Bonillo, Loren
C204

Finals Lab Task 5. CLI using Mysql and Python

1. Make sure you have installed the following pre-requisites before proceeding:
   a. Mysql-connector
   b. Mysql-connector-python
   c. Xampp is running along with Apache and Mysql in the background
2. Create the following database in Mysql;
   a. Database name: **moviesDB** with the ff: fields:

      movie_id int(10) Primary Key
      title varchar(50) NOT NULL
      main_actor varchar(50) NOT NULL
      director varchar(50) NOT NULL
      genre varchar(25) NOT NULL
      gross_sales float
      ratings (G, PG, R13, R16,X) varchar(5)

   b. Insert at_least 5 records
   c. Create a user named **test_user** and assign a **password** and give it an admin access by checking necessary SQL functions

3. Guided by the Demo code attached in this task. test_DemoDB.py
4. Kindly continue working on the code that will allow the user to navigate through the Database and perform simple CRUD operations. Follow the following **CLI Menu Options:**

```
----- MOVIE DATABASE CLI -----
1. Add Movie
2. View Movies
3. Update Movies
4. Delete a Movie
5. Search a Movie
6. Display Total Records
7. Exit
Select an option (1-6):
```

SOURCE CODE:

```python
import mysql.connector
from mysql.connector import Error

6 usages
def create_connection():
    try:
        connection = mysql.connector.connect(
            host="localhost",
            user="test_user",
            password="12345",
            database="moviesDB"
        )
        return connection

    except Error as e:
        print("Error connecting to MySQL:", e)
        return None

1 usage
def add_movie():
    conn = create_connection()
    cursor = conn.cursor()

    print("\n--- ADD NEW MOVIE ---")
    title = input("Title: ")
    main_actor = input("Main actor: ")
    director = input("Director: ")
    genre = input("Genre: ")
    gross_sales = float(input("Gross sales: "))
    ratings = input("Rating (G, PG, R13, R16, X): ")

    sql = """
        INSERT INTO movies (title, main_actor, director, genre, gross_sales, ratings)
        VALUES (%s, %s, %s, %s, %s, %s)
    """

    cursor.execute(sql,  params: (title, main_actor, director, genre, gross_sales, ratings))
    conn.commit()
```

```python
    conn.commit()

    print("Movie updated successfully!\n")
    conn.close()

1 usage
def delete_movie():
    conn = create_connection()
    cursor = conn.cursor()

    movie_id = int(input("\nEnter Movie ID to delete: "))

    cursor.execute( operation: "DELETE FROM movies WHERE movie_id=%s", params: (movie_id,))
    conn.commit()

    print("Movie deleted successfully!\n")
    conn.close()

1 usage
def search_movie():
    conn = create_connection()
    cursor = conn.cursor()

    print("\n--- SEARCH MOVIE ---")
    print("1. Search by title")
    print("2. Search by Movie ID")
    choice = input("Choose option: ")

    if choice == "1":
        keyword = input("Enter title keyword: ")
        cursor.execute( operation: "SELECT * FROM movies WHERE title LIKE %s", params: ("%" + keyword + "%",))

    elif choice == "2":
        movie_id = int(input("Enter Movie ID: "))
        cursor.execute( operation: "SELECT * FROM movies WHERE movie_id=%s", params: (movie_id,))

    results = cursor.fetchall()

    print("\n--- SEARCH RESULTS ---")
```

```
        print("Movie added successfully!\n")
        conn.close()

    1 usage
def view_movies():
        conn = create_connection()
        cursor = conn.cursor()

        cursor.execute("SELECT * FROM movies")
        rows = cursor.fetchall()

        print("\n--- MOVIE LIST ---")
        for row in rows:
            print(row)
        print()

        conn.close()

    1 usage
def update_movie():
        conn = create_connection()
        cursor = conn.cursor()

        movie_id = int(input("\nEnter Movie ID to update: "))

        print("\nEnter new details:")
        title = input("New title: ")
        main_actor = input("New main actor: ")
        director = input("New director: ")
        genre = input("New genre: ")
        gross_sales = float(input("New gross sales: "))
        ratings = input("New rating: ")

        sql = """
            UPDATE movies
            SET title=%s, main_actor=%s, director=%s, genre=%s, gross_sales=%s, ratings=%s
            WHERE movie_id=%s
        """

        cursor.execute(sql, params (title, main_actor, director, genre, gross_sales, ratings, movie_id))
```