import mysql.connector

# ---- Database Connection ----

def get\_connection():

    try:

        conn = mysql.connector.connect(

        host="localhost",

        user="root",

        password="Mathavan@003",

        database="gaming\_zone"

)

        return conn

    except mysql.connector.Error as err:

        print("Error connecting to MySQL:", err)

        return None

# ---- Game Operations ----

def add\_game():

    name = input("Enter game name: ")

    game\_type = input("Enter game type (Racing/Sports/etc.): ")

    charge = float(input("Enter charge per hour: "))

    conn = get\_connection()

    if conn:

        cursor = conn.cursor()

        cursor.execute("""

            INSERT INTO games (name, type, charge\_per\_hour)

            VALUES (%s, %s, %s)

        """, (name, game\_type, charge))

        conn.commit()

        conn.close()

        print("Game added successfully!")

# ---- Member Operations ----

def register\_member():

    name = input("Enter member name: ")

    membership = input("Membership type (Yearly/Monthly/Daily): ")

    conn = get\_connection()

    if conn:

        cursor = conn.cursor()

        cursor.execute("SELECT initial\_hours FROM memberships WHERE type = %s", (membership,))

        result = cursor.fetchone()

        if result:

            total\_hours = result[0]

            cursor.execute("""

                INSERT INTO members (name, membership\_type, total\_hours, hours\_spent, hours\_left)

                VALUES (%s, %s, %s, 0, %s)

            """, (name, membership, total\_hours, total\_hours))

            conn.commit()

            print("Member registered!")

        else:

            print("Invalid membership type")

        conn.close()

# ---- Gameplay Logging ----

def log\_gameplay():

    member\_name = input("Enter member name: ")

    game\_name = input("Enter game name: ")

    hours = int(input("Enter hours to play: "))

    conn = get\_connection()

    if conn:

        cursor = conn.cursor()

        cursor.execute("SELECT id, hours\_left FROM members WHERE name = %s", (member\_name,))

        member = cursor.fetchone()

        cursor.execute("SELECT id, charge\_per\_hour FROM games WHERE name = %s", (game\_name,))

        game = cursor.fetchone()

        if member and game:

            if member[1] >= hours:

                cursor.execute("""

                    INSERT INTO gameplays (member\_id, game\_id, hours\_played)

                    VALUES (%s, %s, %s)

                """, (member[0], game[0], hours))

                cursor.execute("""

                    UPDATE members

                    SET hours\_spent = hours\_spent + %s,

                        hours\_left = hours\_left - %s

                    WHERE id = %s

                """, (hours, hours, member[0]))

                conn.commit()

                print("Gameplay logged!")

            else:

                print("Not enough hours left!")

        else:

            print("Invalid member or game name")

        conn.close()

# ---- Report: Top 3 Most Played Games ----

def show\_top\_3\_games():

    conn = get\_connection()

    if conn:

        cursor = conn.cursor()

        cursor.execute("""

            SELECT g.name, SUM(gp.hours\_played) as total\_played

            FROM gameplays gp

            JOIN games g ON g.id = gp.game\_id

            GROUP BY g.name

            ORDER BY total\_played DESC

            LIMIT 3

        """)

        print("Top 3 Games:")

        for row in cursor.fetchall():

            print(row)

        conn.close()

# ---- Main Menu ----

if \_\_name\_\_ == "\_\_main\_\_":

    while True:

        print("\n=== Gaming Zone Menu ===")

        print("1. Add Game")

        print("2. Register Member")

        print("3. Log Gameplay")

        print("4. Show Top 3 Games")

        print("5. Exit")

        choice = input("Choose an option: ")

        if choice == "1":

            add\_game()

        elif choice == "2":

            register\_member()

        elif choice == "3":

            log\_gameplay()

        elif choice == "4":

            show\_top\_3\_games()

        elif choice == "5":

            break

        else:

            print("Invalid choice.")