

Programming Project 6 (last one)

The end is near.

Program 6, Vue.js

You need to take a JSON data file, and using a Github website you create, your user should be able to read traffic ticket data and display it on 2 different pages. Both pages will contain the links and script tags necessary for Bootstrap and Vue.js, as well as a Bootstrap NAV bar that allows navigating between the pages. The first page displays all of the data, the second page displays some of the data that has been filtered.

There is an example or two similar to this in the second week of lecture notes on Vue (Vue2.html). Please watch this video overview of the assignment.



Our Website	All Tickets	25mph or move over
-------------	-------------	--------------------

Tickets July 4, 2014, Minneapolis/St. Paul

Ticket ID	Time	Posted Speed	MPH Over	Violator Sex
20124	0:11:00	70	25	Female
20125	0:36:00	70	25	Male
20126	1:02:00	55	21	Male
20127	1:02:00	60	17	Male
20128	1:15:00	70	13	Female
20129	1:28:00	55	33	Male
20130	1:33:00	70	27	Male
20131	1:52:00	30	44	Male
20132	3:42:00	55	11	Male
20133	3:51:00	40	10	Male
20134	6:23:00	65	22	Male

Clicking the *25mph or more over* link in the nav bar above should request the second page, which should be titled **25over.html**.

Our Website	All Tickets	25mph or more over
-------------	-------------	--------------------

Tickets July 4, 2014, Minneapolis/St. Paul, 25 mph or more over the limit

Ticket ID	Time	Posted Speed	MPH Over	Violator Sex
20129	1:28:00	55	33	Male
20130	1:33:00	70	27	Male
20131	1:52:00	30	44	Male
20143	7:15:00	60	26	Male
20151	8:00:00	70	26	Male

The data file needed, *tickets1.json*, has the data already set in JSON format. Vue will destringify it for you. As you know, you will need to upload that JSON data file and your *index.html* and *25over.html* pages to your Github web site in order to test your code. I recommend getting this to work a little at a time. Test a little bit, maybe on one page only, then when it works, go on the the next step.

If you have not done so, study the material and code form Vue2 before starting this project.

Your *index.html* page should display like this, more or less, but it should display a nav bar and all the 50 rows of data from the data file (linked to below). This page should be titled ***index.html***.

You need to use Bootstrap's NAV bar on both index and 25over pages, but you do not necessarily have to use the inverted NAV bar as shown above. You must also use Vue.js.

Both pages need to have the links and script tags necessary to load the required Bootstrap and Vue.js libraries.

Important: In total, for full credit, this project will require you send me **3** files in a single zipped folder: *25over.html*, *index.html*, and the data file, *tickets1.json*. In the email that you use to submit these files you also

must *include a working link* to your Github index page. I will use this link to grade your program. I will look at the files you send me only if there are issues.

And, I will give 85% credit if you get only the index page (the page with all the data) working. To get 100% you need to get both pages working. So, I recommend you get the index page working before starting on the gpa page.

Data File link. You will also need the JSON data file, *students.json*. The name of the file is `tickets1.zip`, located in the same folder as this document.

index.html page link. As mentined, I will give you the index.html page, and if you get this one page working it will be worth 85%. For 100% both pages need to work. The index page is also in the same folder as this document:

index.zip

You'll need to unzip the data file before use. Once decompressed, it will have this name: *tickets1.json*. Do not alter the data itself. The data file itself is an array that contains 50 or so JSON-ized objects. Here's an example.

```
{
  "actual_speed": 95,
  "age": 20,
  "posted_speed": 70,
  "stop_date": "7/4/2014",
  "stop_time": "0:11:00",
  "tid": 20124,
  "violator_sex": "Female"
}
```

As we said, the second page, *25over.html*, will use *v-for* to display only those records that have were more than 25 mph over the limit.

Reminder: *you must* zip the folder with your solution files before attaching it to an email and sending it to me. Also, you must have your Github website functioning when you submit the project. I will use your URL to grade the program.

Also Important: Obviously, you will need a Github account to be able to make use of their web hosting service. How to do this is explained in videos in the lecture notes for week 14, entitled "*Your Web Site*".

Grading will follow this approach.

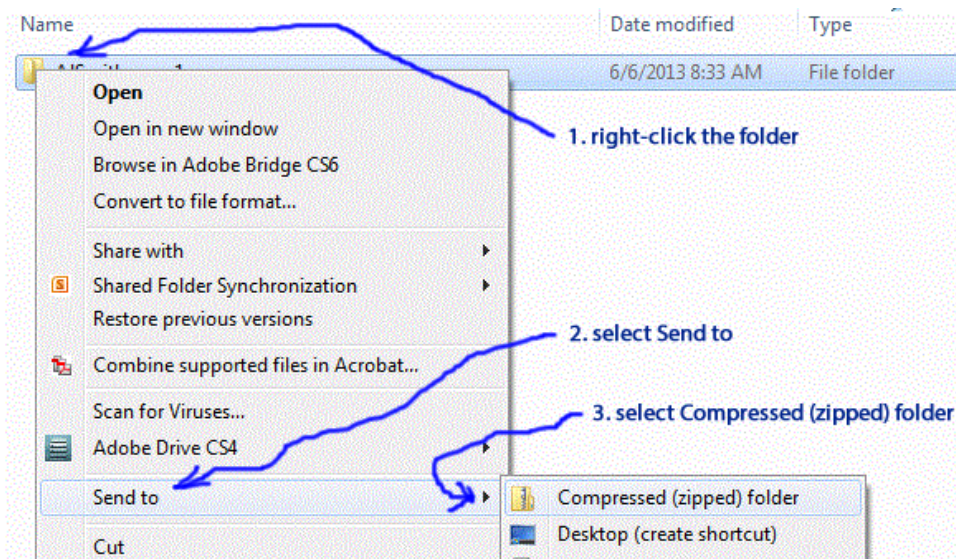
- 100% program runs perfectly with all required items

- 90% program meets all requirement but doesn't always produce the correct output
- 80% program runs without errors, but is missing some requirement.
- 70% program had a good start, most but not all of the code was correct, had one or more fatal errors.
- 40-60% or below. Not very close, many parts missing, would not run, probably a late start.
- 0% no submission or code was not the student's work.

How To Submit Your Program

To be clear: since you are submitting several files, you cannot just attach them to the email. You'll need to compress the folder containing them and attach the compressed folder to an email in order to properly submit it. Name the folder containing your files using your name and project number. For example the folder might be named `AJSmith-prog6`. ***Please put your name on the folder.*** I get lots of programs; putting your name on the folder help keep things organized and shortens the amount of time it takes to get things graded. As mentioned, you need to *type your Github URL in the email* your compressed folder is attached to.

After creating the folder, Zip (compress) the folder by right-clicking on it (on Windows), and then choosing **Send to** from the pop-up menu, then select **Compressed folder**. This will crate a new zipped folder identifiable by the little zipper on the folder icon with a .zip extension.



Then, attach the zipped folder to an email and send it to me.

When you submit the program for grading you need to do two things:

1. Attach your index.html file to the email just like we have done for earlier programs
2. also your email needs to contain a link to your github website. I should be able to click that link and run your program. The point of this exercise is to get a function web site

There are code examples in the notes for "Your Web Page". That is where you should start if you have not already studied those notes.

Please, please start early on this.

If you have trouble, attach a copy of the index.html page only to an email, then send it to me.