API FOR THE WIN

EXERCISE 5

Created By :

Aashika Parekh

Claire O’Malley

Logan Carter

API INTEGRATION APPROACH

API CHOSEN

For our project, we had decided to implement a random name generator into the web experience. The API is called Radomuser.me. This API will generate random information for a user. The information that will be generated will consist of a first and last name, gender, and country. This will make users feel as if they are taking on the life of another individual.

RATIONALE

We chose Randomuser.me for the API because we wanted users in our museum experience to use the API to own their own badges for the experience. We had considered them inputting their own name and finding a generator that could come up with something similar, but realized we needed something completely seperate, as we want users to feel they are taking on the role of someone else.

HOW THE API IS USED

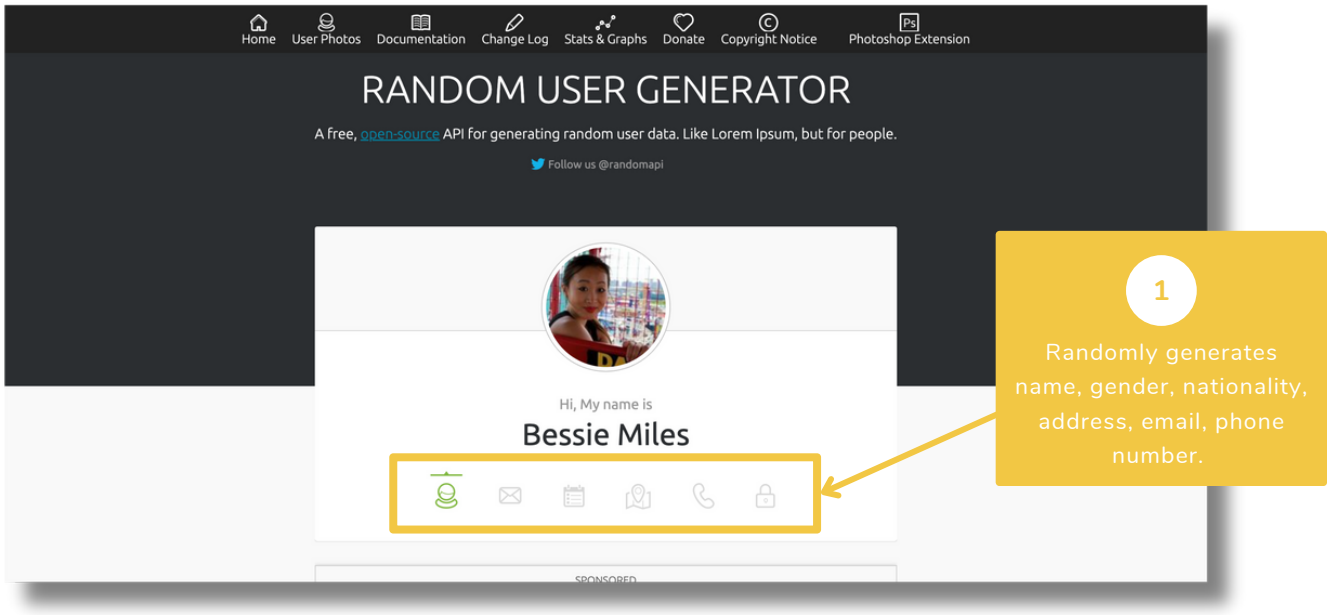
Background

- Before participants enter the theatre, they are guided to a screen that uses the API on the web experience site where they are prompted to use the random name generator to create their own “smuggler ID”
- After receiving their ID, participants will have their ID printed where they can use the attached barcode to redeem a drink later on in the experience
- When the film is over, participants are guided to the speakeasy where they will then use the ID card to redeem a drink and to take a picture in the photo booth
- After exiting the experience, the participants will be allowed to keep or recycle their smuggler ID

