

FSW 115 Grading Rubric

Bryan U school policy is that if an assignment is submitted on time it may be resubmitted as many times as desired through the remainder of the course for full points. New deadlines are easily arranged by messaging me on Slack or Canvas Inbox before the original due date.

Assignments submitted after a deadline are subject to a 10% late point deduction per day up to a maximum of 30% deducted. Late assignments can still be resubmitted for mastery and make up points, but late point deductions can not be recovered.

Week 1

Assignment: **JSON's Deli**

Possible Score: 50 points

Submission: Push code to GitHub. Submit GitHub link to the assignment in canvas.

<i>Criteria:</i>	<i>Points Possible:</i>
Saved as a Json file	10
5 options for menu categories	5
Each menu option is an array	5
2 objects in each submenu	5
Name of the item	5
Ingredients is an array of strings	5
Description is a string	5
Data is valid JSON format	10

Week 2

Assignment: **Finding and Testing API's**

Possible Score: 50 points

Submission: Push code to GitHub. Submit GitHub link to the assignment in canvas.

<i>Criteria:</i>	<i>Points Possible:</i>
Find an API	12
Get Data in Postman	14
Paste data in a JS file	12
Make the data show up on the webpage	12

Assignment: **Create a Stylized List**

Possible Score: 75 points

Submission: Push code to GitHub. Submit GitHub link to the assignment in canvas.

Criteria:	Points Possible:
Find an API	20
Get Data in Postman	20
Paste data in a JS file	10
Make the data show up on the webpage	20
Find a unique API*	5

*Unique API is one we don't use in this course (I.E. not Rick and Morty, StarWars or Pokemon).

Find one listed as "Auth: No" here: <https://github.com/public-apis/public-apis>

Week 3

Assignment: **Pokémon List**

Possible Score: 50 points

Submission: Push code to GitHub. Submit GitHub link to the assignment in canvas.

Criteria:	Points Possible:
Function that Gets Pokémon JSON with XHR	15
Parse Pokémon JSON to create an Array	10
Make each Pokémon name show up on the DOM	10
Styled well	5
Clean Code*	5
Display more data about each Pokémon	5

*Clean code means using variable names that are descriptive, code blocks are indented and tabbed properly and no excessive white spacing

Assignment: **Postman Practice**

Possible Score: 60 points

Submission: Upload screenshots of the Postman requests to either GitHub or to Canvas on the assignment submission. Provide your link if you choose the former.

Criteria:	Points Possible:
GET a list of SWAPI planets	10
GET a single SWAPI person object	10
POST 3 new to-do list objects	10
GET one single to-do list item using its id	10
DELETE one single to-do list item using its id	10
PUT one single to-do list item using its id	10

Week 4

Assignment: **Axios To-Do Part 1**

Possible Score: 40 points

Submission: Push code to GitHub. Submit GitHub link to the assignment in canvas.

<i>Criteria:</i>	<i>Points Possible:</i>
User can see their current list of to-dos	10
To-dos show up as soon as the page loads	10
If item is complete display strikethrough	10
Styled well	10

Assignment: **Website with Multiple Requests**

Possible Score: 75 points

Submission: Push code to GitHub. Submit GitHub link to the assignment in canvas.

<i>Criteria:</i>	<i>Points Possible:</i>
Have 2 pages, each with a button	20
Render data from API on DOM when buttons clicked	20
Styled well	15
2 different API's used*	20

* Two different API's must be used, not 2 different endpoints for different datasets from the same API

Week 5

Assignment: **Axios To-Do Part 2**

Possible Score: 50 points

Submission: Push code to GitHub. Submit GitHub link to the assignment in canvas.

<i>Criteria:</i>	<i>Points Possible:</i>
POST - The user can add new to-dos to their list (title, description and price)	10
Each to-do will have a mechanism to mark complete or incomplete	5
PUT - Marking complete or incomplete should update the database	15
Each to-do will have a mechanism to delete	5
DELETE – Should remove from the database	15

Week 6

Assignment: **Create a Single Page Web App**

Possible Score: 75 points

Submission: Push code to GitHub. Submit GitHub link to the assignment in canvas.

<i>Criteria:</i>	<i>Points Possible:</i>
3 different GET requests	10 each (30 points total)
Data from all requests renders simultaneously on the DOM	15
Use Chaining or Async Await	20
Fully Styled and Responsive	10

Week 7

Assignment: **Capstone**

Possible Score: 125 points

Submission: Push code to GitHub. Submit GitHub link to the assignment in canvas.

<i>Criteria:</i>	<i>Points Possible:</i>
Uses 2 different API's	25
Uses a promise based client (node.js, fetch, Axios, or custom promises)	10
Use 1 API to populate some portion of another API (must not be entirely hardcoded)	25
Fully styled and responsive	25
Contains a GET request	10
Contains a POST request	10
Contains a PUT request	10
Contains a DELETE request	10