import sqlite3

import hashlib

# Connect to the SQLite database

conn = sqlite3.connect('movie\_booking.db')

cursor = conn.cursor()

# Create tables if not exists

cursor.execute('''CREATE TABLE IF NOT EXISTS Users (

user\_id INTEGER PRIMARY KEY,

username TEXT UNIQUE,

password TEXT,

email TEXT)''')

cursor.execute('''CREATE TABLE IF NOT EXISTS Movies (

movie\_id INTEGER PRIMARY KEY,

title TEXT,

genre TEXT,

release\_date DATE)''')

cursor.execute('''CREATE TABLE IF NOT EXISTS Bookings (

booking\_id INTEGER PRIMARY KEY,

user\_id INTEGER,

movie\_id INTEGER,

showtime TEXT,

seats TEXT,

total\_price REAL,

FOREIGN KEY (user\_id) REFERENCES Users(user\_id),

FOREIGN KEY (movie\_id) REFERENCES Movies(movie\_id))''')

# Sample data insertion

cursor.execute("INSERT OR IGNORE INTO Users (username, password, email) VALUES ('user123', ?, 'user123@example.com')", (hashlib.sha256(b'password123').hexdigest(),))

cursor.execute("INSERT OR IGNORE INTO Movies (title, genre, release\_date) VALUES ('Avengers: Endgame', 'Action, Adventure, Sci-Fi', '2019-04-26')")

cursor.execute("INSERT OR IGNORE INTO Movies (title, genre, release\_date) VALUES ('The Godfather', 'Crime, Drama', '1972-03-24')")

cursor.execute("INSERT OR IGNORE INTO Movies (title, genre, release\_date) VALUES ('The Shawshank Redemption', 'Drama', '1994-09-23')")

conn.commit()

# Function for user login

def login(username, password):

password\_hash = hashlib.sha256(password.encode()).hexdigest()

cursor.execute("SELECT \* FROM Users WHERE username = ? AND password = ?", (username, password\_hash))

user = cursor.fetchone()

return user

# Function to browse movies

def browse\_movies():

cursor.execute("SELECT \* FROM Movies")

movies = cursor.fetchall()

print("Movies Available for Booking:\n")

for movie in movies:

print(f"{movie[0]}. {movie[1]}")

print(f" Genre: {movie[2]}")

print(f" Release Date: {movie[3]}\n")

# Function to book tickets

def book\_tickets(user\_id, movie\_id, showtime, seats):

# Calculate total price (assuming $10 per ticket)

total\_price = len(seats) \* 10

cursor.execute("INSERT INTO Bookings (user\_id, movie\_id, showtime, seats, total\_price) VALUES (?, ?, ?, ?, ?)",

(user\_id, movie\_id, showtime, ','.join(seats), total\_price))

conn.commit()

print("Booking confirmed!\n")

# Sample login process

username = input("Username: ")

password = input("Password: ")

user = login(username, password)

if user:

print("Login successful!\n")

while True:

print("1. Browse Movies")

print("2. Logout")

choice = input("Please enter your choice: ")

if choice == '1':

browse\_movies()

movie\_id = input("Please enter the movie number to view showtimes or enter '0' to go back: ")

if movie\_id != '0':

# Assuming showtimes are fixed for simplicity

showtimes = ["12:00 PM", "3:00 PM", "6:00 PM"]

for i, showtime in enumerate(showtimes, start=1):

print(f"{i}. {showtime}")

showtime\_choice = input("Please enter the showtime number to continue booking or enter '0' to go back: ")

if showtime\_choice != '0':

seats = input("Please select your seats (e.g., 1A, 1B) separated by commas: ").split(',')

# Assuming movie\_id and user\_id are available from the login process

book\_tickets(user[0], int(movie\_id), showtimes[int(showtime\_choice) - 1], seats)

elif choice == '2':

print("Logged out successfully!")

break

else:

print("Invalid username or password.")