**Product and Process Items for Credit:**

Min Requirement:

1. Product
   1. Rest endpoint that delivers a collection resource in JSON ( Sprint 3, Card# 29)

* <https://oscarproj1.herokuapp.com/movies/categories/bestpicture/1997>
  1. Rest endpoint that delivers a singleton resource in JSON (Sprint3, Card#29)
* <https://oscarproj1.herokuapp.com/movies/categories/bestpicture/1997/winner>
  1. Rest endpoint that allows search of 1 Oscar category and returns results of nominees in JSON (Sprint3, Card#29)
* <https://oscarproj1.herokuapp.com/search?name=Iron>, <https://oscarproj1.herokuapp.com/search?category=direction&winner=True&year=1987>,
  1. Minimum level of documentation for API input and output (Sprint4, Card#57)
* <https://oscarproj1.herokuapp.com/apiHelp>

1. Process
   1. Product Vision (Sprint 1, Card# 4)

* In first deliverable
  1. One persona (Sprint 1, Card# 3)
* In first deliverable
  1. Source code in repository on Github.com (Sprint 2, Card# 34)
* <https://github.com/jordan-oc/The-Ocsar-Project--CSC131->
  1. Flying donut
* <https://www.flyingdonut.io/app/projects/5f84c8583fa2161715aa7988>
  + 1. Backlog with stories
    2. Story point estimation
    3. Hours for task estimation
  1. Implement one story per sprint
     1. Sprint backlog
     2. Tasks division for story
  2. Unit Testing for 2 classes (Sprint 4, Card# 52)
* searchMovieTest.py and imdbTest.py on Github

Extra Credit:

1. Product
   1. Provide link to external site (IMDB) (Sprint 3, Card#22)

* <https://oscarproj1.herokuapp.com/>
  1. Search feature allows filtering movies by date (Sprint 4, Card# 32)
* <https://oscarproj1.herokuapp.com/search?category=direction&winner=True&year=1987>
  1. More than one category can be searched (Sprint 4, Card# 32)
  2. More than one endpoint that delivers collection resource (Sprint 3, Card#29)
* <https://oscarproj1.herokuapp.com/movies/categories/bestpicture/1997>
* <https://oscarproj1.herokuapp.com/movies/categories/actor/1997>
  1. More than one endpoint that delivers singleton resource (Sprint3, Card#29)
* [https://oscarproj1.herokuapp.com/movies/categories/bestpicture/1997/winner](https://oscarproj1.herokuapp.com/movies/categories/bestpicture/1997)
* <https://oscarproj1.herokuapp.com/movies/categories/actor/1970/winner>
  1. Graphical user interface (Sprint 4, Card#18)
  2. HTML page documenting API endpoints with inputs and outputs (Sprint 4, Card# 57)
* <https://oscarproj1.herokuapp.com/apiHelp>

1. Process

d. Mockups for GUI (Sprint 2, Card# 24)

* API Page: <https://www.figma.com/file/myTGAHLlJKAurqmawVYSCT/Untitled?node-id=0%3A1>
* Home Page: <https://www.figma.com/file/RFQ9Sroe8aSHzPntmZTqZZ/Oscars>

c. Incorporate Contextual Inquiry/Elicitation techniques to create personas (Sprint 1, Card# 3)

* In first deliverable

j. Product/Rest service is publicly accessible via HTTP (Sprint, Card# 10)

* <https://oscarproj1.herokuapp.com/movies/categories/bestpicture/1997/winner>
* <https://oscarproj1.herokuapp.com/>

k. Explain and document obstacles encountered during the project and how those obstacles were handled

* In obstacles and solutions section

**Obstacles and Solutions:**

One of our first issues we encountered was the supplied json file. We had code sort though the json fill to give us the information we needed to accomplish our product vision. Two other issues arose during a code review. In our getURL method for retrieving the link for a desired movie, it would give an incorrect link to IMDB. This was an easy fix, movies with multiple words or more specifically a “space” in their title would return a link to a movie with the same first string but not the right movie. It was due to a “&”, we needed a  “+” for the OMDB API . The following issue was found in the code review was another link problem. The problem was the correct movie but for the incorrect year an example would be trying to find 1997 Titanic and we would get 2012 Titanic. This was a harder fix, It wasn't necessarily our code but the json file we were using as our database. It was due to a variance on the year nominated versus year awarded on IMDB. We solved this by grabbing the corrected year out of the original json file. Once the json file was modified the problem was solved. Finally the last problem found was also about receiving the incorrect link, but this time it was due to not including an apostrophe in titles. This was fixed by modifying our getURL method to add the character.

**Scrum Artifact:**

<https://www.flyingdonut.io/app/projects/5f84c8583fa2161715aa7988>

**Version Control (Github):**

<https://github.com/jordan-oc/The-Ocsar-Project--CSC131->

**Website Link:**

<https://oscarproj1.herokuapp.com/>

**Mockups:**

* API Page: <https://www.figma.com/file/myTGAHLlJKAurqmawVYSCT/Untitled?node-id=0%3A1>
* Home Page: <https://www.figma.com/file/RFQ9Sroe8aSHzPntmZTqZZ/Oscars>

**Persona Contextual Inquiry Notes:**

Interview 1:

1. Name and Age: Justin, 21
2. Location of interview: discord
3. Duration of interview: 15 min
4. Questions and Answers:
   1. Who are they? They are a college student.
   2. What do they want to accomplish? They want a quick and easy way to find good movies to watch. They want to find a movie and then immediately watch it. They also want to earn a degree.
   3. Where are they performing the task? They will be using the app at home.
   4. How often do they perform it? They will use the app on the weekends during the evenings.
   5. Are there time constraints? They will mostly use the app on the weekends and evenings due to school work.
   6. Where do they live? Sacramento California

Interview 2:

1. Name and Age: Jordan, 26
2. Location of interview: Discord
3. Duration of interview: 15 minutes
4. Questions and Answers:
5. Who are they? College Student
6. Goals? Graduate from Sac State with a Bachelors in Computer Science, get into the software engineering field
7. Where do you live? Sacramento California
8. Education backgrounds? 3 Associates of Science degrees
9. What do they want to accomplish? Easy way to browse a list of award winning movies without a lot of hassle. Must be quick and efficient to be able to use after work or on weekends, headache free.
10. Where are they performing the task? Either on a laptop or desktop computer. Ideally they would be able to use it on an app, so it could work with a smart TV.
11. How often do they perform it? On their break at work so they could have a list of movies to watch when they have time or after work/school.