

A responsive web application to encourage home-cooking and reduce food waste

Contents

Draft Project Report.....	2
Problem description.....	2
Account of related literature	2
Account of project work and its outcome	3
Review.....	5
Review of current stage of project work	5
Review project management.....	9
Assess risks to project completion.....	10
Review of personal development	11
References	12
Appendices.....	13
Appendix A.....	13
Appendix B.....	14
Appendix C.....	15
Appendix D.....	18
Appendix E	19
Appendix F	20
Appendix G.....	21
Appendix H.....	22
Appendix I	23
Appendix J.....	24
Appendix K.....	25
Appendix L.....	26
Appendix M.....	28
Appendix N.....	29
Appendix O.....	31

Draft Project Report

Problem description

The problem that I am aiming to solve is that of food wastage in the average household. It is estimated that each year 1.3 billion tons of food produced worldwide is wasted, wasting money for the consumer, and increasing the worldwide levels of greenhouse gases (Kirk & Scott, 2018). My solution aims to reduce waste at the consumer level, by encouraging the cooking of ingredients nearing or in some cases just passed their expiration date. It will do this by providing facilities that will enable the user to search for recipes with the ingredients that they have in their fridge that were potentially going to be discarded. The user will also be able to manage an online store cupboard where they can enter items that they would like to be included in these recipes that they may already have stored at home. There is of course a potential risk with any food that is beyond its use-by date, and so the application will also provide some information on food safety to enable the user to make safe decisions about what to eat.

The goal of the project is to deliver a responsive web application that:

- Allows searching of online recipes by ingredients which are specified by the user
- Allows viewing of method, nutritional information and ingredients for selected recipe
- Provides information on food safety
- Allows maintenance and viewing of an online store cupboard

The solution that I am proposing is that of a web-based application, that can be used on desktop or mobile devices, due to its responsive nature. I aim to build the application using a JavaScript framework, that I will research and utilise during project. The possible frameworks that I have explored are React and Vue. I have some previous experience using Angular, and so for this project I would like to gain some new skills and avoid the use of Angular if possible. I also aim to integrate with an 3rd party recipe API to provide the search feature for my application.

The scope of the project has been expanded since my last TMA to also include the implementation of a database to store the store cupboard entries. This means that I will need to implement a login feature, so entries can be stored and retrieved for a user, and the scope of the project now includes coding for the UI elements concerned with logging in, as well as user authentication and authorisation and data security activities. Data access code will also need to be written to communicate with this database, and hosting concerns will also need to be considered.

The project scope also involves the evaluation and selection of several tools that I will use throughout the project, including that of an integrated development environment (IDE) and database tools.

Account of related literature

I have made use of 'Jump Start Responsive Web Design' (Ward, 2017) to research responsive design principles and techniques while carrying out research for my project. This book provides an in-depth view of many topics concerning responsive design, including what it is, ways in which it can be achieved and code examples of how techniques can be implemented. There are multiple chapters within the book which anticipate will be valuable while I am implementing the CSS code for my responsive design. The accompanying coding examples contained in the book have also given me

practise of implementing the techniques specified. I can find little information about the author of the book online, but other readers have reviewed the material highly, so I deem it to be a very useful resource. The only limitation of the book is that it does not go in to detail about the use of responsive CSS frameworks, which is an area I wanted to explore before choosing how best to implement the styling of the application.

A main source of information that I have utilised extensively during the project is the documentation available from the developers of the Vue.js source code (Vue, 2018). I can count this as a reliable source of information as it is produced by the developers who create the code for the framework and it is in their best interest to keep it relevant and up to date so that developers such as me adopt the use of the framework. In conjunction with the Vue.js documentation I have used the Vue router guide (router.vuejs.org) to implement routing within my application and Vue CLI documentation during the setup of my project (cli.vuejs.org). These documents provide further information than that supplied in the Vue.js reference documents, but are linked to from within the Vue.js documentation and provided by the same community of developers, and so I have treated these resources as highly reliable

While experiencing issues during my project set-up, I used online information source Stack Overflow (Stack Overflow), which is an online information sharing website, where the developer community can ask and answer questions on various development issues. I found this an extremely valuable source of information and foresee me using it throughout the project when experiencing issues in many areas of the project, including coding, database implementation and hosting. I consider Stack Overflow a reliable source of information due to the point system that is used by the website to indicate the best quality answers. Using this tool, I can ensure I am using the highest quality answers to solve my issues.

Video content site Pluralsight has proven invaluable in my decision to use Vue.js in my project. By watching videos on both React and Vue.js I was able to determine which framework was better suited for my project. The authors of content on Pluralsight are carefully auditioned before they can post content on the website, and as such I consider this to mean content is a reliable source of information (Pluralsight.com, n.d)

During coding activities, I have used the W3C website (w3cschools, n.d) for reference to find the appropriate JavaScript methods to use for my needs, specifically when trying to encode a URI to send as an API parameter. Due to the naming of the website I assumed it had an affiliation with the W3C Consortium, in which case I could class this resource as highly reliable, but on further investigation it seems that this website is a separate entity, and perhaps not a totally reliable source of information at times (Post, 2017). Following this discovery, I made sure to check the official JavaScript specification document to ensure that the method I have used has not been deprecated, which I was glad to find that it hadn't (Mozilla, n.d).

I have also been utilising the documentation provided by the developers of the Edamam API during my integration of this with my project (Edamam, 2019). I have treated this as a reliable source of information due to the fact that is provided by the creators of the API, and is likely to provide accurate instructions and information.

Account of project work and its outcome

To begin to solve the problem firstly it was essential to finalise the requirements for the product that would provide the solution. I did so by interviewing a potential user of the application to gain

insight into a 3rd party's opinions on the potential solution for the problem. Without an outside perspective it is very probable that my final solution may not have solved the problem effectively, as it would have been based purely on my opinions and ideas. A transcript of the interview can be seen in Appendix A. Following the interview, I compiled a list of requirements that I deemed essential for the solution to solve the specified problem, which is shown in Appendix B.

The next task was to create wireframes for the application based on the gathered requirements. These wireframes are low-fidelity, as I feel the requirements are quite simple and don't require any complex prototyping to assess their validity in solving the problem. I produced wireframes for both a standard desktop view and mobile device, which can be seen in Appendix C.

Using these requirements and wireframes, I then wrote an acceptance test script, using the approach described by Scott Ambler (n.d). I have chosen this approach as I have experience creating tests in this format from my previous OU study during TT284 and feel that the tests created following this template provide good detail about how to carry out and what to expect from the tests. This will be important if I utilise the help of an external party to carry out the tests to provide extra assurance that the requirements have been met. An example of the test cases that I have written is shown in Appendix D.

Following the analysis and design stages, I then began multiple research tasks. The first was to choose the JavaScript framework that I would use to create the components for the application. Using various sources, I settled on the use of Vue.js due to its simplicity and similarity to frameworks that I have used before. The second was to research principles and solutions for responsive design, which I am still yet to complete. The third was to find an appropriate 3rd party API to integrate with, and settled on Edamam (Edamam, 2019) as this offered the functionality closest to the needs of the application.

I then set up the coding environment, choosing VSCode as the code editor that I would use, as it is lightweight, highly-regarded in the developer community and it provides out-of-the box support and tools for writing TypeScript (Holland, 2018). I then set up a GitHub repository to ensure that my work is backed up should I suffer any hardware failure. The recent commit history for the Git repository can be seen in Appendix E.

After this step, I began to read the Vue.js documentation, and eventually began code implementation. This firstly began with me installing Node.js and Vue CLI as this was the approach I decided to use after consulting the installation documentation (cli.vuejs.org). I chose this approach for installation as I believe it will provide good tooling for later on in the project when I need to think about packaging and deploying the project, although I have not delved too far into this area yet.

Coding began with creating the structure of the application, with a navigation and header component being created. I also created dummy pages for the application so that I was able to set up the routing for the application at this point. To do this I needed to consult the Vue Router documentation (router.vuejs.org). The navigation component code is shown in Appendix F.

The next section that I began implementing was the Recipe Search page. This began with adding HTML code for a simple form to accept ingredients that are added to a list of ingredients to use as parameters for the Edamam API. I created components for a RecipeListItem and the RecipeList which displays these ingredients (Appendix G & H). Following this I then began implementation of the recipe search using the Edamam API. After consulting the Vue.js documentation I decided to adhere the method outlined there, which included using the axios library, which I installed using the Node Package Manager (npm, 2019). Following the example in the Vue.js documentation, I then

added code to call the Edamam API. Appendix I and J show the full Recipe Search code, and Appendix K shows the Network tab of my browser, showing successful communication with the API. At this point in development I realised that the response provided from the API does not include a method for the recipe, which means that some of the requirements of my solution may not be met. As during my research stage of my project I was not able to find any suitable alternative to the Edamam API I will have to perceive with this API, but the solution will not be complete.

Following integration with the API I then created the `RecipeListItem` and `RecipeList` components, which use the `label` property from the JSON response to display a list of the available recipes. These components are shown in Appendix L.

I created a router link within the `RecipeListItem` component which links to a `RecipeDetails` component that I created. This page receives the `URI` for the selected recipe as a route parameter and uses it to call the Edamam API to get the recipe details for that recipe. This component then displays the ingredients for the recipe (Appendix M).

The progress that has been made from a user interface perspective is shown in Appendix N.

Now that I have made good progress with the HTML and JavaScript implementation for these components it is imperative that I implement the styling for the components, which is the next task that I aim to complete.

Review

Review of current stage of project work

I feel that I have made good progress with the project so far, although I have fallen slightly behind in the schedule that I set for the project. This was partly due to my focus being on the other Level 3 OU module that I have been studying alongside trying to complete my project, but as this module has now been completed I anticipate I will be able to focus entirely on the project completion and make up the lost hours.

I'm also behind in the schedule due to the time it took me to become confident enough to start coding with Vue.js, as it was quite a learning curve, and I didn't feel confident to get coding for a significant amount of time. I feel I wasted too much time reading the Vue.js documentation, and should have dived into coding sooner, as I learnt a lot more by 'doing' than I did from simply reading. I will take this into consideration when implementing the database and backend code features, and make sure I am implementing code as soon as possible, and learning as I go.

