

PUI FP4 - Final Project Report

Gaurav Nemade

Section A

PART I

As a busy graduate student, cooking is my creative outlet. Balancing time constraints with nutritious meals inspired my QuickBites website. Friends and colleagues expressing interest in my time-efficient culinary solutions fueled my commitment to sharing expertise on a broader scale. The QuickBites website offers a curated collection of recipes tailored for the hectic lives of students and singles, facilitating enjoyable, satisfying meals within tight schedules.

The website conveys a wealth of culinary information, from diverse recipes spanning various cuisines to detailed pictorial cooking instructions, and captivating images that bring each dish to life. QuickBites goes beyond being a mere collection of recipes; it is a culinary companion that fosters a sense of exploration in the kitchen.

What makes QuickBites truly interesting and engaging is its fusion of functionality and aesthetics. The website is designed with a clean and intuitive interface, ensuring a seamless user experience. Engaging features such as a random recipe selector add an element of surprise, encouraging users to broaden their culinary horizons. The integration of a WebGL API-driven heat haze effect adds a visually stimulating touch, enhancing the overall appeal of the website and making the cooking experience not only informative but also visually delightful.

The target audience for QuickBites is diverse, catering to individuals of all skill levels and culinary interests. Novice cooks seeking simple and beginner-friendly recipes can navigate the website effortlessly, while seasoned chefs will appreciate the extensive database of diverse and challenging dishes. The website's responsiveness ensures accessibility across various devices, making it convenient for users to access QuickBites whether they are experimenting in the kitchen at home or planning meals on the go.

PART II

Here is how would a user interact with my website:

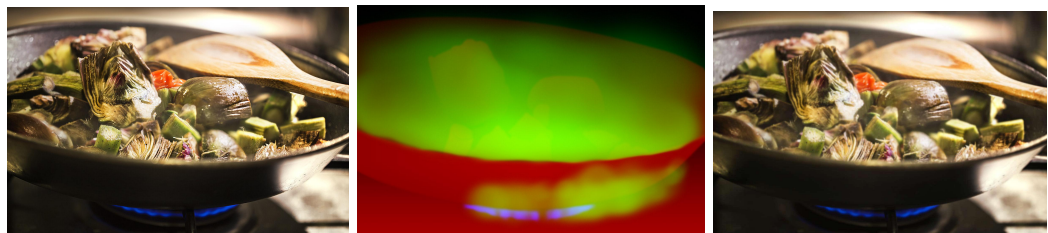
- **Hero Section Animation:** Hover over the hero section on the homepage to experience the animated effect.
- **View Recipes Button:** Click on the 'View Recipes' button below the greeting message on the homepage to navigate to the recipes section.

- **Recipe Cards:** Click on any recipe card in the recipes section to be redirected to the respective recipe page.
- **Step-by-Step Scrolling:** On each recipe page, scroll through the image section to navigate through each step of the recipe.
- **Back Button:** On a recipe page, click the back button to return to the homepage.
- **Slider Section:** Click on the arrows in the slider section on the homepage to navigate through different slides showcasing upcoming recipes.
- **Random Recipe Feature:** Scroll down to the 'Can't decide what recipe to make today?' section on the homepage and click the 'Surprise Me' button to randomly open a recipe from the database.

PART III

Here are some of the JavaScript APIs that I employed in my project:

- **WebGL (Web Graphics Library) API:**
 - **Why did I choose to use it?**
WebGL is a JavaScript API for rendering high-performance interactive 3D and 2D graphics within any compatible web browser without external plug-ins. The Fragment shaders and Pixel shaders in WebGL run a function for every pixel of the area that is being processed and return a color, which will be set for the said pixel. Using this API, we can animate 2D images with graphically animated overlays.
 - **How did I use it?**
I took an image of a cooking pan and made two other copies of the same image one with a blue filter and the other one with a heatmap. These images are superimposed over each other using the fragment and pixel shaders function in WebGL. The blur image animates over the heatmap to produce a heat haze effect.



