PART A

**Associate a README.TXT with the app outlining technical implementation matters**

Included in proj3 git repo

**Write HTML help document and associated ? icon covering common user interface elements in the UI**

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

**Write a Roadmap going forward of the developments to the interface, include in development repository where code is stored**

A close up of a piece of paper

Description automatically generated

**Move 3rd party dependencies to CDN repositories remove 3rd party files from codebase**

I tried to move Bootstrap and VeeValidate to CDN repo and remove 3rd party file from code base.

After hours of trials and errors… no success. Below listed some error I captured.

This two happened when I remove vee-validate from node\_modules and add CDN.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

This code shows an example of how bootstrap-vue official document tell me how to include their CDN and work with it. I tried to apply it, but it gets a runtime error saying my code is for runtime only that don’t work with template complier

A screenshot of a social media post

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Try to include this code in webpack.config.js as bootstrap-vue show but no success..

A close up of a logo

Description automatically generated

If I include CDN in public/index.html and initiate my vue like this

A close up of a screen

Description automatically generated

I will get errors saying my bootstrap tag like <b-row>, <b-container>, <b-alert> are not registered.

And then I see BV example mentioned “Get started quickly without the need for a build system, by using standard <script> and <link> tags to load the required JavaScript and CSS in your page.”

A picture containing knife

Description automatically generated

I searched for runtime-only build and get this result

<https://gist.github.com/anchal20/f2ac9807263e106c1308f7143df1cf09>

I am not sure how can I do that. I tried different methods and still not success.

After hours of searching, I decided to move on from this question first.

**Why is it important to implement the above (1-4)**

For ‘1. Associate a README.TXT with the app outlining technical implementation matters’

This can provide a brief description of what technology in used in the project and also provide information about how we corporate those technology together. With this information, we can have an idea of what the final outcome might look like.

For ‘2. Write HTML help document and associated ? icon covering common user interface elements in the UI’

This help client who are not familiar with our site to understand the use of all the button and input form.

For example:

A screenshot of a cell phone

Description automatically generated

The bin icon in the comment section gives some hint to client that they can delete their comment if needed. The graphic provides a bigger hint compare to plain text.

For ` 3. Write a Roadmap going forward of the developments to the interface, include in development repository where code is stored`

It plans the future development and let you think which step should go next. The roadmap should outline the functionality needed and what new UI interface element is needed to make the function works.

For `4. Move 3rd party dependencies to CDN repositories remove 3rd party files from codebase`

I am not sure if CDN is a must-have because you cannot guarantee a CDN would work 100% of time. If the link of the CDN went down, your frontend elements that depends on it will also went down. In a local stored file, it will not be a problem because everything is settled. However, CDN do have some advantages over local stored files. It gives a faster interaction after the data from CDN is cached. It outsources some frontend data exchange to make user interact with the site faster.

**What other types of documentation may be necessary for this project?**

We have our project outline in PROJ1, frontend prototype in UX1, server-side explanation in PROJ2, technology chosen, implementation description and future development roadmap in PROJ3 and UX3.

I think at this point, a database review documentation outlining how different tables in database work together might be helpful for this project. We can have a better picture of which piece of data is the most important, which piece of that data can be replaced. Because as the website grow and more features added, we might need more data input from the client. At the same time, however, it is not ideal if client need to fill a form that takes forever to finish. With a database review, we can determine which data is not necessary and delete it. Even if all existing data is valuable, we can see how our ‘old’ data can work with the ‘new’ data input from the feature and see if we need our database structure changed.

PART B

**What portions of the development went particularly well**

To be honest, I do not think any part of the development ‘went particularly well’. If I have to choose one, I would say using Bootstrap-Vue to control frontend elements is the simplest part in the project. It may because I have experience in Bootstrap 4 so the grid control (b-row, b-col), button control, different colour variant (primary / warning / danger) makes more sense to me. Some of the knowledge in Bootstrap 4 can also apply to Bootstrap-vue

**What was the most difficult to implement**

Instead of ‘the’ most difficult part to implement. I would give 3 most difficult parts I have encountered.