My task list does show a lot of tasks as incomplete, but I have made partial progress in the completion of a good proportion of them, so I am still confident in my ability to complete the project successfully, especially now that I can focus my attention on the project.

The main task that I am significantly behind schedule with is the responsive design research and CSS implementation for the project. I have made some progress with the reading required, but still have several chapters of the 'Jump Start Responsive Web Design' (Ward, 2017) left to get through. I am worried that this reading may have been a waste of time, as although it is very good for my personal development, I think it may be more time effective to cease the reading and investigate the use of a CSS framework such as Foundation or Bootstrap. Now that I have made good headway with the JavaScript code implementation, my next immediate task is to get the responsive design

implemented. I would like to implement responsive styles from scratch but need to be aware that if necessary an alternative must be found.

The requirements for the final solution have been tweaked slightly as the project has progressed, namely to include deletion of a recipe search item, and to show a loading indicator while communicating with the API's. I don't think that these changes have greatly affected the scope of the project however, and I should be able to fit these additional items into the schedule without issue. Although I have previously stated that I wanted to avoid requirements creep, I do think in this case the additional requirements are essential to provide a good user experience and therefore a good solution to the problem.

The following shows the tasks that need to be carried out to complete the project satisfactorily, along with their associated risks.

Tasks	Sub-Tasks	Resources required	Skills required	Risk	Complete?
Planning and Design	Produce and evaluate wireframes	Computer based wireframing tool	Wireframing skills using chosen tool, which I have from previous module studies.	Low risk due to experience of wireframing during previous modules and familiarity with chosen tool.	Y
	Write acceptance tests	Usefulness to project relies on a thorough and useful set of requirements being gathered from my potential users	Skills on how to write acceptance tests will need refreshing as have not done since level 2 modules.	Carrying this task out early in the process means requirements need to remain fixed for tests to remain meaningful and useful.	Y
	Carry out responsive design research	Reliable learning sources in this area	Basic CSS and HTML understanding, which has been gained from previous modules.	Time concerns with learning, potential for different methods of achieving desired responsiveness may lead to research overload. Can be mitigated by keeping research as brief as possible and concentrating learning on favoured approach.	In progress
Recipe Search	Write HTML, CSS and JavaScript for <i>Recipe Search</i>	IDE for development	HTML, CSS and JavaScript	I may need to begin this task before I feel I have researched the new skills required thoroughly enough, due to time constraints.	In progress
	<i>Recipe Search</i> testing	Potential users could be utilised to provide more thorough testing	Issue tracking system for bug, need to research what would be suitable. Trello could possibly be used as project tasks are already tracked here.	I have no separate testing resource other than myself, so must ensure I do not become blinkered and miss obvious issues. Could include users if possible to carry out testing to mitigate this.	N

Tasks	Sub-Tasks	Resources required	Skills required	Risk	Complete?
	Recipe Search refinements	IDE for development	HTML, CSS and JavaScript		N
	<i>Recipe Search</i> API integration	Web search engine, thorough and readable documentation provided by 3 rd part API provider. API testing software.	Those gathered during TM352 module	There may not exist an API that provides the information that I require for the application, or data may not be in a format that the application can easily consume. Documentation could be lacking in areas I need, or API could become unavailable for some reason. Relies on 3 rd party. In this case there is not much I can do to mitigate these risks, and must hope for the best.	Y
Recipe Search Results	Write HTML, CSS and JavaScript for <i>Recipe Search Results</i>	As above	As above	As above	In progress
	<i>Recipe Search Results</i> testing				N
	<i>Recipe Search Results</i> refinements				N
View Recipe	Write HTML, CSS and JavaScript for <i>View Recipe</i>				In progress
	<i>View Recipe</i> testing				N
	<i>View Recipe</i> refinements				N
Database	Research and implementation of database	Documentation for database technology chosen, potentially additional database management software	Either relational database or document-based approach will be adopted, experience of which I have from TM351, but that will need expanding upon	This was not in original scope of project, and may be a push to fit in, depending on how long it takes to decide on an approach to use and implement the scheme.	
	Write code for data access layer	IDE and documentation on desired implementation.	It should be possible to utilise existing coding skills to produce this code, but may require some research regarding the	This is the task I feel least comfortable with and may take a while to implement. Need to use a technology compatible with Vue.js and database technology used, which will probably require	N

Tasks	Sub-Tasks	Resources required	Skills required	Risk	Complete?
			desired architecture of software components.	additional learning and time.	
Login	Implement user login feature	IDE, documentation on best practises, or potentially a 3 rd party authentication provider such as Auth0	Existing coding skills can be utilised, but knowledge on the area and best practises will need improving.	This task could also take a fairly long time, as it is critical to the application security and must be done properly.	N
Add Store Cupboard Item	Write HTML, CSS and JavaScript for <i>Add Store Cupboard Item</i>	As above	As above	As above	In progress
	<i>Add Store Cupboard Item</i> Testing				N
	<i>Add Store Cupboard Item</i> Refinements				N
View Store Cupboard Items	Write HTML, CSS and JavaScript for <i>View Store Cupboard Items</i>				In progress
	<i>View Store Cupboard Items</i> Testing				N
	<i>View Store Cupboard Items</i> Refinements				N
	Write HTML, CSS and JavaScript for <i>Food Safety Index</i>				N
Hosting	Website and database hosting	Hosting platform	I have gained knowledge about cloud hosting from previous OU modules, but this will need refreshing	I will need to find an appropriate platform for hosting which may take time and require refreshing of the knowledge I gained in the area from previous modules. It may be difficult to do without incurring costs.	N
Testing	Acceptance Testing			Any shift in requirements will affect this task	N

Tasks	Sub-Tasks	Resources required	Skills required	Risk	Complete?
	Cross browser and mobile device testing	Tool that allows emulated testing across multiple browsers	Knowledge of how to operate various types of devices for testing purposes.		N
Module tasks	TMA01				Y
	TMA02				Y
	TMA03				Y
	Final project report				N

The Gantt chart shown in Appendix O shows the current stage of progress in the project.

Review project management

The lifecycle model that I initially decided to adopt for this project was an iterative waterfall approach. I planned to predominately undertake each stage of the project in sequence, rather than iterating multiple times, but planned to iterate within the implementation stage by completing some testing in this stage. In practise I have not completed this testing after the implementation of each feature, and now plan to complete all acceptance testing once the implementation has been completed. Therefore, in practise, I have used a more standard waterfall project lifecycle. This was not a conscious change in methodology, but the momentum I gathered from the enjoyment of learning Vue.js meant that I was eager to carry on coding linked application components rather than stopping the test a feature before I moved onto the next, and I didn't want to slow down the progress I was making with learning Vue.js. I think this change of approach is quite specific to this project as I have been learning a new framework, and if I were more familiar with Vue.js I feel I would take the time to test each feature before moving on to the next.

I feel that I have not rigidly stuck to the waterfall lifecycle due to the poor project management on my part, namely because I have fallen behind with some tasks. This means that although I am currently in the implementation phase of my project lifecycle I may need to revisit the design stage once I have completed my responsive design research. I hope if this is the case that it will be a brief revisit, and there will not be too much that requires changing in the solution design.

I do feel that the waterfall model has been suitable for me in this project, as I have needed to spend a lot of time researching and obtaining new skills and knowledge. The waterfall approach leaves me plenty of time to do this as I am not implementing before I have designed or testing before I have finished implementation. If I had adapted a more iterative approach I would have needed to learn even more upfront, to allow me to complete small chunks of the application in each iteration, meaning it would have been even longer before I was able to jump in and start implementing code. The waterfall approach has enabled me to learn and code in unison. Saying this, I do feel as if adopting an iterative lifecycle would have enabled me to feel like I have achieved more once I started coding, as although I feel like I have made a good amount of progress in the project, an iterative lifecycle would have enabled me to finish complete sections of the project early on, rather than completing everything in one project stage.

Assess risks to project completion

The risks that I have identified are shown below. I have applied a weighting to each risk to allow assessment of the risk severity and highlighted those which would have the most impact on project completion.

Risk No	Major Project Risks	Likelihood	Impact	Weighting	Mitigation
R01	Learning a new JavaScript framework will take longer than anticipated	M	H	5	Try to get going with coding as early in to learning as possible, and code along with the tutorial
R02	Product requirements may change	L	M	3	Ensuring that I am regularly checking the list of defined requirements as I am carrying out development
R03	'Feature creep' may occur as I begin coding and the application will become more complex than I anticipated	M	H	5	Ensure that the wireframes and requirements are referred to regularly throughout the development process, so that I am focused on the solution that has been proposed, and nothing beyond this.
R04	Hardware failure	L	L	2	Source code will be committed to source control, and other project files are saved to the Dropbox. Recovery should be swift if hardware fails.
R05	Struggles to set up development environment	L	M	3	Ensuring that I have credible and relevant information sources available to help with any issues, such as the Vue.js documentation and Stack Overflow.
R06	Falling behind in project schedule due to concurrent studying of another OU module	H	H	6	Ensure that the schedule includes time needed for commitments for my other module.
R07	Estimations and scheduling activities carried out for project are not accurate or achievable	L	H	4	Make sure estimates include a buffer so that any extra time that is needed for certain tasks can be made up during time set aside for easier tasks.
R08	Inaccessibility of chosen recipe API	L	H	4	As this is a 3 rd party provided service there is not much I can do to prevent any issues caused here. It might be helpful to create a file of dummy data, so that I can

Risk No	Major Project Risks	Likelihood	Impact	Weighting	Mitigation
					work 'offline' with this JSON if needed.
R09	Project scope becomes unmanageable	H	H	6	Do not spend time on any tasks that have not been specified as a requirement for the solution or have not been included in the project schedule. Always keep the task list in mind to ensure only meaningful tasks are completed.
R10	Illness	L	L	2	There should be enough slack in the schedule to allow for days missed due to illness, there is not much I can do to mitigate this risk.

