# MIS2402 Semester-Long Project

# Create a simple application that uses Web APIs

This semester long project is an individual project. Students should work on it independently. It is ***not*** a group project. Students should read this whole document at the start of the semester and review it periodically later in the semester.

As you will discover in this course, web APIs allow programmers to create applications that can perform a variety of tasks. There are APIs designed to look up data (USA state codes, zip codes, etc.), and there are APIs that perform actions. Actions include tasks such as creating charts and graphs, sending text messages, generating QR codes, and many other things.

Your semester-long project is to create a ***dynamic,*** web-based chart of API data that is of interest to you. It is a ***dynamic*** chart because the data used to generate will change over time. Consequently, the chart itself will (correctly) change over time without any intervention from you, the programmer.

Students must use the Image Charts API. See: <https://documentation.image-charts.com/> Students should plan on spending a significant amount of time researching the Image Charts documentation to learn how to achieve the chart effect(s) they desire.

The second API will be chosen by the student. This second API will provide the data to be used by the chart.

Students will discover that these data sources provide much more information than they need. An important part of the semester-long project is exploring the data found at these sources, determining what portions of it are of interest, and (important!) using JavaScript to get the data into the right "shape" to be used by the image chart API.

### **Phase 1 – Planning One (Due Friday 2/10)**

Students should find an API which provides data that can be later analyzed and visualized.

As a result of this step, students are expected to submit the following items:

1. A brief description of selected API.
2. A link to the documentation of an API.
3. URL that returns data.

Students may consider one of the datasets from the following sources:

* Data USA:
<https://datausa.io/about/api/>
<https://datausa.io/api/data?drilldowns=Nation&measures=Population>
* OpenDataPhilly:
[https://www.opendataphilly.org/dataset](https://nam10.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.opendataphilly.org%2Fdataset&data=04%7C01%7Cjeremy.shafer%40temple.edu%7Cdea4f5ca62ce4f2d846608d9cfbefdb0%7C716e81efb52244738e3110bd02ccf6e5%7C0%7C0%7C637769241472034908%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000&sdata=MbXiBYcVyZHJeh91SIGb4zN57N37LEp20e2k84OycVE%3D&reserved=0)

However, students are not limited to these two sources and encouraged to search for an APIs that they find personally interesting. Students should avoid using the exact same API as their peers. If two students do use the same API, they will be expected to take different approaches in analyzing and presenting the data.

Students must choose an API that returns data in the **JSON** format.

|  |
| --- |
| Here is an example of expected submission for Phase 1:1. Film Permits API from NYC OpenData provides a dataset of permits issued by NYC for an exclusive use of city property, such as sidewalks, streets, or parks. This dataset contains information about individual permits, including their data, location and category of production selected by the permit applicant.
2. Documentation: <https://data.cityofnewyork.us/City-Government/Film-Permits/tg4x-b46p>
3. JSON Data: <https://data.cityofnewyork.us/resource/tg4x-b46p.json>
 |

You will submit your summary to canvas for your instructor to review.

### **Phase 2 - Planning Two (Due Friday 3/24)**

Students should examine the Image Charts documentation and plan what sort of chart(s) they plan to produce. Students should provide a short summary (two or three paragraphs) to describe their plan. The summary should include:

1. What kind of chart you intend to make and why.
2. Why what you are presenting is interesting.

Choosing the right type of chart is important. Some chart types are best suited for **comparison** (comparisons of magnitude). Others show **composition**. And other chart types show **trends**. You should be clear about which of those three things you plan to illustrate primarily.

Make an interesting presentation. Please recall that the first basic rule of Data Visualization covered in Data Analytics (MIS2502) is that a chart should "Tell a story". That means that the depiction should enable meaningful comparison (and conversation!), and that the chart should yield insight beyond the text.

Here's something to think about: “If the statistics are boring, then you’ve got the wrong numbers.” (Tufte 2009)

It is not necessary, in this phase, to know what your exact data values are, or to have arrived at a grand conclusion (A silly/stupid example: "COVID mortality rates are lower in states containing the letter R"). In fact, you should expect the underlying data to change between now and the end of the semester. Your goal in this phase is to have a plan, and to think about what you might want to show in the data that would be interesting and/or useful.

You will submit your proposal to canvas for your instructor to review.

### **Phase 3 – Final Version (Due Monday 5/1)**

This is by far the most difficult phase of the project. Here you must create a page, with a single "refresh" button. The click of the button will initiate the retrieval of data from your API and the rendering of at least one chart. Students should include some additional descriptive text on the page so that it is clear to the instructor (or any other reader) what the purpose of their chart is.

