Presentation of deliverables (15 minutes)
%%

In the first part, you'll explain your technical decisions regarding the site design.
You'll cover:

%%%
How you created the heart animation (4 minutes)

- ***-> importing the icons***
  - in the header of the index.html file
    - using the Font Awesome kit link -> not the cloud fare link - issues with the css validator otherwise
  - they were imported in one above the other -> the solid one had to be below the transparent one or it didn't work
  - -> a div was placed around both of them
- ***-> the scss file for the heart animation <- it's own scss file***
  - -> the transparent heart was positioned absolutely at the bottom of the heart scss file so they lay on top of each other
  - -> the opaque heart was styled at the top of the file
  - -> then its opacity was transitioned in its hover state -> to go from not being hovered over to being hovered over and vice versa
  - -> transitions were used and not keyframes for this because the animation was triggered by a pseudostate of the elements
    - -> it could have been done using keyframes or transitions (these methods are detailed in our notes for the project in its GitHub repository, these were the two methods which were presented) -> but keyframes were selected
  - -> the two icons were placed on top of each other using absolute positioning (not relative to the page) and the opacity of the opaque one was transitioned in its hover state
  - -> the opaque heart was set on the bottom / left edge of the box

%%
How you built the dish selector animation (4 minutes) <- the pages scss file control f for "choice__card__check"

- ***-> transition not keyframes***
  - -> it was triggered by the hover state of the entire choice card
- ***-> in the html***
  - -> it's a font awesome icon which is surrounded by a div
  - -> that div is targeted with the class called -> "choice__card__check"
- ***-> in the scss***
  - -> this was for the menu / restaurants pages -> so it's in the pages scss file
  - ***-> the styling of the tick in its regular state***
    - the icon is in a div and it's being flexed into the centre of that div
    - the colour of that div is also being formatted by that class -> using a variable name (imported at the top of the scss file)
  - ***-> transitioning the green div***
    - -> the right margin of the green div is set to be equal to negative its width -> so it is being moved off of the RHS of the choice card (containing div) and can't be seen
    - -> in the hover state of that entire choice card div -> this is when that margin is being transitioned to 0
      - -> so the green tick now fits back onto the card
      - -> then the same thing is applying in reverse (going from hovered to not

hovered over and vice versa)

%%

- ***Your chosen solution for the home page loader (4 minutes) <- '_loading_animations.scss' scss file***
  - ***-> the styling of that background element <- "loader" class***
    - -> graphic identity -> the colours match the graphic identity in the creative brief
    - -> keyframes not transitions -> the trigger is the page reload, not a user interaction
    - -> a div inside a div
  - ***-> a) the background container in the larger div <- the "loader" class***
    - ***to define the styles of the background***
      - -> there is a larger div which is surrounded by a smaller div
      - -> that larger div is for the background
      - -> the top styles in that scss file are to format its background
    - ***then to transition that background to be behind the content on the page***
      - a keyframe called goAway is used to change the opacity of that background
      - the z index of that element is also transitioned from 1 to -1 (it's going from in front of to behind the content on the index.html file - project homepage)
  - ***-> b) the text on the background <- the smaller div styled by the "word" class***
    - -> the smaller div is for the text (word)
    - -> for the .word class -> the style for that text is first defined
    - -> and then a transition is applied -> which expands its scale
    - -> this was chosen so that it is big enough to fit on the page -> but not small so the text couldn't be seen
    - -> the background was transitioned to be behind the content of the page -> not the text -> because the text was embedded within the background which was being transitioned
  - ***-> there are two keyframes***
    - -> the first is to transition the background to be behind the content for the page and the second is to make the text which is embedded in it smaller

%%

- ***How you used the mobile-first approach for the project code (4 minutes)***
  - ***read through resources***
    - -> we first read through the resource in the milestone guide (following the milestone guide)
      - -> our notes on this are in the project explainers file
    - -> creating notes on them all

  - ***built the homepage with a fixed width (mobile breakpoint)***
    - -> defined a box of fixed (mobile width) -> 375px for the mobile version
      - -> "menu-page-container" <- the class around the entire body
    - -> built the project homepage