Of the initial risks to project completion that I identified I would say the two that have had an impact on the project so far are R01 and R06, although at this time I am confident that the impact will not affect completion of the project as it has been planned. The scope of the project has slightly changed, with the addition of a few requirements once the implementation of the project began. I do not however think this change adds to the risk of the project not begin fully completed, as the changes are small.

Review of personal development

In general, I am pleased with the personal development I've made during the project and particularly with the additional development skills that I have attained so far. I am particularly pleased with the skills I have developed in Vue.js and feel that by the end of the project I will have skills in this area that I will be able to apply in the work place.

I have realised that I learn much more effectively from practically using something, rather than purely reading about it, and will ensure in the future that I do not get lost in documentation when exploring new technologies or concepts. The best way for me to learn about these new things is to use them.

I am still struggling with my time management skills, but this project has definitely aided me in giving me good practical experience of issues that occur due to poor time management and has provided good lessons for me to learn from in future endeavours. I do however believe that during the course of the project my shortcomings in this area have been reduced, in part due to the rigorous planning that I have had to do as part of the project. These practises have given me good tools to carry forward into future projects, although there is still room for improvement going forwards in this area.

The project has also improved my problem-solving skills, as I have been able to effectively solve issues I have encountered during the project. I have been able to quickly identify useful sources of information and obtain the pertinent information from these sources to enable me to solve issues

quickly. I have also been able to effectively utilise the module forums and social media groups when I needed clarification on issues that I have encountered throughout the project.

One area that I still feel I could improve in is my communication skills. I have conducted the project mainly in isolation, except for discussions with potential users during the requirements gathering stage. I have mainly used online sources and forums to aid me when anything has been unclear to me during the project, and it may have been more efficient to clarify these issues with my tutor.

Word count: 4855

References

- Kirk, A & Scott, P (2018) *The Guardian* [Online]. Available at <https://www.telegraph.co.uk/news/2018/01/02/no-time-leftovers-astonishing-scale-food-waste-uk-around-world/> (Accessed 28 February 2019)
- Stern, D. (2019). *Vue.js: Big Picture*. [video] Available at: <https://app.pluralsight.com/player?course=vuejs-big-picture&author=daniel-stern&name=a9d0eb37-1a2c-413f-a37c-7a9ef277f353&clip=0&mode=live> [Accessed 16 Mar. 2019].
- Ward, C. (2017) *Jump Start Responsive Web Design*, Victoria, SitePoint
- Pluralsight.com. (n.d.). *Teach | Pluralsight*. [online] Available at: <https://www.pluralsight.com/teach> [Accessed 16 Apr. 2019].
- Edamam. (2019) *Recipe Search API Documentation*. [online] Available at: <https://developer.edamam.com/edamam-docs-recipe-api> [Accessed 11 Apr. 2019].
- Holland, B. (2018). *Visual Studio Code Can Do That? — Smashing Magazine*. [online] Smashing Magazine. Available at: <https://www.smashingmagazine.com/2018/01/visual-studio-code/> [Accessed 16 Apr. 2019].
- cli.vuejs.org. (n.d.). *Installation*. [online] Available at: <https://cli.vuejs.org/guide/creating-a-project.html#vue-create> [Accessed 01 Jun. 2019].
- cli.router.org. (n.d.). *Named Routes*. [online] Available at: <https://router.vuejs.org/guide/essentials/named-routes.html> [Accessed 15 Jun. 2019].
- npm (2019) *axios* [online] Available at <https://www.npmjs.com/package/axios> [Accessed 1 Jul 2019]
- w3cschools (n.d) *JavaScript encodeURIComponent() Function* [online] Available at https://www.w3schools.com/jsref/jsref_encodeuricomponent.asp [Accessed 1 Jul 2019]
- Post, T (2017) *Stack Overflow* [online] Available at <https://meta.stackoverflow.com/questions/280478/why-not-w3schools-com> [Accessed 1 Jul 2019]
- Mozilla (n.d) *Mozilla eb docs* [online] Available at https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/encodeURIComponent [Accessed 1 Jul 2019]
- Ambler, S.W. (n.d.), 'Customer Tests as Requirements Artifacts'; available at <http://www.agilemodeling.com/artifacts/acceptanceTests.htm> (accessed 06/06/2019).

Appendices

Appendix A

Interviewer: Hi [name]! Just to make you aware of the rights as a participant in my interview, you have the right to terminate at any time, your data will be used by me for my EMA and a transcript included in my TMA02 document, that will be viewed by my tutor.

Interviewee: Ok.

Interviewer: First of all, do you currently use any recipe finding applications or websites?

Interviewee: I do use websites that I find via Google, but I never stick to one.

Interviewer: So you literally just search for recipes using the search engine rather than a particular site?

Interviewee: Yeah, just because there's never all the things that I want really on one website. If there was one that I would say I go to most it's probably the BBC food website

Interviewer: So when you say there's not one that has all the things you want in one place, what are all the things you want?

Interviewee: Recipes for different meals like dinner, breakfast etc, also baking ... like a little bit of everything. You find websites for baking, but then can't use the same for other recipes.

Interviewer: So it's not that the interface of the application that doesn't offer you what you want, it's just they don't have the recipes you want?

Interviewee: Uh-huh

Interviewer: Ok ... So you literally just link through to a recipe from Google, you don't use a search function on the websites?

Interviewee: Uh yeah, I do occasionally, it depends. If I found a website, and I liked it, then I would use the search function on that website again, but I don't always find one that's useful.

Interviewer: So what features do you find useful in a recipe search, if you were to use one?

Interviewee: There is one, I can't remember which one it is, but when you use it on a mobile it has little tabs, so you have the ingredients you need on a tab, and then you just click on the next tab at the top, then it's got the method bullet pointed on it.

Interviewer: So that's better for mobile view?

Interviewee: Uh-huh

Interviewer: What about when you're actually searching? So by ingredient, or recipe name or ...

Interviewee: Um, I think sometimes it hard to search by ingredient, so I usually just search by a name, but I think if you could search by ingredients that would be more useful.

Interviewer: Do you mean it's hard because the website or app doesn't do it easily

Interviewee: It just doesn't recognise what you want

Interviewer: OK. So if you could make yourself a recipe finder that did everything that you want, what would be the features that you would include as a priority?

Interviewee: As we were just saying, search ... so you could just put in ... like if you opened your cupboard and you had chicken and noodles and a few random things then you could type in what you had and then it would give you a recipe, not necessarily with everything you had, but roughly based on then ingredients that you have.

Interviewer: Cool, and what about the way it would return you your results, would you like just a list, or pictures ...

Interviewee: Definitely pictures. I wouldn't ever make anything if there wasn't a picture.

Interviewer: OK,

Interviewee: A list with a picture is fine.

Interviewer: And you said on mobile that the tabs on that particular site are good for mobile, how about on like a normal computer monitor, would you want a different layout, or tabs,

Interviewee: Um, if it was a computer then I'd probably go for the layout like all on one page, so you can see it all at once. But I just think on a phone that that would be too small and you'd constantly be zooming in. But I think if you've got a bigger screen and you can see it all at once then it would be a lot easier.

Interviewer: What about any friends and family? DO you know if they use anything particular [for recipe finding], anyone got a favourite that you know of?

Interviewee: Yeah they do use them, but I think they're just a bit the same as me, like there's never one that they can always find stuff on, so they just go everywhere.

Interviewer: Cool ... so in your household, what happens with food that's going out of date, or has maybe just gone passed the sell-by date?

Interviewee: It all gets thrown away. We throw away so much food

Interviewer: And can you think of anything that would encourage you to not throw it away?

Interviewee: If I knew that it was gonna go out of date, then I would use it, but sometimes it just sits in the fridge doesn't it, you buy something and you forget it's in there, and then you end up not using it, and it goes in the bin. So if you knew that that was coming up then I'd definitely use it up.

Interviewer: Cool, that's it then, thank you so much for your participation

Appendix B

Requirements

1. Allow viewing of a virtual store cupboard.
2. Allow the addition of an item to a virtual store cupboard.
3. Allow the deletion of an item from a virtual store cupboard.
4. Add store cupboard ingredients to ingredients list when searching
5. Remove an item or quantity of an item from the store cupboard once it's been used in a recipe.
6. Allow addition of multiple ingredients to a search a recipe database
7. Display a list of available recipes based on selected ingredients
8. *Allow sorting and filtering of recipes*
9. Show method and ingredients list of a selected recipe
10. Display information about food safety
11. Allow searching of information related to food safety

12. Display a loading indicator while making API requests
13. Allow the user to choose whether to include store cupboard items in the recipe search

