

Innovation:

```
MoviesDB
-----

MENU
====
1 - View Directors & Films
2 - View Actors by Month of Birth
3 - Add New Actor
4 - View Married Actors
5 - Add Actor Marriage
6 - View Studios
7 - Recommend a Film
x - Exit application
Choice: 7
How old are you? 12
Are your parent(s)/guardian(s) watching too? (y/n)y

Recommended Film: Shanghai Knights
Rated: 12A
```

I created a seventh option for the app that generates a film recommendation based on age. If appropriate it will also ask if a parent or guardian is present. Then, based on the user input it will select a film at random from a list of age appropriate films.

CertificateID	Certificate
1	U
2	PG
3	12
4	12A
5	15
6	18

The certificates are as above.

I decided that children 9 and below that have their parents or guardians present would be recommended films PG and lower, otherwise they would be recommended films rated U only. Children between 10 and 11 would be recommended PG films by default. Films rated 12A would only be recommended to children aged 12 with a parent or guardian present while 13 and 14 year olds would be recommended these by default. Children aged 15, 16 and 17 would be recommended films 15 and below. And finally anyone aged 18 and above would receive recommendations from all classifications.

I used this [reference](#) to base my decisions.

```

while True:
    age = input('How old are you? ')
    try:
        age = int(age)
        break
    except ValueError:
        continue
if age <= 12:
    while True:
        parents = input('Are your parent(s)/guardian(s) watching too? (y/n) ').lower()
        if parents in ['y', 'n']:
            break
if age >= 18:
    category = 6
if age <= 17:
    category = 5
if age <= 14:
    category = 4
if age == 12 and parents == 'y':
    category = 4
if age == 12 and parents == 'n':
    category = 3
if age <= 11:
    category = 2
if age <= 9 and parents == 'y':
    category = 2
if age <= 9 and parents == 'n':
    category = 1

```

The appropriate film rating is selected using a series of if statements..

```

connect()
sql_recommend = "SELECT f.FilmName as 'Film Name', c.Certificate " \
    "FROM film f INNER JOIN certificate c " \
    "ON f.FilmCertificateID = c.CertificateID " \
    "WHERE c.CertificateID <= %s;"
with conn:
    cursor = conn.cursor()
    cursor.execute(sql_recommend,(category))
    recommend_result = cursor.fetchall()

```

It then fetches all appropriate films of that age rating and below.

```

length = len(recommend_result)
selection = random.randint(0,length)
print(f"\nRecommended Film: {recommend_result[selection]['Film Name']}\n" \
    f"Rated: {recommend_result[selection]['Certificate']}\n")

```

Then using the random package, it generates a random number and selects the film at that index and displays the title and rating to the user.