Designing for Data Informed User Experience

Interactive Project

Aim

The aim of the project is to allow you to demonstrate your technical understanding of web APIs through the creation of novel interactive interfaces.

Task

The project requires you to define a design problem and create a functional solution using dynamic data retrieved from APIs.

The Interactive project is broken into 3 components:

1) Project proposal.
2) The interactive project.
3) Production journal.

Requirements

Interactive project to be fully functional and available online. Code available in GitHub repository.

MUNCH
menu

PRODUCTION JOURNAL

David Grant - U3179044

University of Canberra – Bachelor of Web Design

DEVELOPMENT PROCESS

Before commencing this project, I ensured the requirements and the task was established as this was something I made an error in for the first API Sketch project. Following this, I explained to a number friends the power of API's and the possibility of data. I gave a number of examples of how API's could be used and one friend suggested an interactive application which provided users with a randomised meal generator. A number of other friends agreed and believed designing a solution to this problem would be useful.

SCOPING

I wanted to decide on what information I wanted to offer to the user. I decided that a user would have three options.

1) A randomised meal would appear on the application opening up
2) Users would be able to search for key words "Chicken" and be provided with a number of options of meals that included this.
3) Once a meal was found that they liked / loved, the user could save the meal into their favourites which would add to the user's experience.

PREVIOUS MISTAKES

Having a little more understanding following the first assignment on the power of fetching API data I wanted to keep the applications as simple as possible and to focus more on using a single script.js abilities. Focusing on design elements for more html files something I over engineered in the first assignment.

Taking on the feedback from my first assignment I resolved the following:

1) I compiled my JS file into one concise and only included files which the site would actually use. I got into the trap of submitting JS files previously as a 'just in case' mentality.
2) I removed any 'fancy' loaders for the site and ensured the theme of colours for the application were consistent.
3) I removed the need for the user to have various dashboards with no need to go between different pages.

API

In my initial proposal I had identified spoonacular.com as a possibility for fetching the required information to populate for the application, however in the end I used https://www.themealdb.com/ API. The reason I made this change was due to spoonacular.com upon reflection having almost too much API data to differentiate between.

My goal for this assignment was to have it functionally correct. In addition, I also discovered there were a number of other users on GITHUB who had also use similar API's.

I ended up fetching the following API's:

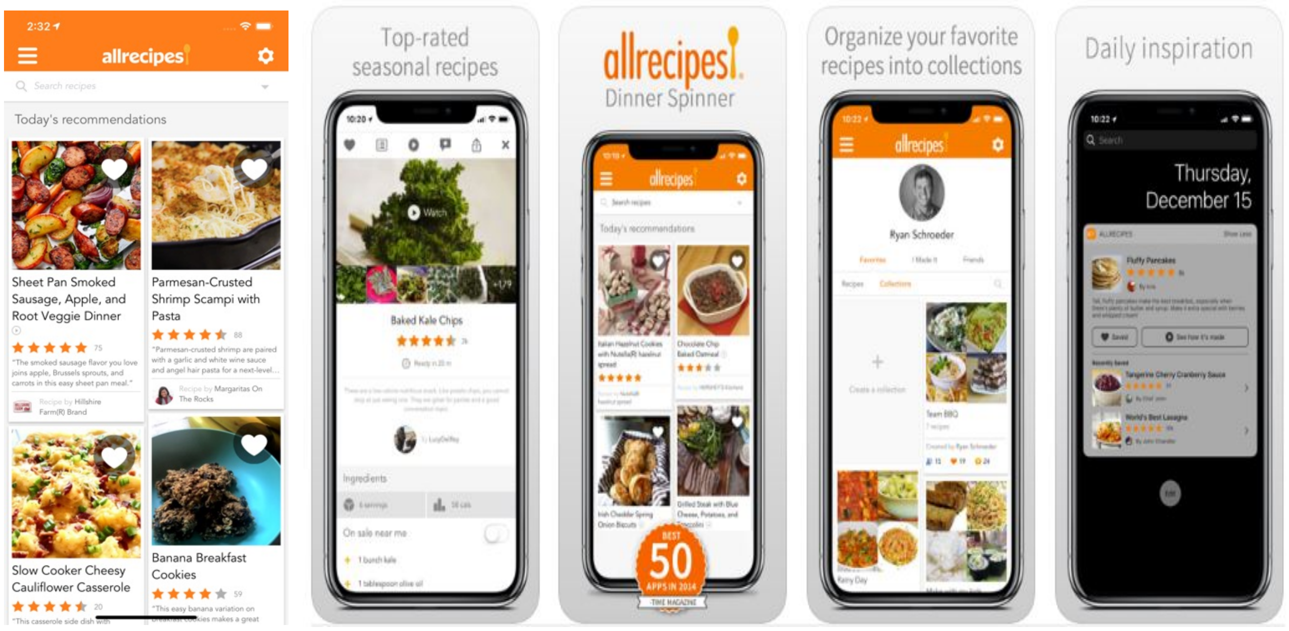
```
 // On page load or refresh of page, a random meal will be provided to the user.
async function getRandomMeal() {
    const resp = await fetch(
        "https://www.themealdb.com/api/json/v1/1/random.php");
    const respData = await resp.json();
    const randomMeal = respData.meals[0];
    addMeal(randomMeal, true);
}

 // Simply click the search tool will populate the list of meals of the Chosen API from
 themealdb
async function getMealById(id) {
    const resp = await fetch(
        "https://www.themealdb.com/api/json/v1/1/lookup.php?i=" + id);
    const respData = await resp.json();
    const meal = respData.meals[0];
    return meal;
}

 // Entering in key words "chicken" will provide the user with a list of meals with the
 ingriendent of chicken in title or in ingriendients.
async function getMealsBySearch(term) {
    const resp = await fetch(
        "https://www.themealdb.com/api/json/v1/1/search.php?s=" + term);
    const respData = await resp.json();
    const meals = respData.meals;
    return meals;
}
```

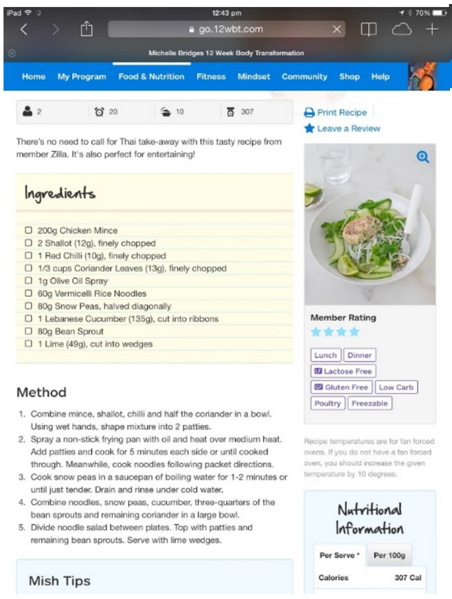
I was able to randomise the initial meal when a user opens the application, which was something I was unable to complete in assignment 1 and the application had to wait for a user to interact before the data returned.

I found it very useful in this project to, once I had a certain piece of code working effectively to put a // Description within the JS file.

LAYOUT / DESIGN

Having established the scope of the project, deciding on what information I wanted to display and what I was wanting the user to have access too helped in designing the UX for the user. I looked at a number of similar websites / application out there. The below is from and app named All Recipes.


allrecipes
Today's recommendations
Sheet Pan Smoked Sausage, Apple, and Root Veggie Dinner
Parmesan-Crusted Shrimp Scampi with Pasta
Slow Cooker Cheesy Cauliflower Casserole
Banana Breakfast Cookies
Top-rated seasonal recipes
Baked Kale Chips
allrecipes Dinner Spinner
BEST 50 APPS IN 2014
Organize your favorite recipes into collections
Ryan Schroeder
Daily inspiration
Thursday, December 15



Michelle Bridges 12 Week Body Transformation

Home · My Program · Food & Nutrition · Fitness · Mindset · Community · Shop · Help

2 · 20 · 10 · 307

Print Recipe
Leave a Review

There's no need to call for Thai take-away with this tasty recipe from member Zilla. It's also perfect for entertaining!