Appendix C

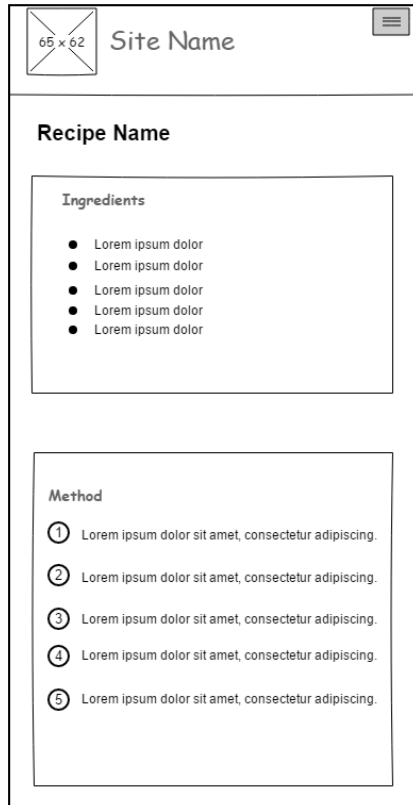


Figure 1. View Recipe mobile view

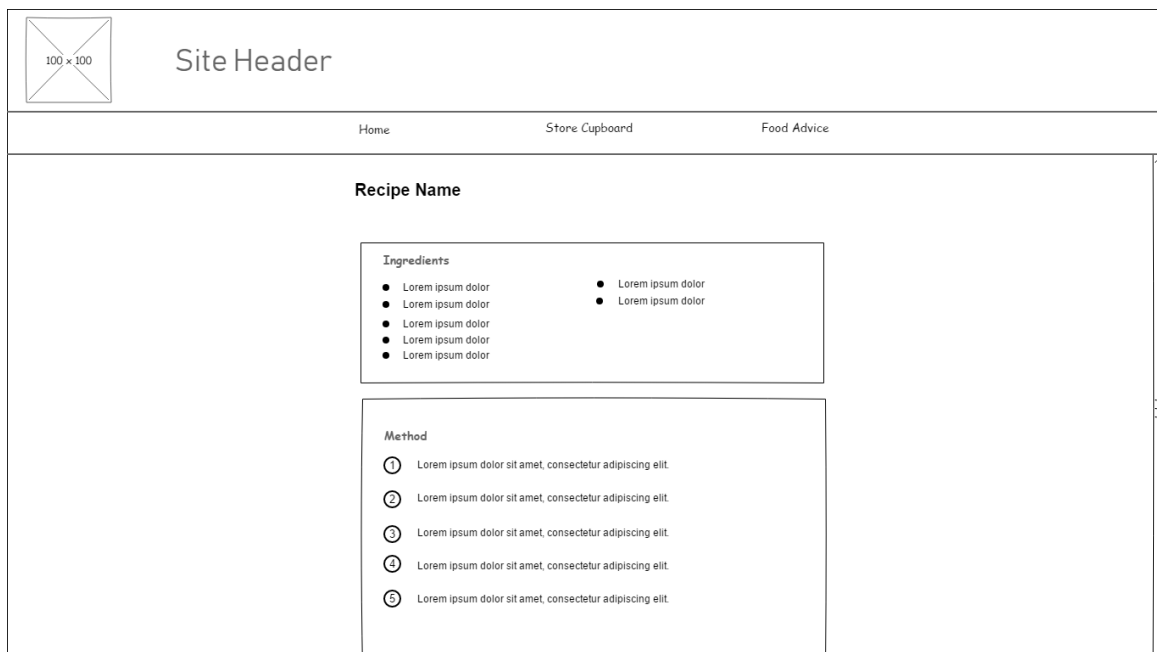




Figure 2. View Recipe desktop view

The mobile view of the Store Cupboard interface features a header with a 65 x 62 placeholder icon, the text "Site Name", and a menu icon. Below the header is a section titled "Add ingredient" containing a text input field and a blue plus icon. Underneath is a "Store Cupboard" section with a list of three items: "Item 1", "Item 2", and "Item 3", each followed by a square button containing an "X".


Figure 3. Store Cupboard mobile view

The desktop view of the Store Cupboard interface has a "Site Header" with a 100 x 100 placeholder icon. Below the header is a navigation bar with three links: "Home", "Store Cupboard", and "Food Advice". The main content area includes a section titled "Add store cupboard item" with a text input field and an "Add" button. Below this is a "Store Cupboard" section with a list of three items: "Item 1", "Item 2", and "Item 3", each followed by a square button containing an "X".

Figure 4. Store Cupboard desktop view

 Site Name

Add ingredient




Ingredient 1 ☐


Ingredient 2 ☐

Ingredient 3 ☐

Search

Results

Recipe Title

Recipe Title



Recipe Title

Figure 5. Recipe Search mobile view

 Site Header

HomeStore CupboardFood Advice

Add ingredient


Add


Ingredient 1 ☐ Ingredient 3 ☐

Ingredient 2 ☐

Search

Results

Recipe Title

Recipe Title


Recipe Title

Figure 6. Recipe Search desktop view

Appendix D

Acceptance tests

Test ID	T001
Description	Test user can add Store Cupboard item to store cupboard
Setup	Navigate to 'Store Cupboard' page
Instructions	Enter 'garlic' as a value in input field, then click Add button
Expected Results	A new entry for 'garlic' should appear in listed store cupboard ingredients. If 'garlic' already exists in the ingredients list then an error should be displayed

Test ID	T002
Description	Test user can view Store Cupboard
Setup	Add a new item to the store cupboard if it is currently empty
Instructions	From 'Home', click 'Store Cupboard link.
Expected Results	If the user has items in the Store Cupboard, a list of these items should be displayed. If not, a message should be displayed indicating as such

Test ID	T003
Description	Test user can delete an item from the Store Cupboard
Setup	Add a new item to the store cupboard if it is currently empty
Instructions	From 'Home', click 'Store Cupboard link. Select an existing item in the Store Cupboard list and click the 'Delete' button, indicated with a 'X'
Expected Results	The deleted item should be removed from the Store Cupboard list displayed

Appendix E

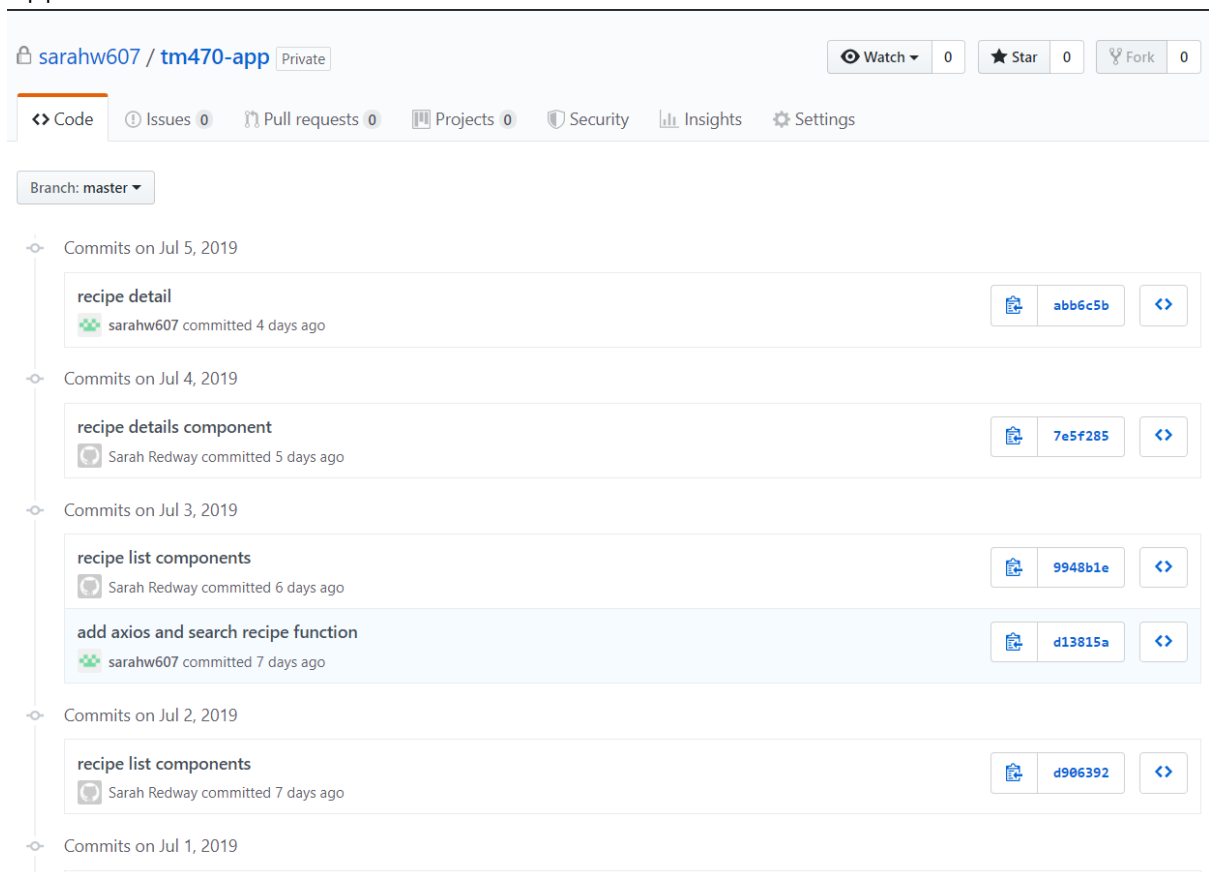
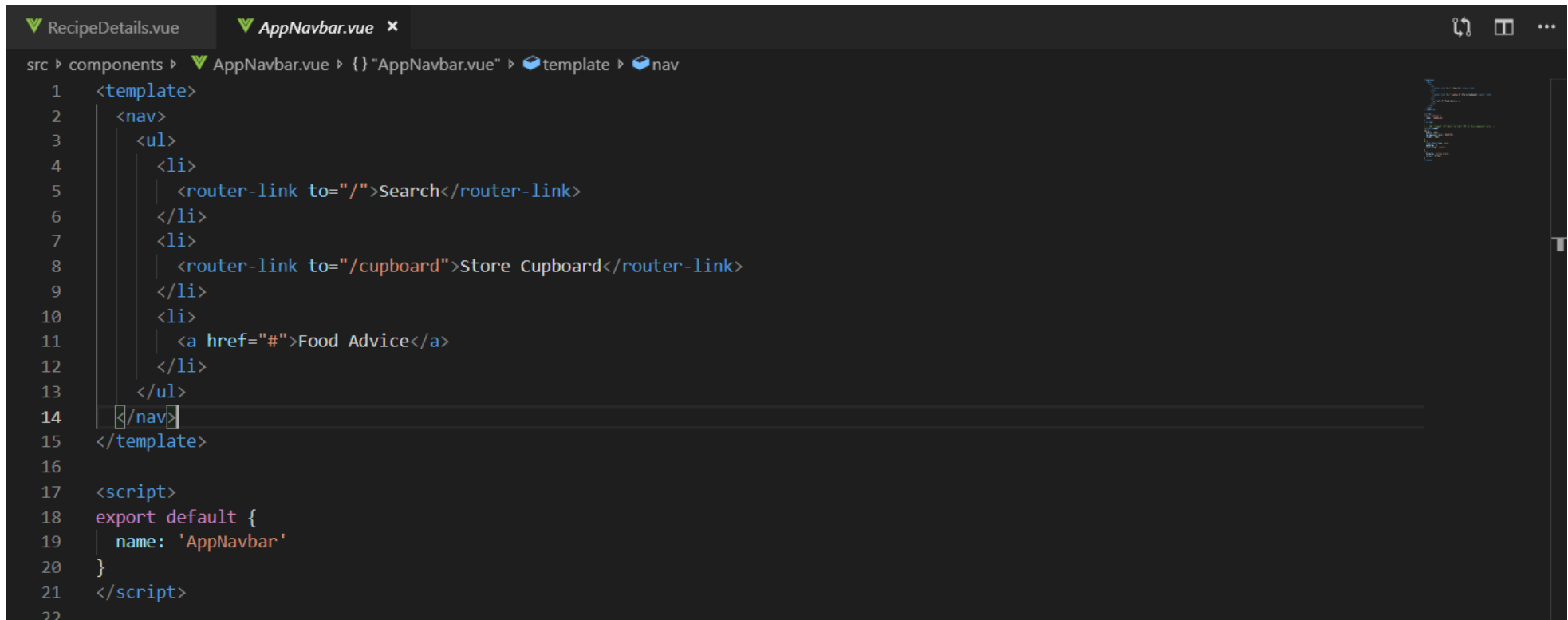


