**PA - 342.3.1 - Practice Assignment - Python Database API**

**Objective:**

In this assignment, you will practice Python Database API.

**Requirements:**

1. **Creating Database:**

Create a Python Program to create a Database, named RegistrationDB.

1. **Creating Method:**

Create a file named **usertable\_manipulation.py**. Create a Python program with the following methods:

➢ generate\_user\_table()

➢ get\_all\_users()

➢ get\_user\_by\_name(name)

➢ validate\_user(email, password)

➢ update\_user(email, name, password)

1. **generate\_user\_table() Method:**

➢ Establish a connection to the RegistrationDB.

➢ Create a user table with the following fields/columns:

▶ email varchar(100)

▶ name varchar(50)

▶ password varchar(30)

➢ Insert the following values into the table:

▶ 'ywbaek@perscholas.org', 'young', 'letsgomets'

▶ 'mcordon@perscholas.org', 'marcial', 'perscholas'

▶ 'mhaseeb@perscholas.org', 'haseeb', 'platform'

1. **get\_all\_users() Method:**

➢ Establish a connection to the RegistrationDB.

➢ Print out the email, name, and password of all users.

1. **get\_user\_by\_name(name) Method:**

➢ Establish a connection to the RegistrationDB.

➢ Print out the email and password of the user with the given name.

1. **validate\_user(email, password) Method:**

➢ Establish a connection to the RegistrationDB.

➢ Return True if a user exists with the given email and password. Return False otherwise.

1. **update\_user(email, name, password) method:**

➢ Establish a connection to the test\_db.

➢ For the user with the given email, set the values of name and password to the given name and password.

➢ Return True if successful, False otherwise.

➢ Test the functions by running them in the program.

➢ Check the database manually to see if the changes are applied correctly.