%%

Presentation of deliverables (15 minutes) <- you talk at them for 15 mins **F Panteli**

%%

You'll present your work and explain your technical decisions:

- **part 1: screen share and open the project in a browser *THE PROJECT IN THE BROWSER***
  - **Present an overview of the project context (what has been produced, why and for whom). <- this is the context, the 'why' and 'for whom'**
    - -> we are doing a work placement as a web developer at a startup called Booki
    - -> Sarah is the CTO/ assessor and Lenny is the UI designer
    - -> we are given a mockup from the UI designer Lenny, a brief, basic starting html on git, and courses on the OpenClassrooms platform
    - -> the task was to convert the mockup of a webpage into a webpage using html and css
      - to make the website from the mockup using html, css
    - -> for whom - the webpage is for customers booking holidays / looking for things to do in cities of their choice
  - **Present the visual aspect of the project / demonstration of how it works <- this is 'what has been produced'**
    - -> breakpoints (for desktop and large desktop versions) - resize the screen to show the breakpoints - centering of the content, green below mobile breakpoint because only asked to do desktop and large desktop versions
      - the green background colour when the mobile breakpoint region has been entered
      - because indicates the screen size below which the brief has asked us to consider
    - -> the links in the nav bar - there is a navy top border in the hover state for those elements of the nav bar
    - -> the search bar
      - text can be entered into the search input form
      - the format / styling of that text -> it's called the placeholder text
    - -> the hyperlinks around several of the cards
      - you could inspect it in the console (or not)
    - -> filters - the pseudo class hover state over them and the svg icons which were exported from Figma
    - -> the mouse effect class in css - so when you hover over text, the mouse doesn't change into a cursor icon
    - -> the main index.html and the html sandbox files - changes were made to the project brief and these are those changes which I made in this sandbox
- **part 2: screen share and open the html / css in VSCode *THE CODE FOR THE PROJECT***
  - **Present the project code in your chosen code editor (for example, Visual Studio Code). Feel free to highlight the more complex aspects you had to produce, reflect on the project and your deliverables.**
    - **problem solving thought process**
      - -> breaking down the webpage into different sections and treating each of these as a problem to be solved <- what we want, what we know, bridges, choosing the best one, analysing the results
        - **what we want <- THE STEPS**
          - -> the problem was how do we get this portion of the mockup coded into a section of the webpage using html and css
        - **what we know <- IT'S HOW THE HTML COMMENTS ARE STRUCTURED**
          - -> **first consuming the course content**
            - there were three new courses, covering css and mockups
              - Figma
              - CSS

        - breaking down and integrating mockups
    - **inspecting what we want from the mockup**
        - -> the flex(box) direction
        - -> the atoms / repeated parts of the webpage (the cards in this case)
    - **the approach we used**
        - -> this is how the html is structured - with the problem solving thought process done by each section - the approach we found worked
        - -> we did a snowball approach - where we treated each section of the webpage like a problem to be solved, did the next easiest problem based off the one we were doing, and then built upwards

2

- **then pointing out the individual challenges**
    - **-> <footer> semantic html tags**
        - reading through the brief at the end
        - splitting the entire thing into one giant problem
        - -> the footer elements are in thirds / elements in blocks
            - we initially made this out of divs, rather than semantic <footer> html tags
            - the footer being made out of three parts which have the same widths
    - **-> the clickable links for the nav bar**
        - -> for the nav bar at the top the h3 elements have default margins in css which were reset
        - -> to make the links clickable
    - **-> figuring out how to insert the svgs / icons from Figma**
        - -> exporting the html for them from Figma
        - -> inspecting them in the console
        - -> the margins on the stars were another one - solved by putting the stars in the sandbox at the bottom of the index.html file and editing them with classes
    - **-> the widths of the filter buttons**
        - -> this was a problem with the svgs
        - -> this was solved by id'ing each of the filters and defining their widths as percentages
    - **-> the html sandbox**
        - -> search input form
            - -> the search input form <form> semantic html tags, which contain the search <input> parameter / attribute
        - -> the semantics below
            - -> links around the cards
        - -> one challenge is that they changed the mockup halfway through the project completion, -> bold on the cards
            - in case changes are made to the project mockup
        - -> so we made the changes in a new html file, html sandbox -> which used different css classes
        - -> the new sandbox html file linked to the same css file, using atoms of the webpage in that sandbox file

3

- **things you did after writing the code**
    - **removing repeated classes in the css**
        - control eff'ing different css classes and seeing if they were being used in the html or not
        - deleting the css classes which weren't being used
    - **checked the page resizing**
        - -> the mockup showed widths defined in pixels, but these were converted into percentages for use in css, so that these elements could withstand

page resizing

- heights in pixels, widths in percentages
- testing the appearance of the webpage -> page resizing

  - **validate it with w3c (css and html)**
    - -> from validating the code - w3c, with html, css -> 'aria-label' for the search bar svg, alt attributes (accessibility)
      - -> the same as alt tags for images but for svgs
      - -> the spaces in the relative file paths are %20's
- know the argument for css specificity
  - if you have one element which is being targeted by multiple classes, then the one with the highest priority takes precedence
  - so for example with the filters, if one filter was being targeted by an id and the filters class, then the css from the id would take precedence
  - this was important because the widths of the filters were given by id'ing them, rather than css from the .filters class which was overarching