Figure 7 Git commit history

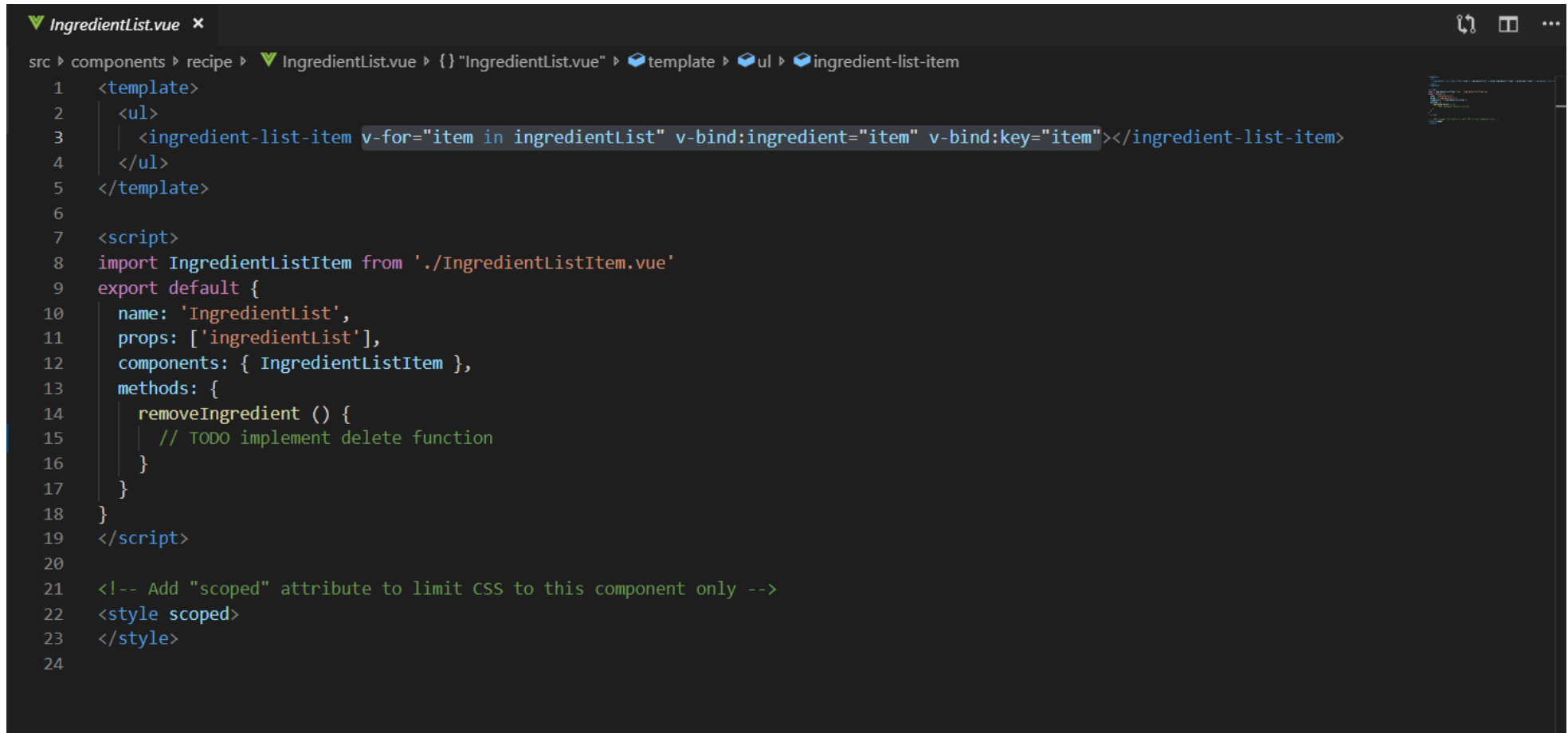
Appendix F



```
src ▸ components ▸ ▾ AppNavbar.vue ▸ {} "AppNavbar.vue" ▸ template ▸ nav
1  <template>
2    <nav>
3      <ul>
4        <li>
5          <router-link to="/">Search</router-link>
6        </li>
7        <li>
8          <router-link to="/cupboard">Store Cupboard</router-link>
9        </li>
10       <li>
11         <a href="#">Food Advice</a>
12       </li>
13     </ul>
14   </nav>
15 </template>
16
17 <script>
18 export default {
19   name: 'AppNavbar'
20 }
21 </script>
22
```

Figure 8 Application navigation component

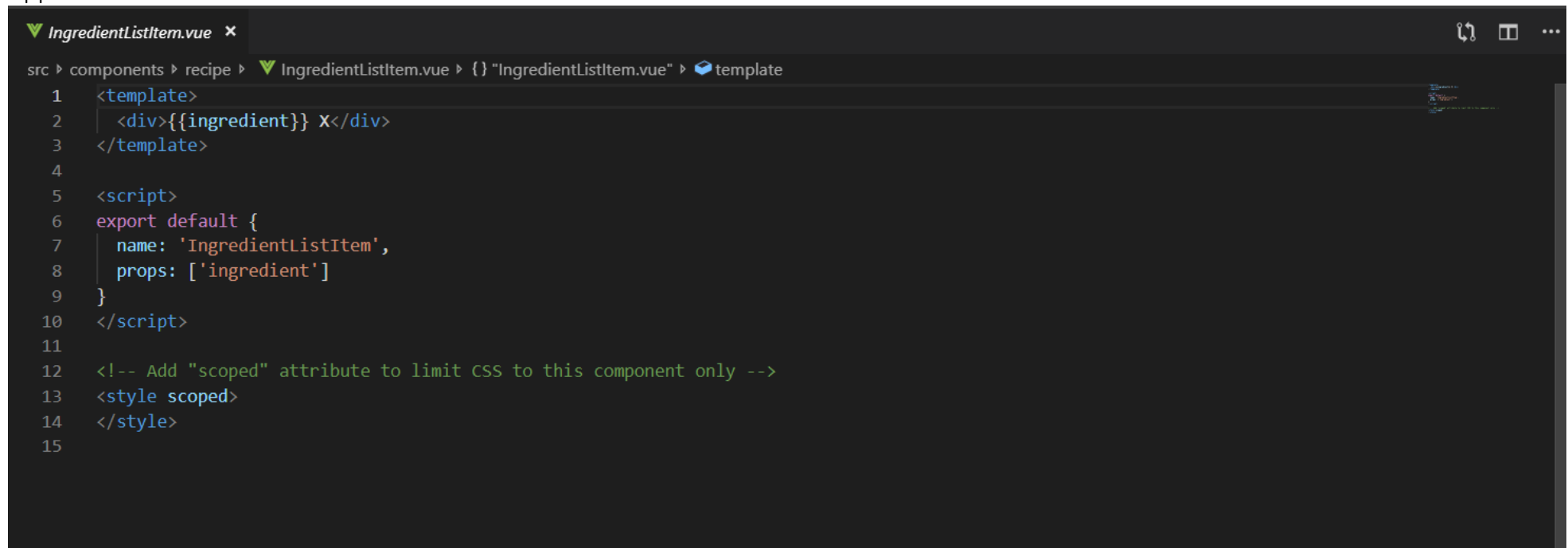
Appendix G



```
IngredientList.vue x
src > components > recipe > IngredientList.vue > {} "IngredientList.vue" > template > ul > ingredient-list-item
1 <template>
2   <ul>
3     <ingredient-list-item v-for="item in ingredientList" v-bind:ingredient="item" v-bind:key="item"></ingredient-list-item>
4   </ul>
5 </template>
6
7 <script>
8 import IngredientListItem from './IngredientListItem.vue'
9 export default {
10   name: 'IngredientList',
11   props: ['ingredientList'],
12   components: { IngredientListItem },
13   methods: {
14     removeIngredient () {
15       // TODO implement delete function
16     }
17   }
18 }
19 </script>
20
21 <!-- Add "scoped" attribute to limit CSS to this component only -->
22 <style scoped>
23 </style>
24
```