- **What does it add to my website?**
This API enables my website to have an interactive and live animation. It adds to the aesthetics of the website and enhances the user experience.

- **Swiper JS API:**

- **Why did I choose to use it?**

Swiper JS is a free and open-source mobile touch slider with hardware-accelerated transitions and amazing native behavior. Use it on websites, web apps, and mobile native/hybrid apps. I chose to use this API as I needed to add the scroll functionality for each recipe image on the recipe page. Swiper JS allows me to seamlessly traverse through the different images in recipe steps and on scroll.

- **How did I use it?**

I used the swiper-slide class from Swiper JS on my recipe card template on the recipe page. This template is connected to the database that contains paths for different images for each step in the recipe. Adding the card-swiper class on this template enables me to fetch the respective images from the database and switch between them on scroll. I applied swiper-wrapper to the container that contains this template which is essentially a wrapper that adds the functionality of the scroll or swipe (on touchscreens).

- **What does it add to my website?**

Swiper JS adds the functionality to swipe and scroll on particular objects within the website. It allows room for animations like fade in transitions when scrolling and swiping through the template section. It also adds pagination for the slider to add a semantic indicator for the slides.

PART IV

During the implementation of this website, I underwent several iterations, endeavoring to explore diverse user perspectives and functionalities:

- **Iteration 1:** Initially, my homepage lacked any animations, featuring a static design with informative content. The sole interactive components were cards directing users to various recipes.
- **Iteration 2:** After some feedback from my peers in class, I changed my homepage design to incorporate some sections like random recipe selector which would allow the user to randomly select any recipe from the entire recipes database. Additionally, I incorporated header and footer sections to prominently display the website's logo and branding which also made my website look complete.
- **Iteration 3:** While implementing and testing APIs for my website, I stumbled upon a captivating heat haze effect utilizing the WebGL API. Recognizing its potential, I saw an opportunity to seamlessly integrate this effect into my website on the homepage, particularly in the context of cooking.

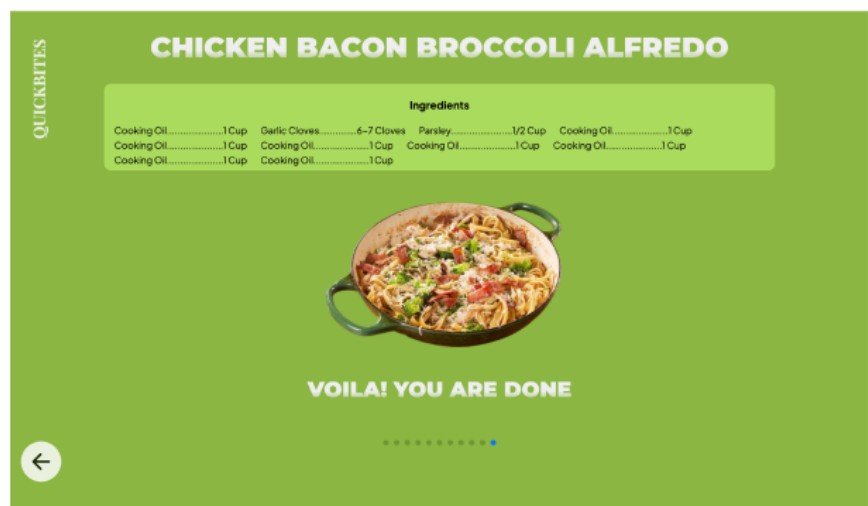
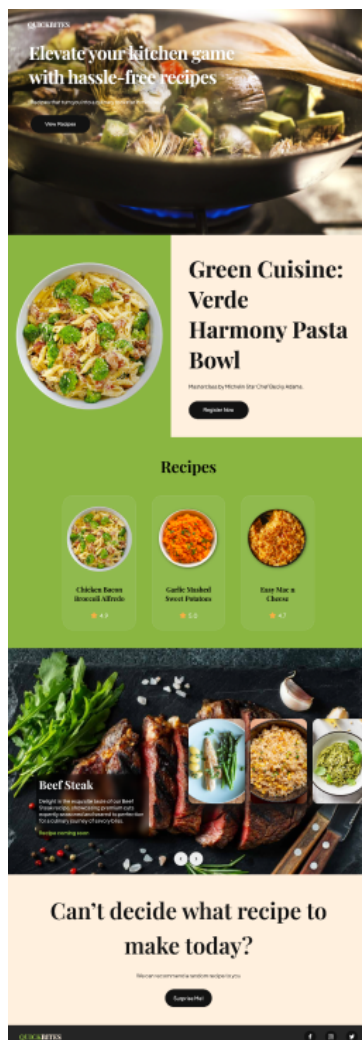
Part 5

