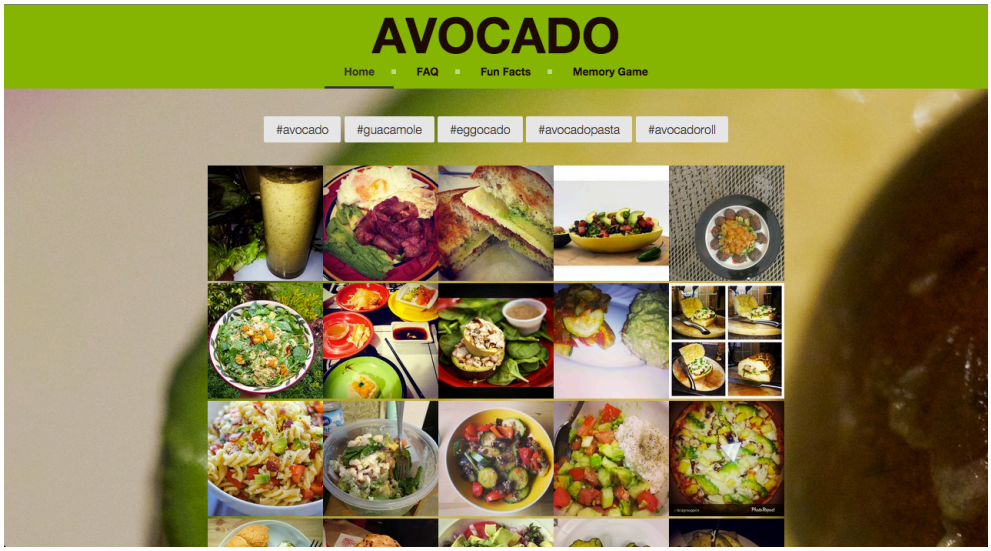
AVOCADO

Developers: Swathi Anand (slanand) & Jenny Yang (hyeayouy)

Web Application URL: http://avocado-hyeayouy.rhcloud.com/

Application Descriptions:

1. **Load a page (html, JavaScript, and optionally CSS) from your server and display something.**
    - From our application homepage, you can see that it loads Instagram photos with the 'avocado' hashtag and displays the photos in the form of a grid.



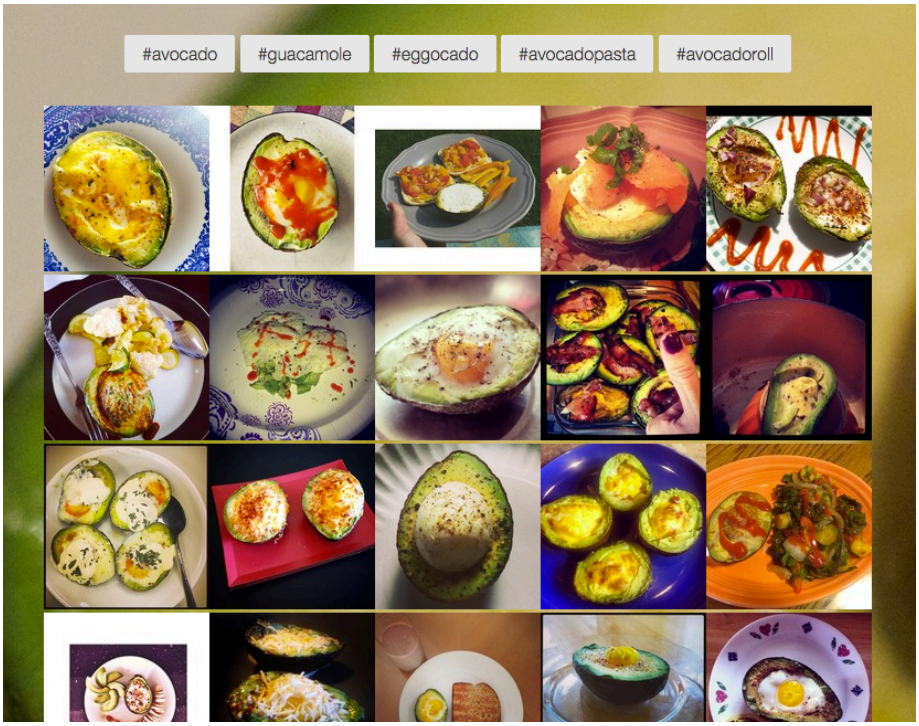
2. **Make an Ajax request to your server and update the display with new content received.**
    - For our 'FAQ' and 'Fun Facts' page, we created two JSON files that have the content for those pages. Upon clicking either of those tabs on the menu bar, it retrieves the information from those respective JSON files and displays them on the webpage.
    - The image below is of the FAQ homepage, followed by a screenshot of code from our 'application.js' file making the request to the FAQ JSON file.


```
81
82  // Avocado FAQ
83  $(function() {    // do once original document loaded and ready
84    $.getJSON("avocadoFAQ.json", function(response, diditwork) {
85        console.log(diditwork);
86        var displayT = "";
87        for (var i = 0; i < response.questions.length; i ++) {
88                var entry = response.questions[i];
89                displayT += "<div id='topic'><h3>"
90                  + entry.question + "<\/h3><ul><li>"
91                  + entry.answer + "<\/li><\/ul>";
92                }
93    $("#faqArea").html(displayT);
94    });  // getJSON
95  }); // onReady
96
97  // Avocado Fun Facts
98  $(function() {    // do once original document loaded and ready
99    $.getJSON("avocadoFacts.json", function(response, diditwork) {
100       console.log(diditwork);
101       var displayT = "";
102       for (var i = 0; i < response.facts.length; i ++) {
103               var entry = response.facts[i];
104               displayT += "<div id='topic'><h3>"
105                 + entry.title + "<\/h3><ul><li>"
106                 + entry.fact + "<\/li><\/ul>";
107               }
108   $("#factsArea").html(displayT);
109   });  // getJSON
```