Figure 9 Ingredients List Component Code

Appendix H

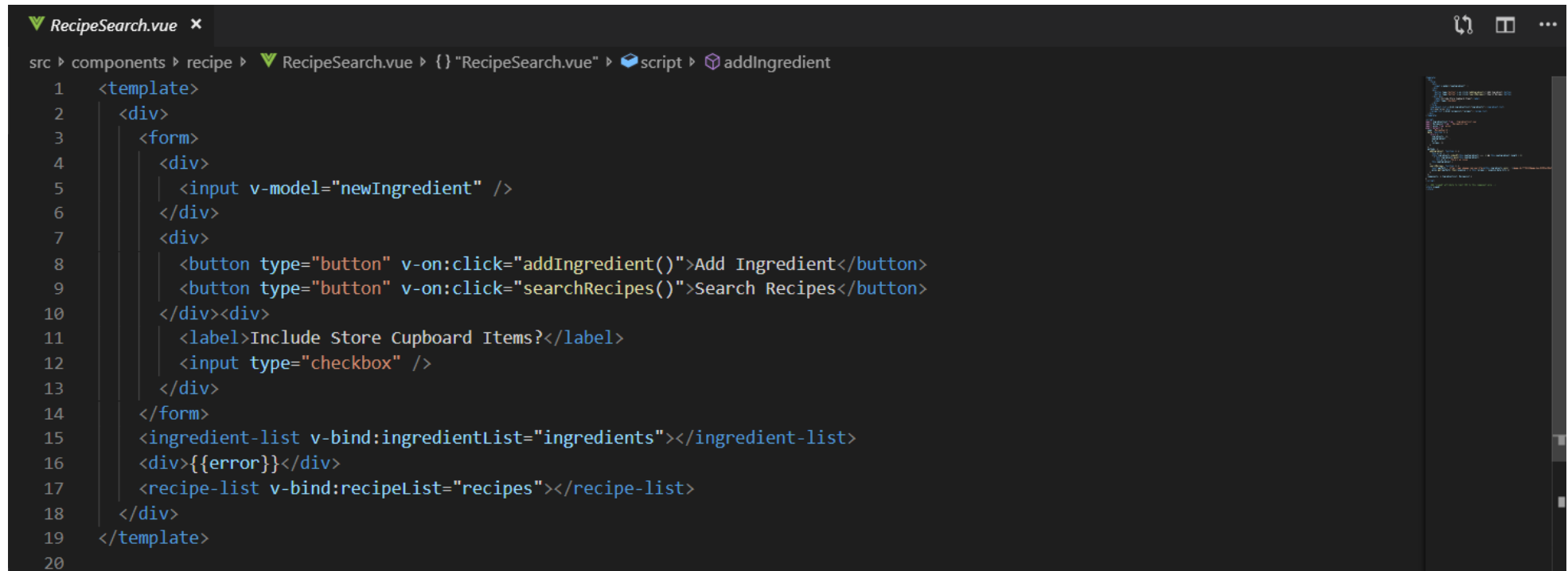


The screenshot shows a code editor with a dark theme. The file explorer on the left shows the path: src > components > recipe > IngredientListItem.vue. The editor displays the following code:

```
1 <template>
2   <div>{{ingredient}} X</div>
3 </template>
4
5 <script>
6 export default {
7   name: 'IngredientListItem',
8   props: ['ingredient']
9 }
10 </script>
11
12 <!-- Add "scoped" attribute to limit CSS to this component only -->
13 <style scoped>
14 </style>
15
```

Figure 10 Ingredients List Item Component

Appendix I

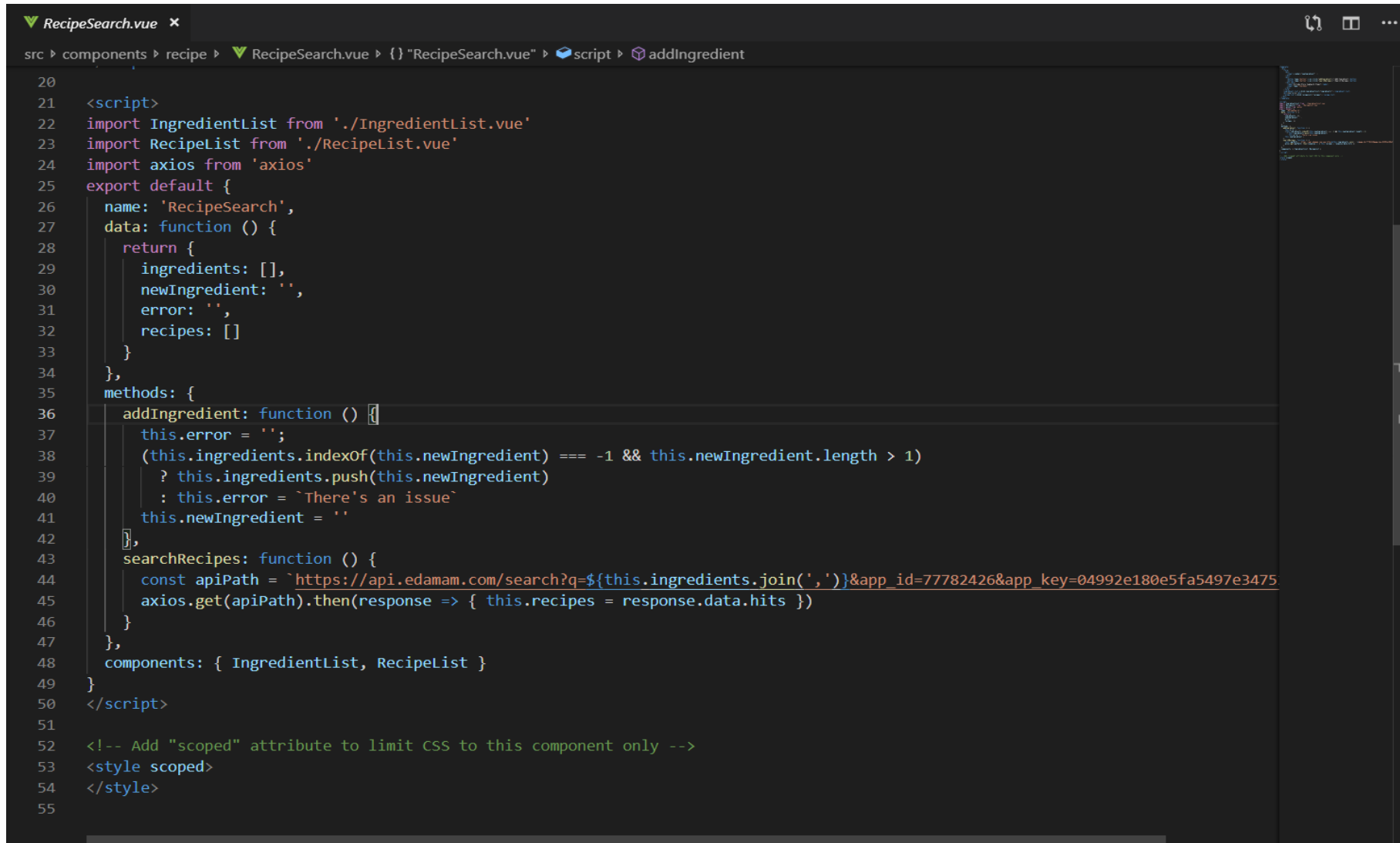


The image shows a code editor window with a dark theme. The file name 'RecipeSearch.vue' is visible in the top left. The breadcrumb navigation shows the path: 'src > components > recipe > RecipeSearch.vue > {} "RecipeSearch.vue" > script > addIngredient'. The code is a Vue.js template with line numbers 1 through 20 on the left. The template structure includes a form with an input field, two buttons, a checkbox, and two list components. There is an error placeholder in the template.

```
1 <template>
2   <div>
3     <form>
4       <div>
5         <input v-model="newIngredient" />
6       </div>
7       <div>
8         <button type="button" v-on:click="addIngredient()">Add Ingredient</button>
9         <button type="button" v-on:click="searchRecipes()">Search Recipes</button>
10      </div><div>
11        <label>Include Store Cupboard Items?</label>
12        <input type="checkbox" />
13      </div>
14    </form>
15    <ingredient-list v-bind:ingredientList="ingredients"></ingredient-list>
16    <div>{{error}}</div>
17    <recipe-list v-bind:recipeList="recipes"></recipe-list>
18  </div>
19 </template>
20
```

Figure 11 Recipe Search Template code

Appendix J



```
RecipeSearch.vue x
src > components > recipe > RecipeSearch.vue > {} "RecipeSearch.vue" > script > addIngredient

20
21 <script>
22 import IngredientList from './IngredientList.vue'
23 import RecipeList from './RecipeList.vue'
24 import axios from 'axios'
25 export default {
26   name: 'RecipeSearch',
27   data: function () {
28     return {
29       ingredients: [],
30       newIngredient: '',
31       error: '',
32       recipes: []
33     }
34   },
35   methods: {
36     addIngredient: function () {
37       this.error = '';
38       (this.ingredients.indexOf(this.newIngredient) === -1 && this.newIngredient.length > 1)
39         ? this.ingredients.push(this.newIngredient)
40         : this.error = `There's an issue`
41       this.newIngredient = ''
42     },
43     searchRecipes: function () {
44       const apiPath = `https://api.edamam.com/search?q=${this.ingredients.join(',')}&app_id=77782426&app_key=04992e180e5fa5497e3475`
45       axios.get(apiPath).then(response => { this.recipes = response.data.hits })
46     }
47   },
48   components: { IngredientList, RecipeList }
49 }
50 </script>
51
52 <!-- Add "scoped" attribute to limit CSS to this component only -->
53 <style scoped>
54 </style>
55
```

