**Final Summary Report**

1) **How was your experience with the project?**

I selected the San Francisco Movies Challenge because I had a semblance idea of how to solve this challenge. In my last semester of university, I had taken a Web Engineering course and had been tasked to implement a location-based map application using Google Maps API. We were supposed to create functionality and UI thus manipulating different objects and events for purposes of the requirements. This gave us an opportunity to work with a new/unfamiliar API. In addition, I noticed the pertinent information of movies and locations were stored in a CSV file, and this also reminded me of a time of extracting key-pair values (but in Java) that I had worked with. Overall, this was a rewarding challenge, it allowed me to travel into unfamiliar territory, especially with regard to API, manipulating objects and events, getting stronger with version control systems like Git and being able to have the experience to build professional grade software. I also like how this project gave me experience with regard to location services and would feel confident in future endeavors.

2) **Which parts you found most difficult?**

One of the biggest challenges to overcome was getting the Google geocoding services to work. Through some searching, I initially fell under the impression that Google Geocoding API was a paid service. This was something I was going to address to Simone and Austin, however after doing more careful researching, I ultimately found that I only needed to “enable” the usage of Geocoding if I already had access to the Google Maps API services. Preceding having found the enable services button, I had to keep changing how I ask the question of what I want and how can I achieve it. Simply put, I found that my search question was Google Geocoding Services to handle requests and responses. I understand fully that navigating through Google’s API documentation and services can be challenging and that it is constantly changing. Another challenge was building the autocomplete for movie searches. I also needed to account for character casing such as upper, lower, spaces between films (The Pursuit of Happyness). Though the frontend presentation can be improved, I can assuredly say that the project does meet the requirements, can easily be adjusted to match a improved design, and the code quality is clean and consistent.

3) **Which code segments you are most proud of?**

First and foremost, I am proud of being able to initialize the mapping and connecting the JavaScript with much ease. I remember when I had first done this assignment in university, I was finding much difficulty in attaining the API key and implementing a mapping service. Because of that prior academic experience, I had a much better idea of how to get past the API implementation, navigate more fluidly throughout the Google documentation, and build the tools based on requirements. I am definitely proud that I was able to build the proper search function, implement the geocode services for films and locations, building the connection between server requests and responses while returning it to the front-end. I am also proud of the fact that I completed the autofill requirements by the autofill function and adding the necessary clear markers upon a new movie search. In addition, I ran a simple yet malicious SQL injection to see if pertinent login credentials had been compromised: <img src onerror="alert(document.cookie)"> This gave me the added security that security vulnerabilities are limited. Overall, I am proud that I was able to work outside my comfort zone and attain workable solutions.

4) **Please include a description and a solution of the problem, and reasons behind your technical choices and architecture.**

My first major problem was figuring out how to retrieve the values of the Film and its Location(s) from the CSV file and organize them into key-pair values via JSON parsing. Finding the fundamental question that ultimately became how to use Google Geocoding Services for requests and response was the first breakthrough to this challenge. Unfamiliar with implementing different services within the same API through objects and events was newer for me. I found the proper addressing that would allow for this, but finding working example code and testing my own API key were experimental for me. Fortunately, when I found the correct encodeURIComponent and pass the value of my location object, I began to isolate my problems and find working solutions. I even had to learn the proper JSON targeting structure when making the server requests via the onreadystatechange event handler. Overall, I made sure that my variables and objects were as descriptive, distinguishable and clear as possible throughout my program and between frontend and backend. In my variable naming convention, I found it best to use camel-casing simply because I understood JavaScript by that convention. I choose my careful spacing and comment structure from prior experiences learning from industry professionals; I wanted to always apply good practice and readable code. Above all, you will see strong consistency and readability that falls well within the single responsibility principle. In addition to the frontend, because time was a limited resource, I made sure that the focus of the code was as close to matching of the requirements, so I wasn’t able to “beautify” the page more, but even in the index.html file, you’ll see that the code structure is still clean, consistent and maintainable.

My **index.html** file is stored in CodeChallenges\MetaNutrition\public

My **index.js** file that works with backend is stored in the MetaNutrition directory.

My **maps.js** file that works with the frontend and targets the maps is stored in CodeChallenges\MetaNutrition\public\javascript

Hope you like it all.