**Project Idea #1 Satellite / Space Station Tracker**

API Link:<https://www.n2yo.com/api/>

Summary:

Tracking Satellites and Space Stations in real time. Data is sent back in the form of JSON strings. It returns a lot of useful info, but the main info we need to know at the moment is the satellite ID, Satellite Name, Latitude, Longitude, and the altitude above sea level etc. Transactions limited to 1000 per hour. NASA also has the same API.

When people come to a new place, they type in where they're from, and they come up with some good local restaurants of their hometown style based on some reviews, and they come up with some interesting local places.

(Can map out when satellites are overhead specific restaurants, for conspiracy theorists?)

NASA API Link: <https://sscweb.sci.gsfc.nasa.gov/WebServices/REST/json/>

Expanding:

→ Implements a predictive algorithm to tell where the objects are going.

→ NASA has many other API options including:

1. AsteroidsNeoWs: Near Earth Object Web Service.
2. APOD: Astronomy picture of the day
3. EONET: The Earth Observatory Natural Event Tracker

And many more

Link to NASA APIs: <https://api.nasa.gov/>

→ Incorporate the planets

→ Google Maps API to show the locations of the objects

Link to Google Maps API

**Q/A**

**What is new/original about this idea?**

This idea will be a teaching tool that will be able to let people learn about how Space Stations and Satellites travel around the earth. Each Satellite or Space Station will have an info page, with lots of useful information pertaining to each object.

**What are related websites/apps? (Be able to answer the question: isn’t somebody already doing this?)**

There are a few other sites that have done this. NASA has a feature on their site that does this and another site called N2YO does this.

**Why is this idea worth doing?**

Satellites and Space Stations are something that not many people take the time to learn about. It would be interesting to learn about trajectories, and normal paths that most Satellites/Space Stations take.

**Why is it useful and not boring?**

Anything related to space is very interesting. It could be a useful tool to teach about the trajectory and how/why objects orbit

**What are a few major features?**

→ Tracking Satellites and Space Stations

→ Each Satellite and Space Station will have its own page, and hold all of the corresponding info for each one.

→ Implements a predictive algorithm that can tell where the objects path is going to go, so we don’t have to use so many requests.

**What resources will be required for you to complete this project that are not already included in the class. i.e. you already have the Microsoft stack, server, database so what else would you need? Additional API’s, frameworks or platforms you’ll need to use.**

API Link:<https://www.n2yo.com/api/>

NASA API Link: <https://sscweb.sci.gsfc.nasa.gov/WebServices/REST/json/>

Link to NASA APIs: <https://api.nasa.gov/>

NASA has many cool APIs that could be incorporated as ‘features’ for this application.

**What algorithmic content is there in this project? i.e. what algorithm(s) will you have to develop or implement in order to do something central to your project idea? (Remember, this isn’t just a software engineering course, it is your CS degree capstone course!)**

The Algorithm would be a predictive one, that allows us to plot the trajectory of the objects. We then would only have to use a few transactions per hour for each one.

**Rate the topic with a difficulty rating of 1-10. One being supremely easy to implement (not necessarily short though). Ten would require the best CS students using lots of what they learned in their CS degree, plus additional independent learning, to complete successfully.**

7/10

**Project Idea #2 Brewery “ROAD TRIP!!” planner**

API Link: <https://api.openbrewerydb.org/>

Summary:

Creating a Road trip map for for people who love breweries. Data is sent in the form of JSON string. The user would select from a list of breweries inside of the desired range, then the algorithm would find the shortest possible path in between the selected locations. We would also allow for the user to sort the breweries into subsets that match their style such as pubs, breweries, cider houses, and so on.

The api for that we could use for this, that is listed above, is free.

When people find the brewery they want, I guess they have to eat something nearby as well. When they type in the types of food they want, they are offered a selection of great restaurants nearby.

When people come to a new place, they type in where they're from, and they come up with some good local restaurants of their hometown style based on some reviews, and they come up with some interesting local places.

Expanding:

→Implements shortest path (Dijkstra's Algorithm)

→ Road conditions map into the algorithm

→ Maybe: Uses weather API to get current weather at locations

→ Uses Yelp API to show information about the place, as well as a few reviews.

Yelp API Link: <https://www.yelp.com/fusion>

→ Maybe: Uses Eventbrite API. Allows users to create and search for events around the places they are planning to go.

Link to API page: <https://www.eventbrite.com/platform/api>

Link to creating an event through API instructions: <https://www.eventbrite.com/platform/docs/create-events>

**Q/A**

**What is new/original about this idea?**

Have you ever seen Dijkstra on a bar crawl? No?

**What are related websites/apps? (Be able to answer the question: isn’t somebody already doing this?)**

Yelp, Eventbrite, pubrally. There are similar apps, however, we have original ideas that will set ours from the rest. We would add road conditions and suggestions for other locations you might like.

**Why is this idea worth doing?**

It can be a hassle for people to find their favourite brewery and they spend a lot of time looking up information. This idea makes people's lives more convenient, and let people who love breweries find what they want more quickly and accurately.

**Why is it useful and not boring?**

It’s useful as it simplifies the ability to bar-crawl your way across Oregon, without the need to use google maps’ clunky UI. Dijkstra on a binge would only be Boring between Gresham and Kelso.

**What are a few major features?**

Our app would provide the user with planning out a road trip to various breweries, by finding the shortest path. Integrating other apps, such as Yelp, would allow more depth via reviews of nearby places to eat. You can’t drink without some good burgers, right?

**What resources will be required for you to complete this project that are not already included in the class. i.e. you already have the Microsoft stack, server, database so what else would you need? Additional API’s, frameworks or platforms you’ll need to use.**

Yelp API, Eventbrite, Roads API

**What algorithmic content is there in this project? i.e. what algorithm(s) will you have to develop or implement in order to do something central to your project idea? (Remember, this isn’t just a software engineering course, it is your CS degree capstone course!)**

In this project we would have to implement Dijkstra’s  Algorithm. We also would have to implement a “You might like this” algorithm that displays other locations that might be of interest based on the original choices for the crawl.

**Rate the topic with a difficulty rating of 1-10. One being supremely easy to implement (not necessarily short though). Ten would require the best CS students using lots of what they learned in their CS degree, plus additional independent learning, to complete successfully.**

6/10