API INTEGRATION APPROACH

01 Explore APIs

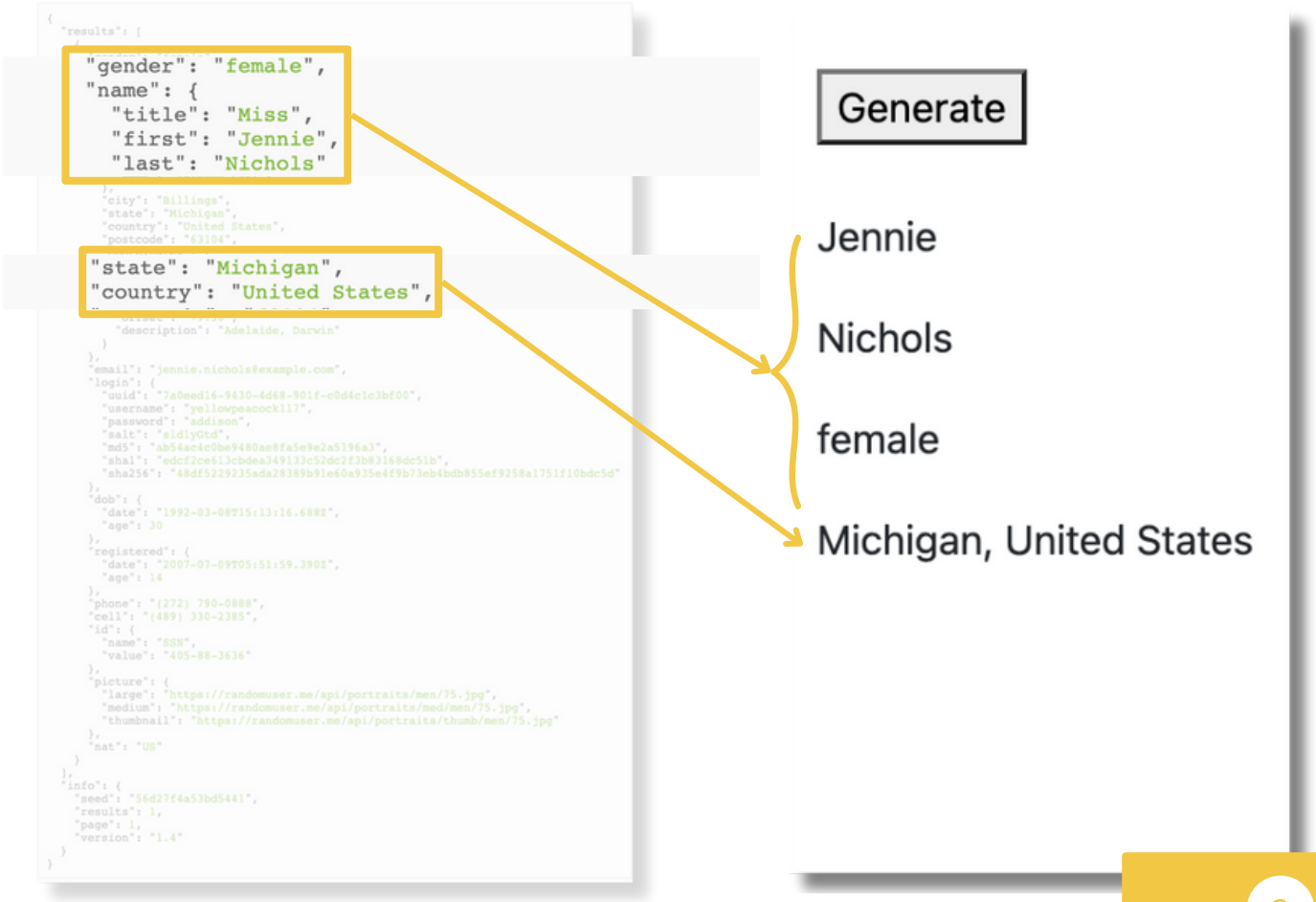
Setting aside sufficient time to thoroughly investigate an API's features and match them with our project idea was essential. Our goal was to ensure that the API integrated with the whole experience, rather than standing out as a separate add-on, by seamlessly integrating it into the museum exhibit from Project 3. We thought about it and decided to use the Random User API to provide our users with fake smuggling IDs. This calculated decision would enable us to effectively keep an eye on everyone's laser weapons throughout the 4D theater game and simultaneously track the drinks being served in the speakeasy, which added to the overall coherence and immersion of the visitor experience.