Figure 12 Recipe Search code including API integration

Appendix K

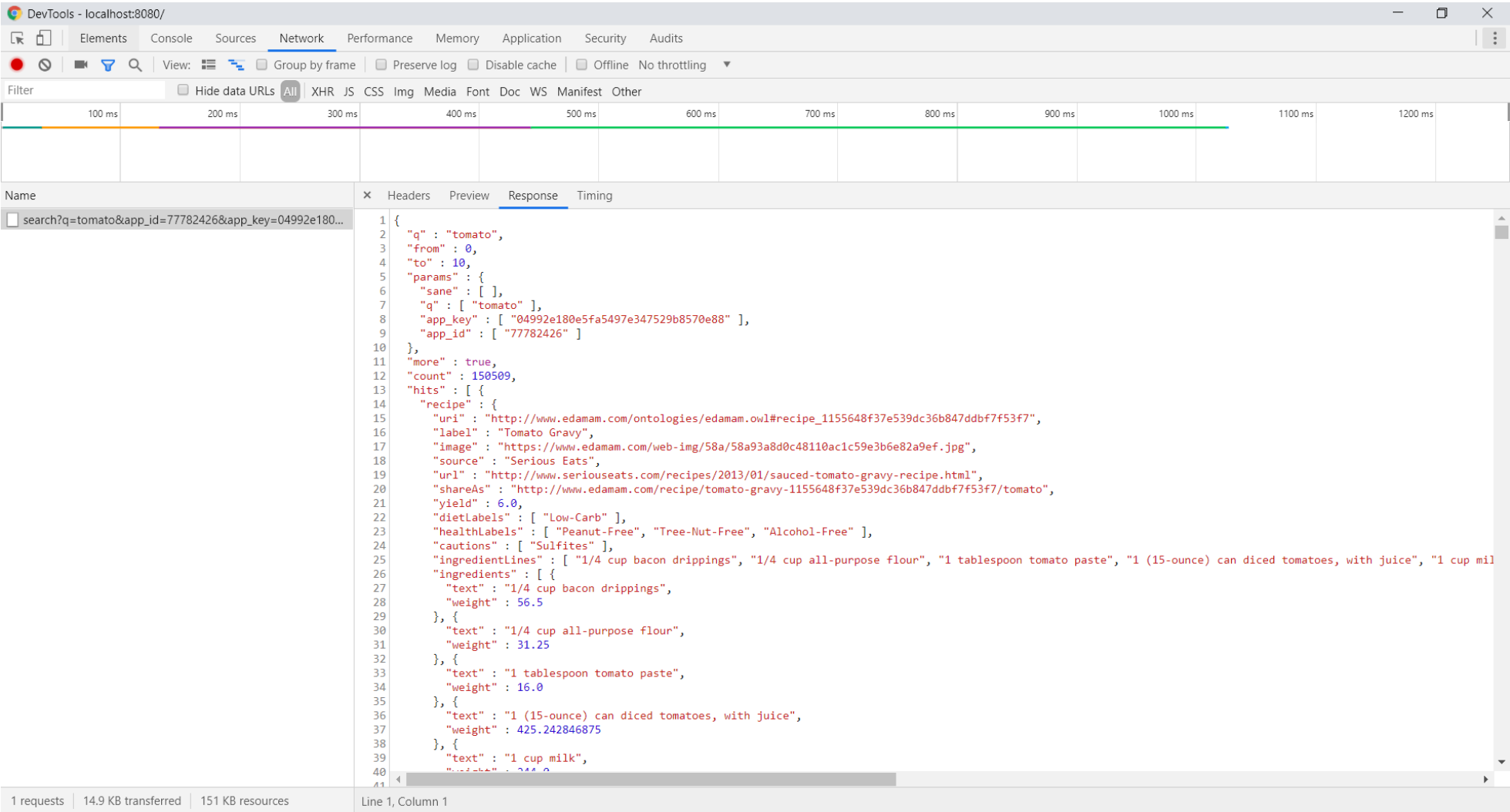


Figure 13 Response from Edamam Recipe Search API

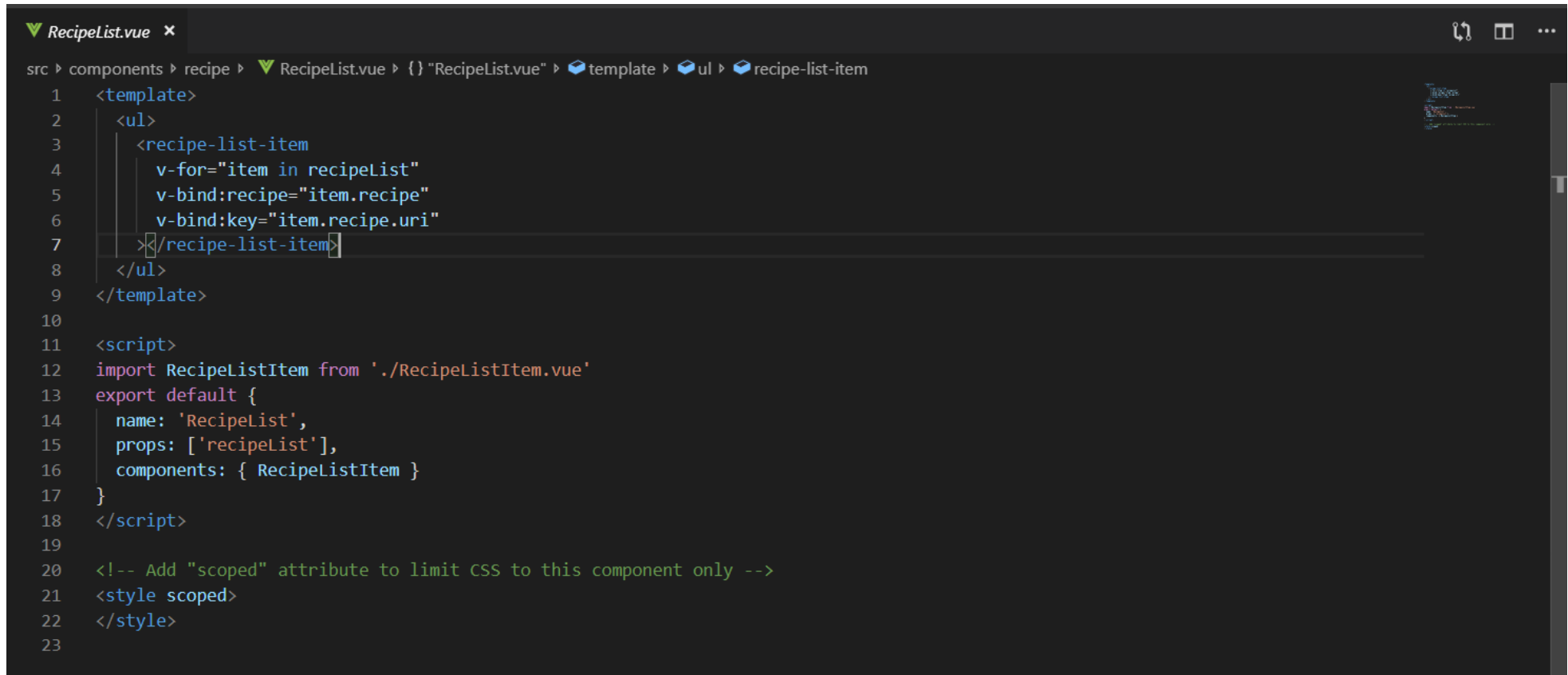
Appendix L



The image shows a code editor window with a dark theme. The file name 'RecipeListItem.vue' is visible in the top left. The breadcrumb navigation shows the path: 'src > components > recipe > RecipeListItem.vue'. The code is as follows:

```
1 <template>
2   <div>
3     <router-link :to="{ name: 'RecipeDetails', params: { recipe: recipe.uri }}">{{recipe.label}}</router-link>
4   </div>
5 </template>
6
7 <script>
8 export default {
9   name: 'RecipeListItem',
10  props: ['recipe']
11 }
12 </script>
13
14 <!-- Add "scoped" attribute to limit CSS to this component only -->
15 <style scoped>
16 </style>
17
```

Figure 14 Recipe List Item Component code

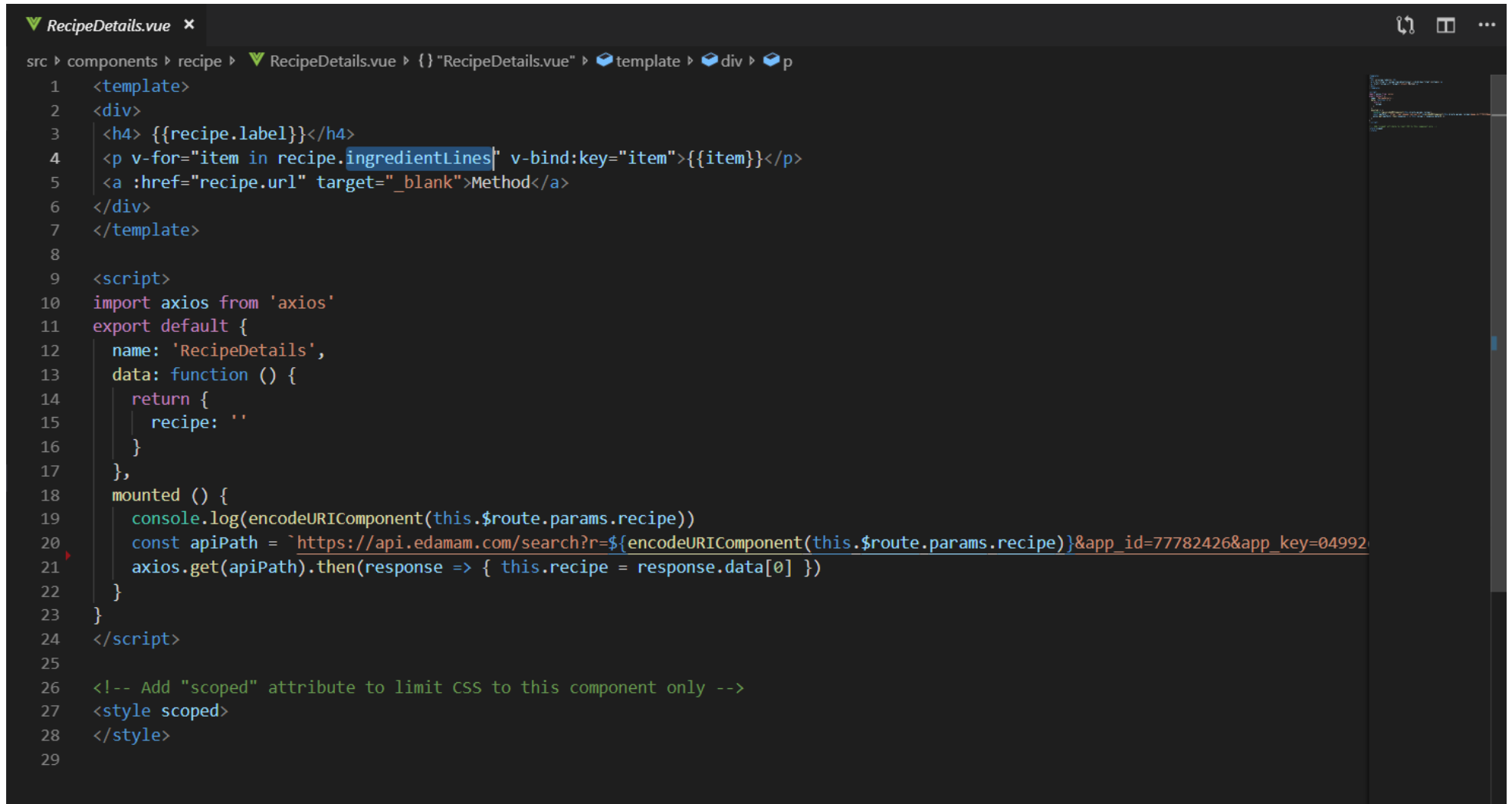


The image shows a code editor window with the file name 'RecipeList.vue' in the title bar. The breadcrumb navigation at the top indicates the file path: 'src > components > recipe > RecipeList.vue > {} "RecipeList.vue" > template > ul > recipe-list-item'. The code is written in Vue.js and is as follows:

```
1 <template>
2   <ul>
3     <recipe-list-item
4       v-for="item in recipeList"
5       v-bind:recipe="item.recipe"
6       v-bind:key="item.recipe.uri"
7     >/recipe-list-item</recipe-list-item>
8   </ul>
9 </template>
10
11 <script>
12 import RecipeListItem from './RecipeListItem.vue'
13 export default {
14   name: 'RecipeList',
15   props: ['recipeList'],
16   components: { RecipeListItem }
17 }
18 </script>
19
20 <!-- Add "scoped" attribute to limit CSS to this component only -->
21 <style scoped>
22 </style>
23
```