Here are some of the challenges that I encountered when implementing this project:

- The most challenging part was integrating the WebGL API as it required me to use npm server from Node JS. Not having used Node JS before, I found it difficult to figure out how to build the code and run it using a npm server.
- Dealing with a website that heavily relies on images, I encountered a significant time-consuming task of establishing a database containing all the necessary asset links.
- Another challenge I faced was ensuring the website's responsiveness for tablet screens, requiring thoughtful adjustments to accommodate varying screen sizes and provide an optimal user experience.

Appendix

Snapshots of my website:



Screenshots from WAVE:

Homepage

The image displays two screenshots of the WAVE (Web Accessibility Evaluation Tool) interface, powered by WebAIM. Both screenshots show the tool evaluating the URL: <https://gnemade51.github.io/pui-project/>. The 'Styles' toggle is set to 'ON'.

Summary View (Left Screenshot):

- Errors:** 0
- Contrast Errors:** 0
- Alerts:** 3
- Features:** 8
- Structural Elements:** 17
- ARIA:** 0


A message at the bottom states: "Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility." A "View details" button is present.

Details View (Right Screenshot):

- 3 Alerts:**
 - ☐ 3 X Redundant alternative text
- 8 Features:**
 - ☐ 7 X Alternative text
 - ☐ 1 X Language
- 17 Structural Elements:**
 - ☐ 1 X Heading level 1
 - ☐ 3 X Heading level 2
 - ☐ 9 X Heading level 3
 - ☐ 1 X Heading level 4
 - ☐ 1 X Unordered list
 - ☐ 1 X Navigation
 - ☒ 1 X Footer

A note at the bottom of the details view states: "If an icon does not appear within the page, turn off Styles above to view it."

Recipe Page







**WAVE**
web accessibility evaluation tool

powered by
[WebAIM](#)


Address: <https://gnemade51.github.io/pui-project/rec> →


Styles: OFF ☐ ON ☒


Summary





SummaryDetailsReferenceOrderStructureContrast


**0**
Errors


**0**
Contrast Errors

**1**
Alerts


**2**
Features

**3**
Structural Elements

**7**
ARIA

View details >

Congratulations! No errors were detected! Manual testing is still necessary to ensure compliance and optimal accessibility.







**WAVE**
web accessibility evaluation tool

powered by
[WebAIM](#)


Address: <https://gnemade51.github.io/pui-project/rec> →

Styles: OFF ☐ ON ☒


Details




SummaryDetailsReferenceOrderStructureContrast

☒ **1 Alerts**


☐ 1 X Redundant link

☒ **2 Features**

☐ 1 X Linked image with alternative text
☐ 1 X Language

☒ **3 Structural Elements**

☐ 1 X Heading level 1
☐ 1 X Heading level 4
☐ 1 X Header

☒ **7 ARIA**

☐ 2 X ARIA
☐ 1 X ARIA label
☐ 1 X ARIA tabindex
☐ 2 X ARIA alert or live region
☐ 1 X ARIA button

If an icon does not appear within the page, turn off Styles above to view it.

6