API INTEGRATION APPROACH

02 API Response

We first concentrated on getting data from the API for printing before working on web dev. Due to this, the three files (HTML, CSS, and JavaScript) have to be first structured and then connected. Although the majority of API calls follow a similar basic syntax, this process is varied a little when calling specific data. The basic code consisted of initiating API calls to obtain the complete JSON file in the console, which was simple. The slight challenge in this overall step was properly parsing the data and printing it on the website.



2

Parsing and printing data from the JSON file

API INTEGRATION APPROACH

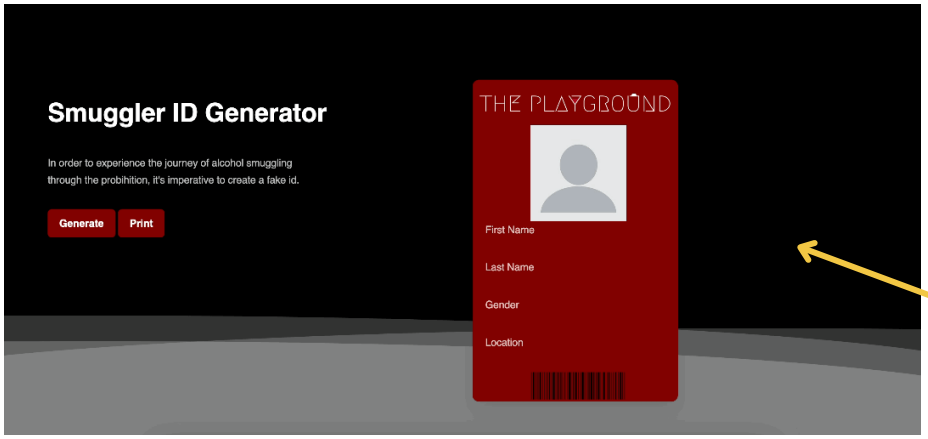
03 HTML Formatting

After the API data was successfully integrated into the website, we turned our attention to organizing the content. We started this phase by introducing the header "Smuggler ID Generator" and giving a brief explanation of the goal of the website. Furthermore, we positioned two buttons under the subheader in a purposeful manner so that people could create and print a new fake ID. We then formatted the HTML content to ensure the best possible user experience, using Bootstrap to improve organization and visual consistency.

API INTEGRATION APPROACH

04 CSS Incorporation

Although the website has functional features at this point, it is missing visual design components. First, we worked on the careful placement of elements in the area, which included a lot of work with the CSS selector '@media screen.' This selector was heavily utilized to guarantee both responsiveness and ideal element scaling across a range of screen sizes. Considering the website's two main purposes, we made the move to add a visually captivating background to draw visitors in. Our idea was to create gradient waves that would periodically and rhythmically flow across the screen, adding to the interface's overall visual appeal. This purposeful design decision complements the website's twin functions by striving to produce an engaging and visually beautiful user experience.



4

Rhythmic gradient waves and new spacing (GIF)

API INTEGRATION APPROACH

05 API Integration

The last stage of this project is to replace the fake ID placeholders with real API data. This was done by seamlessly integrating the JavaScript IDs into the HTML <p> tags.

Javascript File

```
$(document).ready(function(){

    var id = "AIzaSyBDcxzunlBO9_OLhVooAVeoCKtF1xWXSgQ";

    $("#generate").click(function(){


        $.ajax({
          url: 'https://randomuser.me/api/?password=upper,lower,1-16',
          dataType: 'json',
          success: function(data) {
            console.log(data.results);
            var gen = data.results[0].gender;
                $("#genderData").html(gen);
            var fName = data.results[0].name.first;
                $("#firstData").html(fName);
            var lName = data.results[0].name.last;
                $("#lastData").html(lName);
            var loco = data.results[0].location.state + ", " + data.results[0].location.country;
                $("#locoData").html(loco);
            var nation = data.results[0].nat;
                $("#natData").html(nation);
            var idnum = data.results[0].id.value;
                $("#idnumData").html(idnum);
          }
        });

    });
});
```