1. Data exchange in Vue

In Vue, you cannot change the props data in child directly which is different with plain JavaScript. All the props I want to change in child components I would have to $emit a function from child to parent and have a different function to handle the payload in $emit in order to change the data in parent component that passes the data to child. If my description makes you confused, that means you understand me feeling when I try to bind data in Vue. However, it does make sense because every time a component re-render, the data value set as default value. Therefore, the changes you made from child components is gone. It is not so hard to understand why after I invested more time into Vue. Maybe this problem exists only because my lack of knowledge in Vue or simply because I did not use vuex. I would like to share it just because I want it to be a true reflection on the project.

1. Deployment

There are a lot of deployment tutorials teach you how to deploy site in a certain format. For example, in Vue official site in the deployment section, it has information provided on how do you deploy a Vue site on different service provide like Heroku, Amazon S3, Firebase etc. However, it is hard to find information that fit my needs. I wanted to deploy my site which use Vue as frontend, PHP API as backend and I want to deploy them together in a folder. I created a public directory (Vue) in server side php directory and tried to deploy it together and it failed miserably. I believe I am missing a piece of code that can redirect the request and response from server side to client side. After hours of searching and still not find it so I deploy my frontend and backend separately and it went well. It is not so difficult if I follow the steps on official site and deploy them separately, but it is difficult if I want it to deploy the way I want it to be.

1. Move 3rd party dependencies to CDN repositories remove 3rd party files from codebase

I have to add this. Please refer to part A.

**If you had the chance to do this again, what would you do differently**

I would have invested more time in Vuex to see if I would have a better experience in data binding in vue.js

**What parts of the implementation incomplete at this stage of delivery.**

Data from PROJ1 doc.

* Admin have full control on edit/delete posts or comments to prevent intentional misuse
* Show the best fitted result when user search with ingredients on hand
* User can give a thumb up on recipe
* Recipe with more thumb up will be on a higher priority of the list in search result
* “Are you over 18 years old?’ will be asked before access to main page
* If visitor answered ‘NO’ in ‘Are you over 18 years old?’, redirect to [Alcohol laws in Australia](https://www.health.gov.au/health-topics/alcohol/about-alcohol/alcohol-laws-in-australia)
* Menu contain a list of [6 types of spirits](https://www.thespruceeats.com/quick-guide-to-distilled-spirits-760713) to choose from
* User activates will be logged in database
* Admin will blacklist user that intentionally put invalid data to database
* Simple graphic element like icon will be used for easy navigation
* User can input ingredients in search bar and get the result of which cocktail can make with those ingredients

Green = done, Red = not done

**Write and reflect on "Quality Assurance" how are you practicing this?**

In proj2, I created test scripts include the below scenario:

(situation) -> (https response code)

1. Non-logged in visitor read recipe -> 200

2. Non-logged in visitor create/update/delete recipe -> 401

3. Logged in user create/update / delete own recipe -> 200

4. Logged in user update/delete recipe don't belongs to him -> 403

5. Admin update/delete recipe don't belong to him -> 200

6. Non-logged in user create/delete comment -> 401

7. Logged-in user create/delete own comment -> 200

8. Logged-in user delete comment don't belongs to him -> 403

9. Admin user delete comment don't belongs to him -> 200

10. Search with keyword to get a related recipe -> 200

11. Register an account -> 200

12. Login after register -> 200

Also, I manually check every function my project provided (two CRUD and search function).

That’s because the test scripts can tell me if my function working properly, but It can’t tell me the real feeling when clients using my site. Are there enough graphic elements to give hints to client what to click? Are the frontend runs smooth or it lags after certain action taken? Is the form validation feedback enough to provide a guideline to clients? Those are something will not show expect I actually use the site.

**How much of the prototype UX1 remains in the final project?**

UX1:

A picture containing sitting, refrigerator, phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

A picture containing sitting, black, food

Description automatically generated A screenshot of a cell phone

Description automatically generated A picture containing food

Description automatically generated

Final project:

A close up of food

Description automatically generated A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

A close up of food

Description automatically generated A picture containing food

Description automatically generatedA screenshot of a cell phone

Description automatically generated

The layouts are similar, colour chose for button and background are similar. The light/dark theme toggle is no more in the nav-item but in the footer of the nav-bar. I would say the frame of UX1 is still there, but more well-formatted.

**Where has your project Object Oriented programming implemented**

Frontend JavaScript for data handling and backend PHP(PDO) for database configuration and date exchange.