PROJECT BACKGROUND

A movie-goer (me) is asking for a software application to make queries across multiple movie databases simpler. There are many databases on the internet that contain data about movies e.g., IMDB, Rotten Tomatoes, Amazon, Netflix etc. Typically, these sites contain ways to purchase the movie and or review movie ratings. They may also contain some data about rewards the movie has received such as Golden Globe or Academy Awards. These databases are organized on the premise that you have a particular movie title, genre, or even actor in mind and wish to find movies meeting these criteria. The Academy of Motion Pictures has a database containing data regarding every Oscar award won for every category since the awards inception. This database is query-able via their website.

For this project, the premise revolves strictly around Academy award winning movies and linking Academy award queries with corresponding entries. The motivation for this project is the client (me) wants to watch all movies that received best picture awards. I want the ability to query a database, receive a list of these movies and links to information about each movie such as reviews, rental/streaming options, actors. Furthermore, I want to be able to delete, add, and update existing movies to best fit my needs.

Is / Is-nots

Since this project is *modified* to fit the schedule for the semester and the skill set for each student. The following IS and IS-NOT are true:

Is

- A large coding project where changes are made to a Git repository hosted on GitHub
- A robust REST API in any language
- A REST API which calls other external APIs
- A REST API which parses through downloaded data
- Full CRUD operations for movie data
- A database storage layer to store queries
- Code to fulfill the below movie queries (See Queries)

Is-Not

- A graphical interface (GUI / UI)
- A completed app usable from a web browser
- A ‘true’ tech stack

CRUD

1. **Create**
    a. Ability to add new movies via well-structured REST endpoint
2. **Read**
    a. Ability to query for a specific movie by ID or title (see query 1) via REST
3. **Update**

   a. Ability to update an existing movie's data based off a REST request
4. **Delete**
   a. Ability to delete a movie's date by ID using a REST request.

Queries

1. **Basic movie info by title**
   a. GET movie with title as a parameter
   b. Query the IMDB for the movie
   c. Format basic info: title, year, director, language
   d. Return the data to the User
2. **Academy Awards Oscars Nominees (and Winners) by Year**
   a. GET academy award nominees with year as parameter
   b. Parse through Academy Awards data
   c. Return nominees and winner to User
3. **Oscar winners for Best Picture by Year**
   a. GET request Oscar winners for Best Picture with year as parameter
   b. Query the Academy DB by category and year
   c. Receive list of matching movies
   d. Query IMDB for id# or title to each movie
   e. Build record linking matching movies to corresponding IMDB record
   f. Return data to User
4. **Oscar winners for Actor by Year**
   a. GET request Oscar winners for Best Actor with year as parameter
   b. Query the Academy DB by category and year
   c. Receive list of matching movies
   d. Query IMDB for id# or title to each movie
   e. Build record linking matching movies to corresponding IMDB record
   f. Return data to User
5. **Top movies recommended based on previous queries**
   a. Store all queries
   b. Create a custom REST endpoint for a user's recommendation
   c. Produce a unique approach to recommending movies based off previous queries

Architecture

1. A well-structured REST API
2. Basic Error handling for each REST endpoint
3. Well documented REST end points
4. A simple database to store saved queries and other data from external resources
5. Clean code written in an object-oriented database
6. Clean architectural layers between different aspects of the code base
7. Unit testing key areas of the code base

Resources

There is no official IMDB REST API. Instead, the Open Movie Database (OMDb) offers a free alternative API.

IMDB:

- http://www.omdbapi.com/

There is no single Academy Award REST API. Instead, here are a couple of data sources. Simply download them. Instead of calling an external API, you simply crawl through the downloaded files.

Academy Awards:

- https://datahub.io/rufuspollock/oscars-nominees-and-winners#data-cli
- https://www.kaggle.com/unanimad/the-oscar-award

GRADING

Product 50%

Your product must deliver on the **core functionality** based off the client need. This project should be taken seriously keeping ease of use in mind. There is no programming language required. It is better to restrict and implement than to try and accomplish everything at once. For example, it is a good idea to start by implementing simple 'pass-though' APIs that calls other external APIs. Then expand from there and try and implement each query. Ultimately, the grading for this section will be determined by me and how well you can demo your progress. I will be inspecting your Git repo to determined adequate code commits lines up with expectations.

Process 30%

Must store source code in a Git repository Instructor must receive invite to/location of repository.

i. There must be evidence of ongoing repository activity for each sprint.

Must use Trello to track and adhere to Scrum process.

i. Include a product backlog with all stories (user stories, technical debt, bug fixes, etc.)
ii. Use hours estimation for tasks
iii. Tasks must be completed and marked done at the end of each Sprint

Must incorporate some unit testing for product business logic

Presentation 10%

Oral project presentation.

i. Slides
ii. Discuss project and or process
iii. Demo a working product to the instructor

Extra Credit

Announced after Sprint 1 will be a series of optional extra credit opportunities for the project. Stay tuned for more details.

Sprint Breakdown

Sprint	Start	Finish	Goals / Notes
Sprint 1	September 26th, 2022	October 10th, 2022	Set up GitHub, Trello, API key permissions, and downloads Determine tech stack
Sprint 2	October 10th, 2022	October 24th, 2022	Set up web server. Hello world rest interaction. Set up database
Sprint 3	October 24th, 2022	November 7th, 2022	Establish at least one working query
Sprint 4	November 7th, 2022	November 21st, 2022	Finish query development Start extra credit work
Sprint 5*	November 21st, 2022	December 14th, 2022	Finishing polish and error handling Construct presentation and demonstration
*Extended into finals week			