5

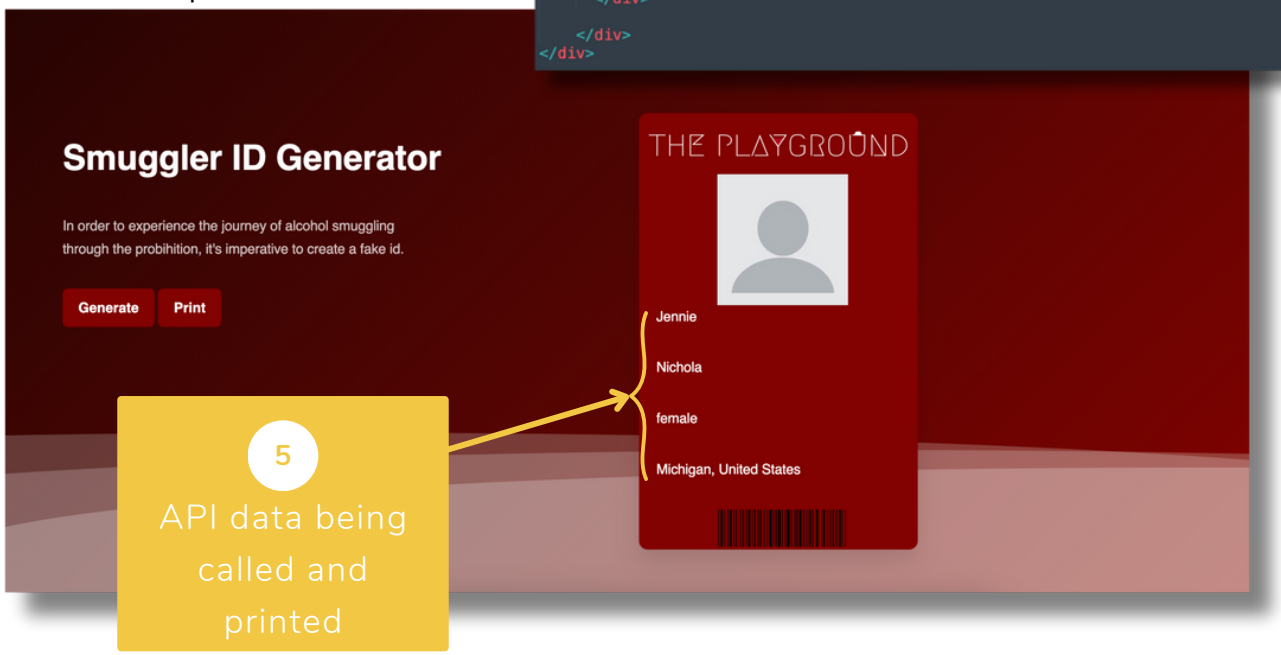
Copying the id from JS and pasting it in HTML (peak the highlighted code)

HTML File

```
<div id='printMe' class="col-6">
    <div class="card border-secondary mb-3" style="max-width: 20rem;">
      <img style="margin: auto; width: 300px; height: 50px; padding-top: 20px" src="assets/
      images/title.png">
      <div class="card-body">
        <img src="assets/images/profile.jpg">
        <p id="firstData">First Name</p>
        <p id="lastData">Last Name</p>
        <p id="genderData">Gender</p>
        <!--p id="natData>Nationality</p-->
        <p id="locoData">Location</p>
        <img style="margin-left: 70px; width: 150px; height: 44px;" src="assets/images/bar.png"
        >
      </div>

    </div>
</div>
```

Website Output



5

API data being called and printed

API INTEGRATION APPROACH

06 Printing Feature

The second functionality was to print only the IDs once they were generated. To do so, we needed to first create a JS function that only prints specific tags. Next, we added the function ID into the HTML, and then we concluded by adjusting CSS properties. The CSS modification was essential because only certain selectors can be printed without utilizing an extra plugin.