Ingredients

- ☐ 200g Chicken Mince
- ☐ 2 Shallot (12g), finely chopped
- ☐ 1 Red Chilli (10g), finely chopped
- ☐ 1/3 cups Coriander Leaves (13g), finely chopped
- ☐ 1g Olive Oil Spray
- ☐ 60g Vermicelli Rice Noodles
- ☐ 80g Snow Peas, halved diagonally
- ☐ 1 Lebanese Cucumber (135g), cut into ribbons
- ☐ 80g Bean Sprout
- ☐ 1 Lime (49g), cut into wedges

Method

1. Combine mince, shallot, chilli and half the coriander in a bowl. Using wet hands, shape mixture into 2 patties.
2. Spray a non-stick frying pan with oil and heat over medium heat. Add patties and cook for 5 minutes each side or until cooked through. Meanwhile, cook noodles following packet directions.
3. Cook snow peas in a saucepan of boiling water for 1-2 minutes or until just tender. Drain and rinse under cold water.
4. Combine noodles, snow peas, cucumber, three-quarters of the bean sprouts and remaining coriander in a large bowl.
5. Divide noodle salad between plates. Top with patties and remaining bean sprouts. Serve with lime wedges.

Mish Tips

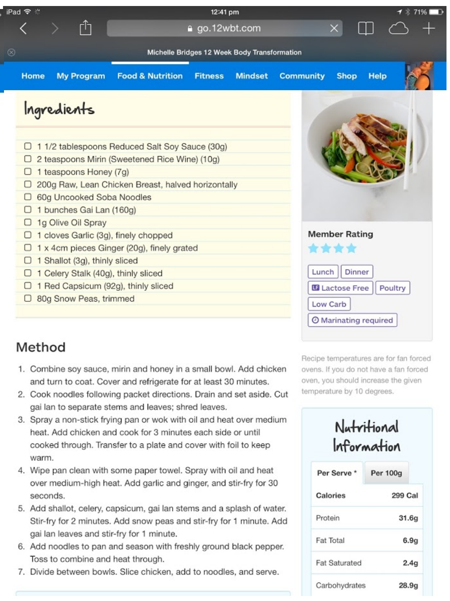
Member Rating ★★★★

Lunch · Dinner · Lactose Free · Gluten Free · Low Carb · Poultry · Freezable

Recipe temperatures are for fan forced ovens. If you do not have a fan forced oven, you should increase the given temperature by 10 degrees.

Nutritional Information

Per Serve *	Per 100g
Calories	307 Cal



Michelle Bridges 12 Week Body Transformation

Home · My Program · Food & Nutrition · Fitness · Mindset · Community · Shop · Help

Ingredients

- ☐ 1 1/2 tablespoons Reduced Salt Soy Sauce (30g)
- ☐ 2 teaspoons Mirin (Sweetened Rice Wine) (10g)
- ☐ 1 teaspoons Honey (7g)
- ☐ 200g Raw, Lean Chicken Breast, halved horizontally
- ☐ 60g Uncooked Soba Noodles
- ☐ 1 bunches Gai Lan (160g)
- ☐ 1g Olive Oil Spray
- ☐ 1 cloves Garlic (3g), finely chopped
- ☐ 1 x 4cm pieces Ginger (20g), finely grated
- ☐ 1 Shallot (3g), thinly sliced
- ☐ 1 Celery Stalk (40g), thinly sliced
- ☐ 1 Red Capsicum (92g), thinly sliced
- ☐ 80g Snow Peas, trimmed

Method

1. Combine soy sauce, mirin and honey in a small bowl. Add chicken and turn to coat. Cover and refrigerate for at least 30 minutes.
2. Cook noodles following packet directions. Drain and set aside. Cut gai lan to separate stems and leaves; shred leaves.
3. Spray a non-stick frying pan or wok with oil and heat over medium heat. Add chicken and cook for 3 minutes each side or until cooked through. Transfer to a plate and cover with foil to keep warm.
4. Wipe pan clean with some paper towel. Spray with oil and heat over medium-high heat. Add garlic and ginger, and stir-fry for 30 seconds.
5. Add shallot, celery, capsicum, gai lan stems and a splash of water. Stir-fry for 2 minutes. Add snow peas and stir-fry for 1 minute. Add gai lan leaves and stir-fry for 1 minute.
6. Add noodles to pan and season with freshly ground black pepper. Toss to combine and heat through.
7. Divide between bowls. Slice chicken, add to noodles, and serve.