3. **Make an Ajax request to a 3rd party server using CORS or JSONP and update the screen with that content.**
    - Our website makes Ajax requests to the server whenever one of the hashtag buttons are pressed on the top of the homepage. This updates the display with Instagram photos for a particular hashtag; the default hashtag for the homepage is '#avocado'.
    - We used the Instagram API to make the Ajax request using JSONP to update the homepage with the captured photos. The photos are the first 20 photos

recently posted with the particular hashtag. Using the Instagram API does not require user authentication, only that you acquire a client ID.

- Instagram API - http://instagram.com/developer/
- The image below is of photos retrieved from Instagram when the '#eggocado' button is pressed, followed by a screenshot of code from our 'application.js' file.



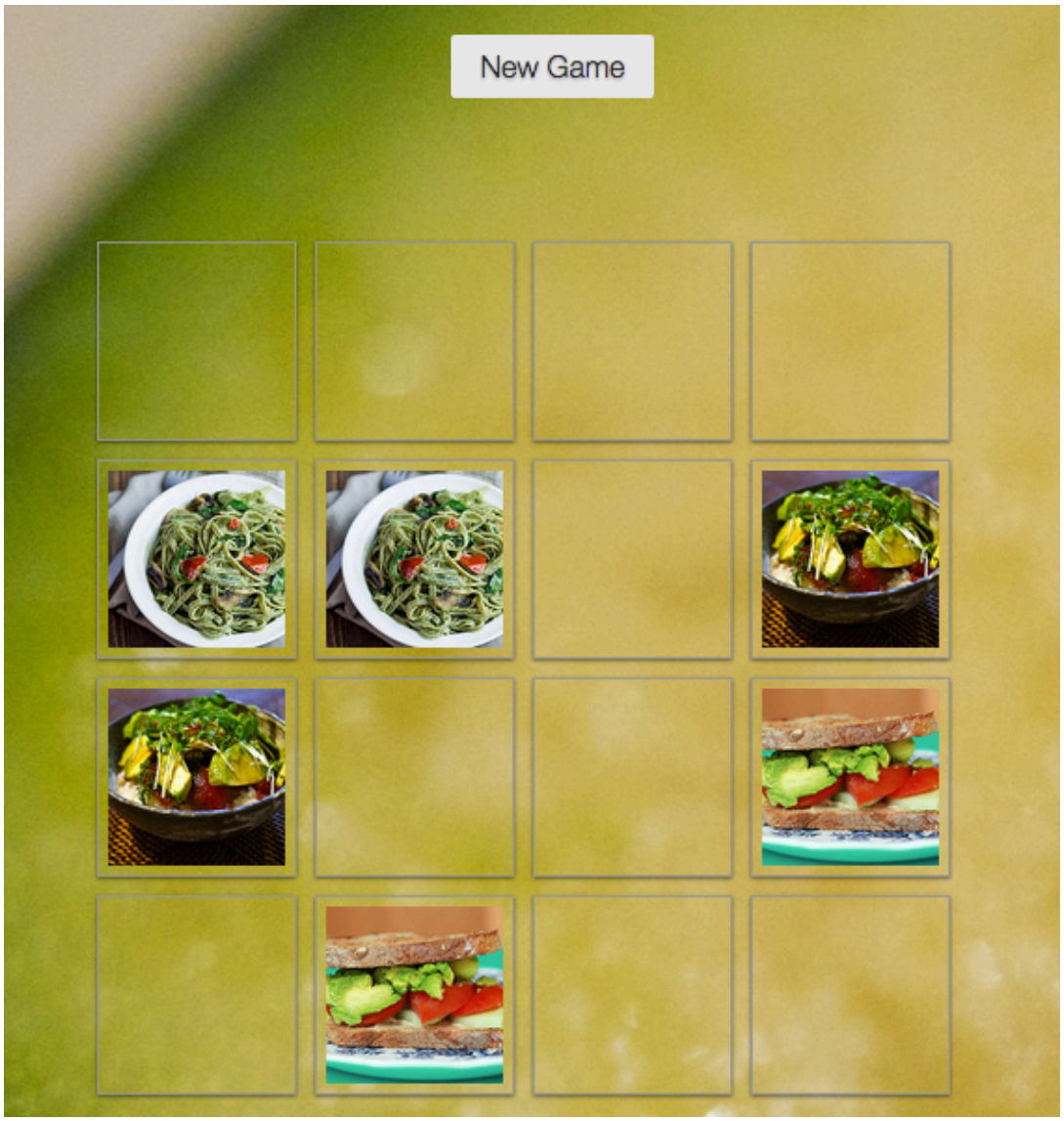

```
// call avocadoInsta() when document is ready
$(function() {
  avocadoInsta();
});

// index Instagram photo #buttons
function avocadoInsta() {
  try {
    $.ajax({
      type: "GET",
      url: "https://api.instagram.com/v1/tags/avocado/media/recent?client_id=f084959fc2c24ca38cd1083389
      jsonp: false,
      dataType: "jsonp",
      crossDomain: true
    });
    return false;
  } catch (error) {console.log(error.description);}
}

function guacInsta() {
  try {
    $.ajax({
      type: "GET",
      url: "https://api.instagram.com/v1/tags/guacamole/media/recent?client_id=f084959fc2c24ca38cd10833
      jsonp: false,
      dataType: "jsonp",
      crossDomain: true
    });
    return false;
  } catch (error) {console.log(error.description);}
}

function eggoInsta() {
  try {
    $.ajax({
      type: "GET",
      url: "https://api.instagram.com/v1/tags/eggocado/media/recent?client_id=f084959fc2c24ca38cd108338
      jsonp: false,
      dataType: "jsonp",
      crossDomain: true
    });
```

