**Programming Project: Password Manager**

**Project Description:**

Write a basic password management that generates random passwords when given a username and URL. PBKDF2 will be used to derive the cypher key from the master password. The master password is EUID, and the same master password will be required for all operations. Passwords are encrypted and just one master password is required to access the "vault." The vault includes all passwords that have been saved. To generate, save, and query for new passwords, use a terminal interface. The encrypted passwords for a specific URL will be kept in. dat file format.

**Tools:** Python and PostgreSQL

**Libraries :**

* Argparse
* os
* Psycopg2
* sql\_statements
* db\_connect
* Password\_generator
* master\_password
* hashlib

**General Steps:**

* Setup environment (infrastructure)
  + PostgreSQL Database
  + Interface with python, psycopg2
  + PDKB2
  + Cryptodome
* Create database named csceproject
* Create vault DB with URL, username, and password
* Generate a random, 20+ character password with uppercase, lowercase, integers, and special characters.
* Manually store username and password inside the Database
* Create python functions for INSERT, UPDATE, AND DELETE
* Create terminal interface using argparse library. Options to add (-a or --add), query (-l or --list), (-d or delete) , (-q or query) and (-uurl or --updateURL) ,(-ap or add\_password),(-uuname or update\_username),(-upasswd or update\_password)
* Master Password: Create master password function which will be hashed (with salt) and combined with an encryption key. This will provide the encryption key to the locked database which will unlock the database.
  + Connect PostgreSQL table to argparse, terminal will generate a random password.
* Hash master password with salt PBKDF2.
* Master password must be supplied when running the python script.
* To add a new record and update the custom password manually
* To store the encrypted password in a .dat file