  - ***added in media queries for elements so it worked for desktop and tablet versions***
    - -> stacking the boxes on the project homepage and in the footer
    - -> then added in the media queries for each of the sections in the sass
    - -> these were added wherever those specific sections of code were used
    - -> the tablet breakpoint was used for the media queries -> at which point the different sections of the webpage were targeted with those queries

  - ***-> there is a media query log in the project explainers -> as to where in the Sass each of these can be found***

- ***various changes were made using these media queries***
  - the footer elements get justified onto a row
  - the purple circle cards are justified into a column
  - flex grid for the four elements on the project homepage
  - the margins around the beige sections in the restaurant (menu pages are gone)
- ***this process was repeated across the different project pages -> 5 of them***
  - -> Kitchen West was used as the template -> there are thought process / notes around it
  - -> the other one is the index.html file for the project

Discussion (10 minutes)

%%

Your assessor, still in the role of Paul, will ask you some questions. They might question you on the following points:

%%

- ***How you created the animations***
  - ***-> the loading animation***
    - the "loader" class
      - -> the class which targets and formats the background
      - -> everything stays in the centre of the page
      - -> is in front of the page
      - -> and then is animated to go behind the page (keyframes)
    - the "word" class
      - -> the class which targets and formats the text
      - -> this class is embedded inside the loader class -> which moves the elements
      - -> keyframes to make the font size bigger
    - keyframes for it
      - -> the entire thing was done using keyframes (it wasn't triggered e.g by a hover state)
      - -> two keyframes
        - -> one which makes the background go away
        - -> one which makes the text expand
      - -> these are defined in the keyframes partial, and then imported in

  - ***-> the button animation***
    - -> it was done with a transition not keyframe
    - -> when it's being hovered over and when it's not being hovered over (two of them)
    - -> two changes
      - 15% drop in opacity
      - background shadow -> to match Figma mockup
    - -> there were two classes -> one centres the content and one makes the button
      - -> order__button

  - ***-> the heart animation***
    - -> transition not keyframe (triggered by a hover state)
    - -> there are two Font Awesome icons -> one for the full heart and one for the empty heart
    - -> they are positioned on top of each other using absolute positioning
    - -> you have the css (sass) for the full one, then the sass for the transparent one

        - -> the background of the full one is first set -> then it's opacity is transitioned
        - -> the transparent one is a lot simpler
        - -> the entire thing is set in the heart animation scss file

- ***-> the choice card blocks fading in one by one / the entire menu pages fading in***
    - -> this was done with a keyframe (it wasn't triggered e.g by being hovered over)
    - -> the same keyframe was used for
        - fading all of the menu pages
        - fading the individual sections of the menu pages
        - -> it was set / used four times in the scss -> to apply it to each of those individual sections
            - -> three (starter/ mains/ desserts)
            - -> and then once for the entire page
        - -> and then it was also defined in the scss file -> for the keyframes
    - -> in other words
        - -> we are using a keyframe and not a transition -> because there is no hover state triggering it
        - -> when we are defining the classes which we want to do those transitions in the scss
            - -> there are four of them, which we want to transition their elements
            - -> so we define one keyframe and then use it across the definitions of those classes / selectors -> we just use them in different ways
            - -> so we have four classes -> one for the entire page, and then one for the starters / mains / desserts
            - -> then we apply those classes across the different html files of the site -> and they control the transitions for those elements

- ***-> the green ticks***
    - -> transition not keyframe (triggered by a hover state)
    - ***-> this is a Font Awesome icon which is surrounded by a div***
        - -> those icons having to be imported in the header of the html files
        - -> not using the cloudfare link -> instead using one which was generated from the Font Awesome library using a kit
    - ***-> this div was targeted by the class called choice__card__check***
        - -> it's in pages.scss <- because it's for the restaurant / menu pages
        - -> and it's not a keyframe -> so it's not being done in the keyframes.scss file
            - -> either keyframes or transitions were used to create animations -> where keyframes were used they were put into the keyframes scss file and where transitions were used they were defined in the scss files where they were used
        - ***-> there is the sass which defines the appearance of it***
            - -> it's a box and the icon has been flexed into the centre
            - -> then which moves it out of the larger div which it's in -> with the negative margins -> the negative RHS margin equals the width of that container
        - ***-> then for the hover state***
            - -> we are transitioning the margin to be 0 -> in other words -> in the hover state of the entire choice card, we are transitioning the margin of the green tick to be 0
            - -> so it's being moved into the larger div which contains it
            - -> and we are transitioning that margin
            - -> and doing it in the pages scss file
                - -> because it's not a keyframe (it's being triggered in a hover state)
                - -> and it's for the menu pages (not the index.html file homepage for the