JS File

```
function printDiv(divName){
    var printContents = document.getElementById(divName).innerHTML;
    var originalContents = document.body.innerHTML;

    document.body.innerHTML =

    window.print();

    document.body.innerHTML =

}
```

HTML File

```
<div class="col-6">
    <div>
        <br>
        <h1>Smuggler ID Generator</h1>
        <p>In order to experience the journey of alcohol smuggling or/
        through the probihition, it's imperative to create a fake id.
        <div class="row">
            <div class="col-6">
                <button id="generate">Generate</button>
                <button onclick="printDiv('printMe')">Print</button>
            </div>
        </div>
    </div>
</div>

<div id='printMe' class="col-6">
    <div class="card border-secondary mb-3" style="max-width: 20rem;">
      <img style="margin: auto; width: 300px; height: 50px; padding-top: 20px" src="assets/
      images/title.png">
      <div class="card-body">
        <img src="assets/images/profile.jpg">
        <p id="firstData">First Name</p>
        <p id="lastData">Last Name</p>
        <p id="genderData">Gender</p>
```

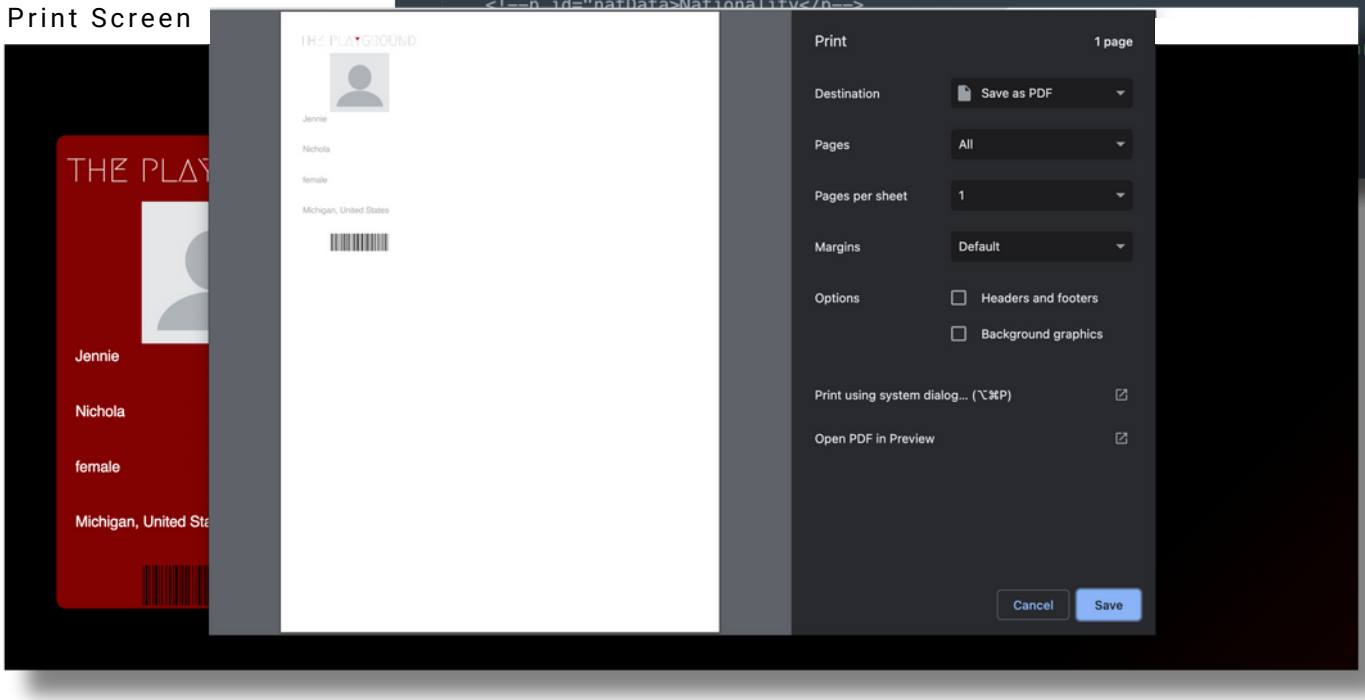
6

Button element "printMe" id connecting to <p> id "printMe"