4. **Interaction: Have at least three points of user interaction. I.e. it should handle at least three GUI events.**
    - Our website definitely has at least three points of user interaction. Users can interact on the home page by clicking the various hashtag buttons to view the Instagram photos and navigating through the different pages on the nav bar. Additionally, the user can play an avocado foodie memory game and start a new game if the user wants to play another memory game.



5. **jQuery: Include at least 10 instances of using jQuery . This is arbitrary, but is meant to be indicative of a more sophisticated app.**
    - Our web application definitely has around 10 instances of using jQuery. Most of the jQuery is implemented in the memory game.
    - Below is a snippet of our 'memory.js' file.

```
shuffle();

function openCard() {

    id = $(this).attr("id");

    if ($("#"+id+" img").is(":hidden")) {
        $("#grid div").unbind("click", openCard);

        $("#"+id+" img").fadeIn('fast');

        if (imgopened == "") {
            boxopened = id;
            imgopened = $("#"+id+" img").attr("src");
            setTimeout(function() {
                $("#grid div").bind("click", openCard)
            }, 300);
        } else {
            currentopened = $("#"+id+" img").attr("src");
            if (imgopened != currentopened) {
                // close again
                setTimeout(function() {
                    $("#"+id+" img").fadeOut('fast');
                    $("#"+boxopened+" img").fadeOut('fast');
                    boxopened = "";
                    imgopened = "";
                }, 400);
            } else {
                // found
                $("#"+id+" img").addClass("opacity");
                $("#"+boxopened+" img").addClass("opacity");
                found++;
                boxopened = "";
                imgopened = "";
            }
```

6. **Have good separation of HTML, JavaScript, (and CSS if you use it)**
    - We have separated our HTML, Javascript and CSS files from each other. The CSS and Javascript for the memory game are separate from the rest of the websites' CSS and Javascript.
7. **Comment your JavaScript well**
    - We have commented our code in our 'application.js file' as well as in the 'memory.js' file.
8. **Bonus: +3 for significant displays of creativity or usefulness, +3 for excellent quality of visual design**
    - Jenny and I have tried to make our website look as 'start-upy' as possible! Our design decisions were definitely not random. We kept the avocado color scheme throughout the entire website, and we chose to keep a faded out background image of an avocado so that the content will display clearer against the background, as opposed to using a high-definition image. We're very proud of our website and we hope you like it too!!!

Citations

Avocado FAQs: http://www.avocadocentral.com/how-to/faqs

Avocado FAQs: http://www.drweil.com/drw/u/QAA401311/Avoid-Avocados.html

Avocado FAQs: http://www.thekitchn.com/how-to-know-when-your-avocado-is-perfectly-ripe-inside-172933

Avocado Fun Facts: http://www.webmd.com/diet/features/8-healthy-facts-about-avocados

Button CSS: http://purecss.io/buttons/

Instagram API: http://instagram.com/developer/

jQuery JSON request: http://api.jquery.com/jquery.getjson/

Memory game: http://www.blackieandkanuto.com/games/memory/index.html

Nav Bar: http://cssmenumaker.com/menu/simple-responsive-menu