Approach:

Step 1: Read input file car\_input\_v2.txt at location: <<path to input location>>

* Check for multiple input files with ‘.txt’

Step 2: Extract vehicle registration numbers using regex and based on pattern for ex : AD58 VNF, BW57 BOW, KT17DLX, SG18 HTN.

* Loop through multiple input files to identify car registration pattern.
* Create array list with REGISTRATION number of all cars.
* Check for additional space in between registration number and handle it.

Step 3: Launch Chrome browser and go to url: <https://www.cazoo.co.uk/value-my-car/>

Step 4: Enter Car Registration from array list.

Step 4: Get actual output values of each car from website and store in variable.

* Extract full output using xpath. Cut first word and assign it to MAKE variable.
* Rest of the line to MODEL variable.

Step 5: Get the expected output values from the output files and use it to compare the details obtained in step 4.

Step 6: For each input registration value, compare against expected output value. If values match, then PASS else FAIL. Write consolidated test results to html.

Step 7: The results are displayed in a html file created by the script in the same folder: test\_results.html. This contains all the values required for the analysis. Total count of tests, total tests Pass/Fail are displayed on the console.

Step 8: Ensure all the scenarios mentioned in the Output scenario analysis.xlsx are implemented and tested.

Enhancements:

1. Code can be split into multiple files for Extraction of registration number, scrapping details from the GUI and comparing output and reporting. This will be a better structure for scalability of tests.
2. Enhance code for error handling and logging.
3. html output file can be enhanced to be more end user friendly.