Member Rating ★★★★

Lunch · Dinner · Lactose Free · Poultry · Low Carb · Marinating required

Recipe temperatures are for fan forced ovens. If you do not have a fan forced oven, you should increase the given temperature by 10 degrees.

Nutritional Information

Per Serve *	Per 100g
Calories	299 Cal
Protein	31.6g
Fat Total	6.9g
Fat Saturated	2.4g
Carbohydrates	28.9g

From this, I combined a number of attributes of both examples including the ‘love heart’ icon which allows users to add a meal to their ‘My Favourite Meal’ and when the meal is selected it provides the user with both the instructions and the ingredients needed to make the dish. Users also had the ability to search for key terms in meals, and get a return of options.

IMPROVEMENTS

In my original proposal I was wanting to have a user login where users could have a more interactive and personalised application which they could change. Being able to save meals and share meals with friends, being able to send the list of ingredients to an application such as Coles “click and collect’ would be ideal as users would save time in needing to go shopping. I also had the thought it would be good for users to add to an active API with their own recipes also. So having a ‘two way’ interaction with the app would allow for more involvement on the users end.

Note: Along the way of adding a number of Top navigations and Body formatting, I was unable to have the ‘mobile-container’ to flex and be responsive as the rest of the application.

REFLECTION

Having completed an application which overall responds to the users’ needs and fetches the correct information and displays it in a way the user can use was very rewarding. The initial scoping of the project and ensuring I was able to achieve my task was important. Although the presentation of the proposal in the initial stages was much more impressive ‘visually’ then the final application, I know this is something I can work on for future.

I ensured this time round I started the project almost immediately and allowed time for mistakes to be made and for many YouTube videos to be watched and learning from previous examples. Having the overall understanding of what information can be drawn from API’s allows for other future projects to be successful.

REFERENCES

Florin Pop. (2020, January 16). *Random Meal Generator - JavaScript tutorial - Day 16*. YouTube. https://www.youtube.com/watch?v=nYzrWnwpw34

Soegaard, M. (2020, July 26). *Interaction Design for Usability*. The Interaction Design Foundation. https://www.interaction-design.org/literature/article/usability-a-part-of-the-user-experience

Dev Ed. (2020, April 11). *Beginner Vanilla Javascript Project Tutorial*. YouTube. https://www.youtube.com/watch?app=desktop&v=Ttf3CEsEwMQ

HamScript. (2018, June 26). *Using APIs in React - Create a Recipe Application using React Router | React tutorial for beginners*. YouTube. https://www.youtube.com/watch?app=desktop&v=PbJt7-a2274

Allrecipes. (2016, June 28). *Allrecipes Dinner Spinner App - The Latest Version is Better Than Ever*. YouTube. https://www.youtube.com/watch?app=desktop&v=FQGB7DTJODA

Dani Krossing. (2018, November 4). *35: What Are Event Listeners In JavaScript | JavaScript Events | JavaScript Tutorial | mmtuts*. YouTube. https://www.youtube.com/watch?v=jqU3uaRgQyQ

JavaScript DOM EventListener. (2020, January 1). W3schools. https://www.w3schools.com/js/js_htmldom_eventlistener.asp

Code Bushi. (2019, December 21). *Fetch API & Rendering Data with JavaScript*. YouTube. https://www.youtube.com/watch?v=FN_ffvw_ksE&t=1517s

How to filter through JSON return from API with similar prop? (2018, March 30). Stack Overflow. https://stackoverflow.com/questions/49580528/how-to-filter-through-json-return-from-api-with-similar-prop

Loops and iteration - JavaScript | MDN. (2021, March 29). Developer Mozilla. https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Loops_and_iteration