- **don't mention**
  - -> kebab case in css
  - -> icons - these were done with svgs exported from Figma
  - -> not duplicating the css classes
  - -> testing the appearance of the webpage in Chrome / Firefox (two different browsers)
  - -> first breaking down the mockup
    - all the elements in a hierarchy
    - flexbox
  - -> hover pseudoclasses in css over the filters section - .filter:hover {
  - }
  - -> css specificity (they will probably ask us about this)
  - -> breakpoints <- do this in the first part to show the page expansion etc -> but don't divulge into the css for it too much unless they ask you
    - -> css media queries
    - -> there is a 1440px breakpoint, a 1024px minimum <- desktops and large desktops, not mobile
    - -> testing it with different page expansions to see if there is consistency
    - -> centring the content

%%
Discussion (10 minutes) <- you talk with them for 10 mins - way less structure just get the points right
%%
Your assessor, still in the role of Sarah, will ask questions about your HTML and CSS code. They might question you on the following points:

- **The W3C validation of your code**
  - -> one of the points which was brought up with this was not having a link <a> semantic html tag around a button
  - -> there were points like missing divs which were brought to attention with this
  - -> there is a W3C validator extension in VSCode which was used whenever minor changes to the code were made (and so the web page did not break)
- **The use of semantic HTML tags**
  - -> the footer tab was inserted using divs initially
  - -> this was later changed to semantic <footer></footer> html tags, after reading through the project brief when checking the code - which changed nothing major to the appearance of the webpage

    - -> it was initially planned to create the nav bar with divs, but it was mentioned in an html skills session that the <nav> semantic html tag should be used rather than a div for this - and so the project uses this
    - -> it was also noted in the project brief that the search bar should be surrounded by a <form> semantic html tag, rather than three divs flexboxed together within a larger div
- **Why it's important to run your code through validators**
    - -> the code validator notified me of multiple divs in the html which had not been closed
    - -> validators enable developers to identify errors in the code to adhere to best practices as prevention before these errors (if left unaddressed) build and the entire webpage breaks (at which point they are harder to fix entirely)
    - -> validating my html also provided error messages regarding my use of alt tags on images, which allowed me to make it more accessible once rectified. This also provides learning opportunities to developers by providing them feedback on their code
- **The code editor setup**
    - -> I used two monitors
    - -> the first had the IDE (in this case VSCode) with the css and html files for the project open
    - -> the second monitor had the main index.html file for the project open in the Chrome and Firefox browsers using the live server VSCode extension
        - edit the code in the first monitor and the code in the second monitor updates in the browser
    - -> I had the index.html file open in two different browsers using the live server extension because the project brief specified webpage compatibility with these two browsers under its requirements section
    - -> the second monitor I used this on was for a large desktop, as the project mockup in Figma required that the webpage work above the desktop breakpoint (for desktops and large desktops)
- **CSS specificity (most likely to be asked question)**
    - **what css specificity is**
        - **-> which specific style will be applied to that element in css, when there are multiple of them targeting it**
        - **-> if one element is targeted by more than one piece of css, then the css which actually styles it will be the css which is higher on the hierarchy -> depending where on the hierarchy it is**
    - **one example of this in our code is -> the filters**
        - **-> ids (targeting one element) take precedent over classes (multiple elements)**
        - **-> there is a .filter class in css for all of the filters, and then there are ids for each individual filter**
        - **-> those ids have been defined to make it easier to target the width of the specific filter elements using css**
        - **-> if the widths were set for the filter elements using the filter class, this would have no effect, because they have also been set by the ids for those individual filters, which have a higher specificity / precedence in css**
- **Mockup analysis and breakdown**
    - -> the course said to use a top down approach
        - To first break down the mockup into a structured html file, by looking at its different sections and annotating them with the relevant semantic html tags on an iPad
    - -> looking at the atoms / reusable parts of the mockup <- in this case it was with the different cards
    - -> looking at which parts would be created from a flexbox in a row / column flex direction <- inspecting the flex direction of the different elements on the page
        - e.g the 'things to do in Paris' cards were flexboxed into a column, and with the 'Paris

Accommodation' cards this was done in a row
    - -> snowball approach
        - once I had the mockup structured into different sections in html (using a top down approach), I completed each of the different sections in html by choosing the next easiest problem and building upwards
- **Why it's important to separate the HTML and the CSS**
    - -> separating the content from the styling means changes can be made to one without changing the other
    - -> reusability - separating out the css (styling) from the html (content) allows multiple html pages to link to the same css file, allowing for reusable code
        - the css doesn't have to be repeated across the different html files
        - it only has to be changed once, and all of the html elements which use those classes (across multiple webpages) can change
        - scalability -> designers can work on the css and developers can work on the html

%%%
To prepare, you need to have
%%%
- setup
    - -> the screen share software ready
    - -> the project open in a browser window in one tab
    - -> VSCode open in another -> the index.html and css files of the project open (with the html, ready for talking through the problem solving thought processes we've annotated it with in comments)
    - -> open this note in the second monitor and present the first monitor, split screen it with a timer <- you need to time it when you do it or it might not pass
- gone through the questions they might ask you (notes above) in the discussion session afterwards
- uploaded the code to the OpenClassrooms platform 48 hours before the assessment