project)

%%

- ***How you chose to structure your CSS code***
  - -> according to the 7-1 file structure in the animations course -> our notes for this are in the GitHub repository for the project
  - -> importing them into the main scss file
  - -> because of the specificity -> they were done in a specific order
  - -> some of the 7 were used and some weren't <- one of the challenges was knowing which of them to include and which not to include
    - -> so when the compiled css was being divided into scss partials -> only the ones which were necessary were used
    - -> as opposed to starting off with all seven of those partials in the 7-1 file structure and then having ones which weren't used
  - -> the file tree is annotated in the project explainers on the git repo

%%

- ***How you used Sass <- the backwards approach -> writing the compiled css first and then migrating it all over to sass***
  - ***css***
    - -> the css was written into the compiled css (styles.css file)
    - -> and then this was divided into sections which compiled into that css and moved into the sass partials
  - ***css to sass partials***
    - -> then each of the partials was made into Sass with the Sass beautifier (the link to it is in the project readme file)
    - linking and compiling
      - -> those partials were set up to be imported into the main scss file
      - -> and then that's being compiled into the styles.css file for the project -> sass main.scss styles.css
        - -> you can do a live demo (we have the project backed up)

%%

- ***How you approached the responsive aspect of the site***
  - ***read through resources***
    - -> we first read through the resource in the milestone guide
      - -> our notes on this are in the project explainers file

  - ***built the homepage page with a fixed width (mobile breakpoint)***
    - -> defined a box of fixed (mobile width) -> 375px for the mobile version
    - -> built the project homepage

  - ***added in media queries for elements so it worked for desktop and tablet versions***
    - -> stacking the boxes on the project homepage and in the footer
    - -> then added in the media queries for each of the sections in the sass
    - -> these were added wherever those specific sections of code were used
    - -> the tablet breakpoint was used for the media queries -> at which point the different sections of the webpage were targeted with those queries

  - ***-> there is a media query log in the project explainers -> as to where in the Sass each of these can be found***
    - ***various changes were made using these media queries***

        - the footer elements get justified onto a row
        - the purple circle cards are justified into a column
        - flex grid for the four elements on the project homepage
        - the margins around the beige sections in the restaurant (menu pages are gone)
    - ***this process was repeated across the different project pages -> 5 of them***
        - -> Kitchen West was used as the template -> there are thought process / notes around it
        - -> the other one is the index.html file for the project

Debrief (5 minutes)
%%

At the end of the assessment session, the assessor will stop playing the role of Paul so that you can debrief together.

- ***-> context***
    - -> the context for the assessment was that we were a junior web developer for the ohmyfood company
    - -> it was a startup which was the equivalent of a deliveroo company, but for gourmet food
    - -> there are four menus on the page -> to start off with
    - -> we coded the website, which the UX designer made -> and which Paul the company CTO agreed to
    - -> this is us presenting the website we made to Paul -> and how various parts across it work
    - -> so the assessor is playing Paul in the presentation (it's a role play) and we are playing the web developer
- ***-> the entire assessment is half an hour long***
    - we start with a **15 minute presentation, which covers four bullet points - at 4 minutes a bullet point -**> which you time with the second monitor
    - -> and then there's a 10 minute discussion
        - -> so in terms of timing - the only thing you need to worry about is covering the bullet points at the start
        - -> and when the discussion happens -> you are still role-playing with the assessor you're talking to (playing the role of Paul the CTO at the company)