UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

COS 216 Practical Assignment 2

- Date Issued: **13 March 2023**
- Date Due: **27 March 2023** before **08:00**
- Submission Procedure: **Upload to the web server (`wheatley`) + clickUP**
- This assignment consists of **5 tasks** for a total of **95 marks**.

1 Introduction

During this practical you will be creating a site to view and compare different cars and brands. The idea is to give users of the site the ability to look at different cars and see the specs helping them make the right choice of which car to buy. Users of the site can choose to view car models, view car brands, compare cars and even use the find me a car feature.

After successful completion of this assignment you should be able to create a web page which complies to the HTML5 and JavaScript Standards. The specific web page for this assignment will showcase the following functionality:

- Javascript animations
- Loading Screen
- Retrieving data from APIs
- Populating your templates with the retrieved API data
- Implementing the Find me a car page
- Implementing the Compare page

2 Constraints

1. You must complete this assignment individually.
2. You may ask the Teaching Assistants for help but they will not be able to give you the solutions.
3. You must produce all of the source files yourself; you may not use any tool to generate source files or fragments thereof automatically.
4. Your assignment will be viewed using Brave Web Browser (`https://brave.com/`) so be sure to test your assignment in this browser. Nevertheless, you should take care to follow published standards and make sure that your assignment works in as many browsers as possible.
5. You may utilise any text editor or IDE, upon an OS of your choice, again, as long as you do not make use of any tools to generate your assignment. (**This includes ChatGPT!!**)
6. All written code should contain comments including your name, surname and student number at the top of each file.
7. Your assignment must work on the `wheatley` web server, as you will be marked off there. However you are free and encouraged to **Develop** using a local tool like XAMP and later move over to wheatley however it must work off wheatley during marking.

8. **You must only use JavaScript for this assignment (no libraries with the exception of the point below)**
9. You may use JQuery **ONLY for DOM manipulation**. Use of JQuery for AJAX calls will result in no marks for that section.

3 Submission Instructions

You are required to upload all your source files (e.g. HTML5 documents, any images, etc.) to the web server (`wheatley`) and clickUP in a compressed (zip) archive. Make sure that you test your submission to the web server thoroughly. All the menu items, links, buttons, *etc.* must work and all your images must load. Make sure that your practical assignment works on the web server before the deadline. No late submissions will be accepted, so make sure you upload in good time. The server will not be accepting any uploads and updates to files from the stipulated deadline time until the end of the marking week (Thursday at 3pm).

The deadline is on Sunday but we will allow you to upload until Monday 8am. After this NO more submissions will be accepted. FTP to wheatley will also be disabled during this time

Note, `wheatley` is currently available from anywhere. But do not rely that outside access from the UP network will always work as intended. You must therefore make sure that you `ftp` your assignment to the web server. Also make sure that you do this in good time. A snapshot of the web server will be taken just after the submission was due and only files in the snapshot will be marked.

Practicals are marked by demonstrating your practical to a tutor during the allotted marking weeks. **If you do not demonstrate your practical you will receive 0 for the practical**. You will only be marked in the practical session that you have booked on the cs portal, if you miss it, you will receive 0 (Unless special permissions have been granted by the lecturer. i.e. You were sick and able to provide a sick note).

4 Online resources

HTML5 - `https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5`
`http://www.w3schools.com/`

CSS - `http://www.w3.org/Style/Examples/011/firstcss`

Javascript - `https://www.w3schools.com/js/`
`https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics`

REST - `https://en.wikipedia.org/wiki/Representational_state_transfer`

JSON - `https://www.w3schools.com/js/js_json_syntax.asp`

HTTP Methods - `https://www.w3schools.com/tags/ref_httpmethods.asp`

AJAX - `https://www.w3schools.com/js/js_ajax_intro.asp`

5 Rubric for marking

API Functionality	
AJAX	12
JSON manipulation	4
Cars/Brands have the correct image pulled from the API	4
Cars page	
Data population	5
Searchable	5
Filterable	5
Sortable	2
DOM manipulation	5
Loading Screen	3
Brands Page	
Data population	5
Loading Screen	3
DOM manipulation	2
Find me a car Page	
Correct results displayed	5
Correct use of API Parameters	5
DOM Manipulation	5
Compare page	
Car Selection	10
Compare stats shown	5
DOM Manipulation	5
Validation	
HTML	3
JS	2
Upload	
Does not work on `wheatley`	**-20**
Not uploaded to clickUP	**-95**
Not Demoed	**-95**
Bonus	10
Total	**95**

6 Assignment Instructions

Task 1: Populate from APIs .... (55 marks)

This task requires you to pull information from a **Wheatley API** and populate all the templates (HTML layouts) you have created in Assignment 1 for the Cars and Brands page. You should retrieve data from the Wheatley API for at least 20 cars and 20 brands.