6

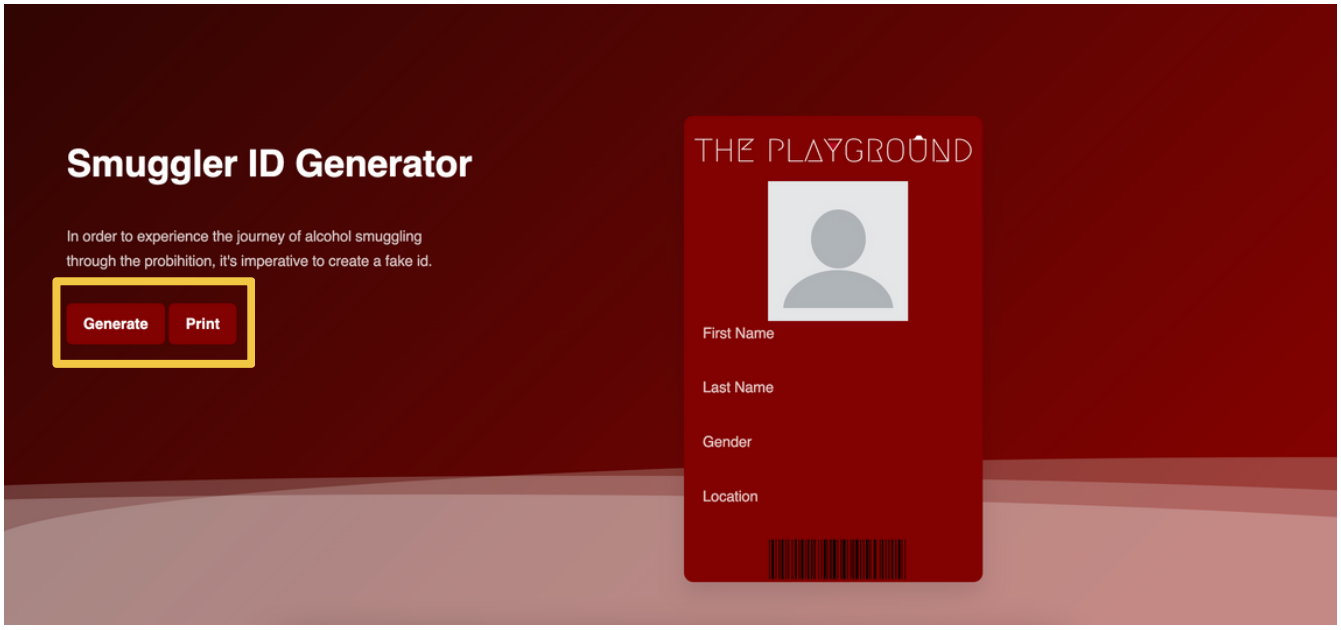
JS "printDiv" connecting to HTML "printDiv"

Print Screen



FINAL PRODUCT

Home



Generated ID examples

REFLECTION

This exercise was particularly challenging. The difficult parts were having to learn how to use the API, and then having to make the page look good after getting the API to work. The first step of learning how to use the API required watching many YouTube videos and looking at many sets of instructions online. Since we didn't learn about APIs in CGT 141, it made it difficult to implement and use an API. Another thing that was difficult about this exercise was the amount of time we had to complete it. With exercise 3, we had quite a while to create our web pages to make them function and look good. The second difficulty for this exercise was making the page look good in the context in which it would be used. Since we were allowed to follow the theme of Project 3 for this exercise, this made it easier to design what the final result would look like. Our final design followed the theme by using the API to create a "smuggler ID" which gives users a fake persona on an ID card which they can also use to redeem a drink at the speakeasy at the end of our museum exhibit. After finally getting the API to work and the design set, everything turned out well and even looked like it would fit right into our Project 3 web experience.