
 cursor.execute(query2, param2)  
  
 result = cursor.fetchone()  
 columns = cursor.column\_names  
  
 input\_display\_option = input('How would you like the data to be displayed? '  
 + '1.Text, 2.JSON, 3.Python Data Structure: ')  
  
 if input\_display\_option == '1':  
 for i in range(len(result)):  
 print('{}: {}'.format(columns[i], result[i]))  
  
 elif input\_display\_option == '2' or input\_display\_option == '3':  
 data = [(dict(zip(columns, result)))] # Combine the column and value together (Php does this automatically)  
 if input\_display\_option == '2':  
 json\_data = json.dumps(data, indent=2)  
 print(json\_data)  
 if input\_display\_option == '3':  
 print(data)  
  
 else:  
 print('Invalid Option Given.')  
  
  
def authenticate\_user():  
 input\_user = input('UserID: ')  
 input\_pass = input('Password:')  
  
 query = "SELECT UserPassPhrase, UserName FROM APIUser a WHERE UserID = %s;"  
 param = (input\_user, ) # Prevent SQL Injection  
  
 cursor.execute(query, param)  
  
 result = cursor.fetchone()  
  
 if result is not None and result[0] == input\_pass:  
 print('Password Correct')  
 print('--WELCOME BACK {}--'.format(result[1]))  
 else:  
 print('Password Incorrect')  
 exit(-1)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 mydb = mysql.connector.connect(  
 host="localhost",  
 user="dba",  
 password="",  
 database="CollegeScorecard"  
 )  
 print("Connected to DB successfully.\n\n")  
 cursor = mydb.cursor()  
 authenticate\_user()  
 get\_basic\_college\_data()  
 get\_advanced\_college\_data()

Screenshots

Text

Description automatically generated

Text

Description automatically generated

Part 4

So looking at the Php code that was given, I could tell that all the inputs were open to SQL injection, because the code is literally inserting the input text into the “WHERE = …” clause in some of the queries. So in theory a user could type in “Drexel; DROP ALL TABLES” and erase all the data in the database. To combat that I used parameters via Python’s SQL Connector library to prevent this from happening. In my queries you can see that I use a “%s” flag in my queries, that are then replaced by a formatted `param` variable in my code. I can format the param variable and perform various checks before inserting that data into the query to ensure that SQL injection does not occur. This type of prevention is done from the application level before the query is even sent to the database, to mitigate the risks before a call to the database can be made.