I looked and used the following to help get me going on this project, it doesn't actually work so the first thing I had to do was to get it working.

https://codepen.io/db001/pen/BzAmYV/

You just need to change the URL in the link and include a "/" after the complete URL so it looks like this: "http://swapi.dev/api/films/3/"

1. Stripped out everything that wasn’t needed such as script links in the html
2. Put the API in to its own gateway file (future improvements meant I could add the starship API at a later date)
3. The number generator was originally part of the gateway with the API, but moved this to its own file (I was going to add a game mode which dealt out 10 cards and followed the rules more closely, e.g. the cards go to the other person) also tailored it so it would select from all the characters in the API.
4. Moved the duplication checker into its own file
5. Developed the score counter so it was the main point for updating the html elements and the variable; for this reason I made it as reusable as I could.
6. Moved nan check to its own file and tidied up the code.
7. Changed the display format so that it would be updating html elements rather than adding in its own html, this was to keep the website using similar code.
8. Created cardSetup so that we just need to call this whether we are resetting the game or getting the next card
9. Created cardNext and gameSetup, both are similar gameSetup updates the scores, both use cardSetup.
10. Logic is what is left of the original JS page
11. Added MainTest & TestingHome pages, using the index.html as the setup for it originally
12. Added the first 3 tests that were checking URL, random number generator and the URL getting changed when it’s a duplicate.
13. Added the title into its own reusable component (I originally added the buttons for reset and 10 card game here, but they struggled to access the required methods, I wanted to separate out some of the functionality so that I could run within a small frame testing to check out the individual components)
14. Checked out how to get individual elements to be called from JS or from within the HTML, tried JQuery, Object and iframes all 3 encountered CORS error, so uploaded it to a free hosting website.
15. Got the iframe working and loaded the index page up
16. Worked out how to access and retrieve data from within the iframe and developed the score counter test.
17. Then got the updateScore counter to work but had to pass the data through the MainTest.JS file itself rather than through the template.
18. Used the above to develop the test for initialisation of the game
19. Developed the test for checking out the results panel, which works by clicking the height button within the iframe, needed to pause the test so that the text could be rendered.
20. The final test checks whether a new card has been drawn after next card button has been pressed, the sleep function for the previous test wasn’t working, the whole website (cont) was being paused, so found a pause function that would use a promise to pause the code but would allow everything else to proceed.
21. Attempted to make changes to the gateway function to turn it in to a class.

Further improvements:

Tidy up the test page by splitting the tests up in to better sections

Adding more tests which can test out the reset feature

Develop existing tests which check out more of the data for example just checking the name has changed when the next button is pressed.

Adding in the ability to allow you to pick people or starships for your game mode

I was thinking about dealing out all the cards to both sides, which would require all the IDs to be checked so that no one had duplicate cards, and would have meant allowing cards to swap hands, and to take in to account ties (usually the player who picked the attribute goes again)

Using a model would have been useful, especially for testing

Using classes