All API calls must be done through JavaScript **AJAX XMLHttpRequest** (`https://www.w3schools.com/js/js_ajax_intro.asp`). You must choose between synchronous and asynchronous calls and provide reasons for your choice.
NB: Your reasoning should be included at the top of your JS file/s as a comment.

For this practical you will use an API Hosted off Wheatley. You must use this API and no others. The documentation and usage of said API can be found at `https://wheatley.cs.up.ac.za/api/doc.html`

The following functionality needs to be implemented:

- You need to **replace your mock data** from Practical 1 with data from the API that will load dynamically when the user views the page.
- You must find a way to make sure all cars and brands have the correct corresponding picture.
- For all dynamic content you need to have a **loading screen/animation** to show that the data is being retrieved/processed.This can be a GIF/APNG/SVG loader for each car/brand or a loading screen for the entire page (it will disappear once all the data on that page has loaded), Decide which is best. You can get resources from (`https://loading.io/`) however you are not limited to that resource.
- The following should also be completed on the cars page
  - Search from the previous practical must be implemented using the API
  - Sort from the previous practical must be implemented using the API
  - Filter from the previous practical must be implemented using the API
  - You should also implement a way to indicate to the user if no results were found from a search and/or filter.

Task 2: Create the "Find Me A Car" Page .... (15 marks)

The find me a car page from the previous practical should be implemented. You might need to alter some of the questions asked to make sure it works with the API. You are encouraged to go back to the previous practical specification and remind yourself about the functionality of the page.

Since the page should be implemented, remove mock data (if present) and check the page works correctly. In the previous practical it was mentioned that you should have a way to show the results of the search (Min 5, Max 8) however from this practical the minimum constraint will be removed as there is a possibility that no car in the API will match what the user is looking for.

You should also implement a way to indicate to the user if no results were found.

Task 3: Create the "Compare Page" .... (20 marks)

The Compare page from the previous practical should be implemented. The main difficulty of this page will be a way of selecting the cars, hence most of the marks will be allocated to the selection of the cars. Just like with GSMArena you should be able to compare any 2 cars in the API against each other.

The way of selecting the cars may link to the main page(i.e. On the Cars page you can select which cars will be compared, an "Add car to compare" type feature). However this is only a suggestion not a requirement.

Task 4: Validation .... (5 marks)

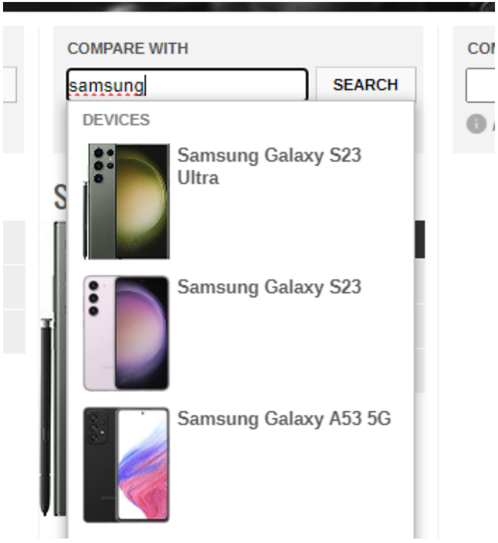
All HTML and JS should be valid and comply with the standards.

Task 5: Bonus .... (10 marks)

Bonus marks will be given for additional functionality and awarded based on the difficulty level. These marks will only be awarded for notable additions (simply adding a background image is not enough).

The following, among others, may earn you extra marks:

- The ability to export the results of the cars page and export to PDF. The PDF would need to look different you cannot just export a PDF of the page. For this the PDF Library is allowed such as JsPDF.
- The ability to compare more then 2 cars.
- A more advanced way to select cars to compare. i.e. a small window with search results with the **pictures** of the car and name, see example image below.


COMPARE WITH
samsung
SEARCH
DEVICES
Samsung Galaxy S23 Ultra
Samsung Galaxy S23
Samsung Galaxy A53 5G