Kindly recall the Tufte's principles of Data Visualization as summarized in the Data Analytics Course (MIS2502)

* The chart should tell a story
* The chart should have graphical integrity
* The chart should minimize graphical complexity

Tufte’s fundamental principle: Above all else show the data!

If your chart does not make sense without a lengthy amount of text to support it, then it probably needs to be improved.

Students are encouraged to focus their attention on portions of the data set that are the most relevant. For example, if you were working in finance, you would ***not*** want a chart that shows 6,000 stock ticker symbols. It would be more useful to show the 10 ticker symbols that performed the best in the last 30 days. You will need to write JavaScript code to get the data elements out that you need. Your solution needs to be flexible enough so that it is not locked in to the same 10 symbols forever. The top 10 performers today, may not be the top 10 performers next week!

Verify (to the best of your ability) that your chart data is changing over time. Turn in your work as directed by your instructor.

### How will the project be graded?

### Participation in Phases 1 and 2.

You will receive 0, 5, 10 or 15 points for completing phases 1, and 2. The objective of these phases is to show thoughtful, deliberate progress towards the development of your solution (i.e. the instructor understands that you are showing a “work in progress”, not a finished product in these phases!). Most students who follow along with the class, show progress, and turn in their work on time will earn 15 points easily in these phases. However, the instructor does reserve the right to award fewer points at each phase for submissions that have glaring imperfections or appear to be hastily / haphazardly constructed.

***Late submissions will receive zero credit.***

### Assessment of Phase 3

Phase 3 must be turned in on time. Late submissions will not be accepted.

Student work will be evaluated by the instructor based on the following criteria.

1. Functionality – Does the application work? Does it reflect an understanding JavaScript and APIs? Does the chart data change over time?
2. Utility – Your chart(s) should "tell a story". Your visualization work should stand on its own and not rely heavily on lengthy written explanation.
3. Accuracy – You must not misrepresent the underlying data. Mistakes to avoid: calculation errors, choosing an inappropriate chart type, omitting important data values, or including the same data values more than once.
4. Visual Appeal - Simple solutions are acceptable … this is not an artistic design class! But the HTML portion of your work should demonstrate attention to detail and thoughtfulness. Similarly, the chart(s) you create should minimize graphical complexity and avoid "chart junk".
5. Originality – **Students are expected to do more than merely mimic the work of their peers**. If you are doing your own work, analyzing your own API data source, and creating your own chart(s) then you can expect full marks for originality. If you are not doing those things, you may lose points in this category or – far worse – violate the academic integrity of the course. Consult the syllabus for the consequences of academic integrity violations.

Your solution will be graded as follows. All points are awarded at the discretion of the instructor.

|  |  |
| --- | --- |
| Phase 1 – Planning (0, 5, 10 or 15 points) | 15 |
| Phase 2 – Planning (0, 5, 10 or 15 points) | 15 |
| Phase 3 – Final Version |  |
| Functionality (0, 5, 10 points) | 10 |
| Utility (0, 5, 10 points) | 10 |
| Accuracy (0, 5, or 10 points) | 10 |
| Visual Appeal (0, 5, 10 points) | 10 |
| Originality (0, 5, 10 points) | 10 |
| HTML is error free - Firefox view source will be used to check for errors. (0, 5 or 10 points) | 10 |
| JavaScript is error free - The Web Developer Console will be used when testing your application. (0, 5 or 10 points) | 10 |

### How can I earn MIS PRO Points?

To be eligible to receive MIS PRO points, a student must first complete the project as described above and earn a minimum of 80 of the 100 possible project points. In addition to that, a student must also:

1. Create a *second* chart that compliments the information presented in the original project assignment. PRO Point solutions should present at least *two* charts, total.

Here are the acceptable chart combinations: ***trend*** and ***comparison*** or ***trend*** and ***composition***.

PRO points submissions must include a chart that represents a ***trend.***

1. Create a GitHub account and upload your project code to a publicly accessible repository there.
2. Include a readme file in that GitHub repository. It must describe the purpose of project. It must also identify the APIs used in the project, describe how they were used, and why they were chosen. (The APIs should be combined to achieve some collective result!)

Students will also receive 0, 25, or 50 PRO points based on the instructor’s assessment of their work.

### Submit your work / Request your points

When you are ready to request your PRO points, do the following:

1. Make sure that have uploaded your work to the class server, and that you can provide the URL to your working project.
2. Make sure that have uploaded your work to GitHub, and that you can provide a URL to your code's repository there.
3. Log in to the <https://community.mis.temple.edu> site. Open your Dashboard. Open “Professional Achievement Points” and “Add New”. Select the option to submit a PRO point request relevant to this course, MIS2402.

Your PRO points request should include a link to your working code the class server, a link to your code's GitHub repository, and a brief description of your work.