Figure 15 Recipe List component code

Appendix M



```
RecipeDetails.vue x
src ▸ components ▸ recipe ▸ RecipeDetails.vue ▸ {} "RecipeDetails.vue" ▸ template ▸ div ▸ p
1  <template>
2  <div>
3    <h4> {{recipe.label}}</h4>
4    <p v-for="item in recipe.ingredientLines" v-bind:key="item">{{item}}</p>
5    <a :href="recipe.url" target="_blank">Method</a>
6  </div>
7  </template>
8
9  <script>
10 import axios from 'axios'
11 export default {
12   name: 'RecipeDetails',
13   data: function () {
14     return {
15       recipe: ''
16     }
17   },
18   mounted () {
19     console.log(encodeURIComponent(this.$route.params.recipe))
20     const apiPath = `https://api.edamam.com/search?r=${encodeURIComponent(this.$route.params.recipe)}&app_id=77782426&app_key=04992`
21     axios.get(apiPath).then(response => { this.recipe = response.data[0] })
22   }
23 }
24 </script>
25
26 <!-- Add "scoped" attribute to limit CSS to this component only -->
27 <style scoped>
28 </style>
29
```

Figure 16 Recipe Details Component code

Appendix N

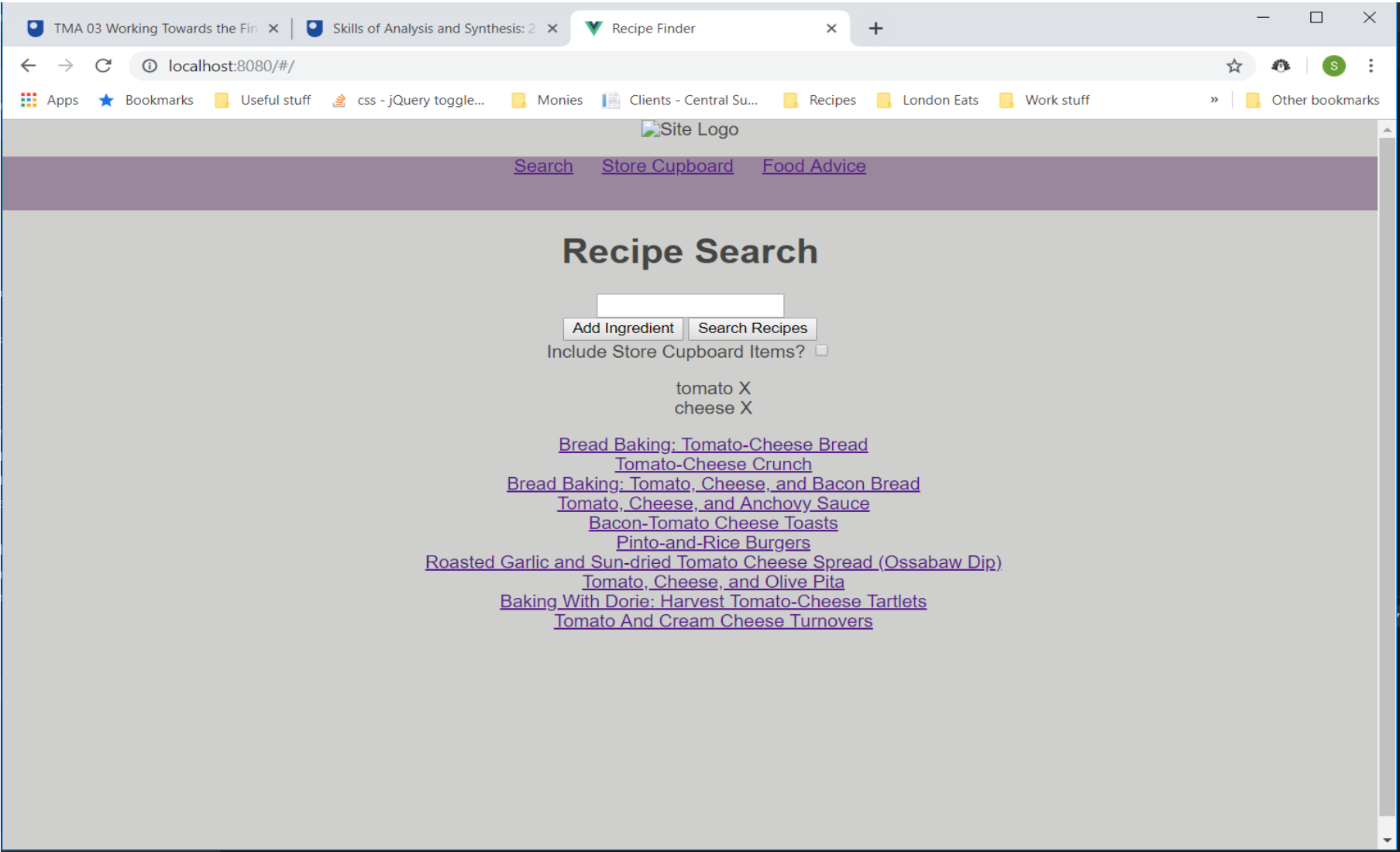


Figure 17 Recipe Search user interface

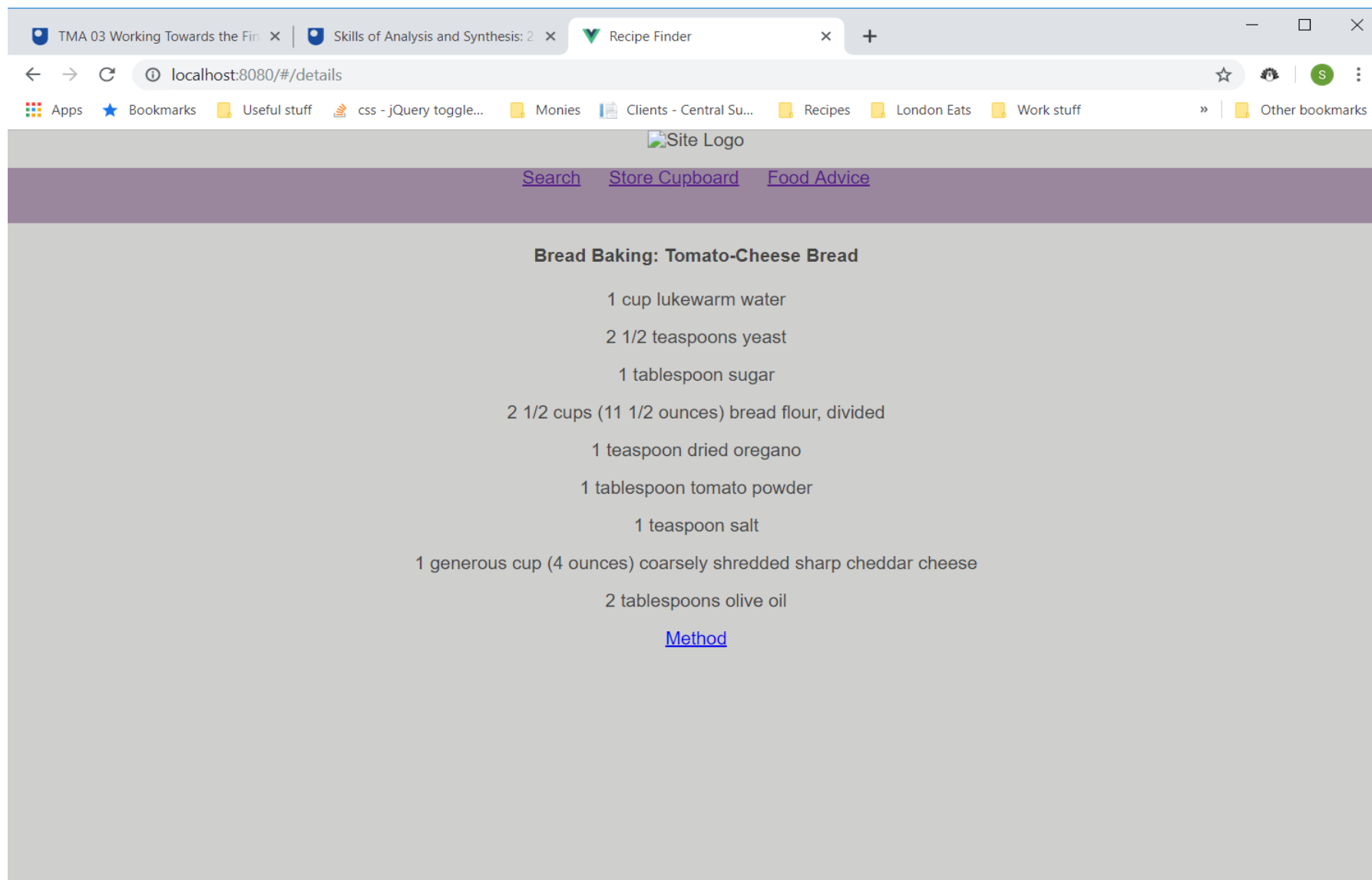


Figure 18 View Recipe user interface

Appendix O

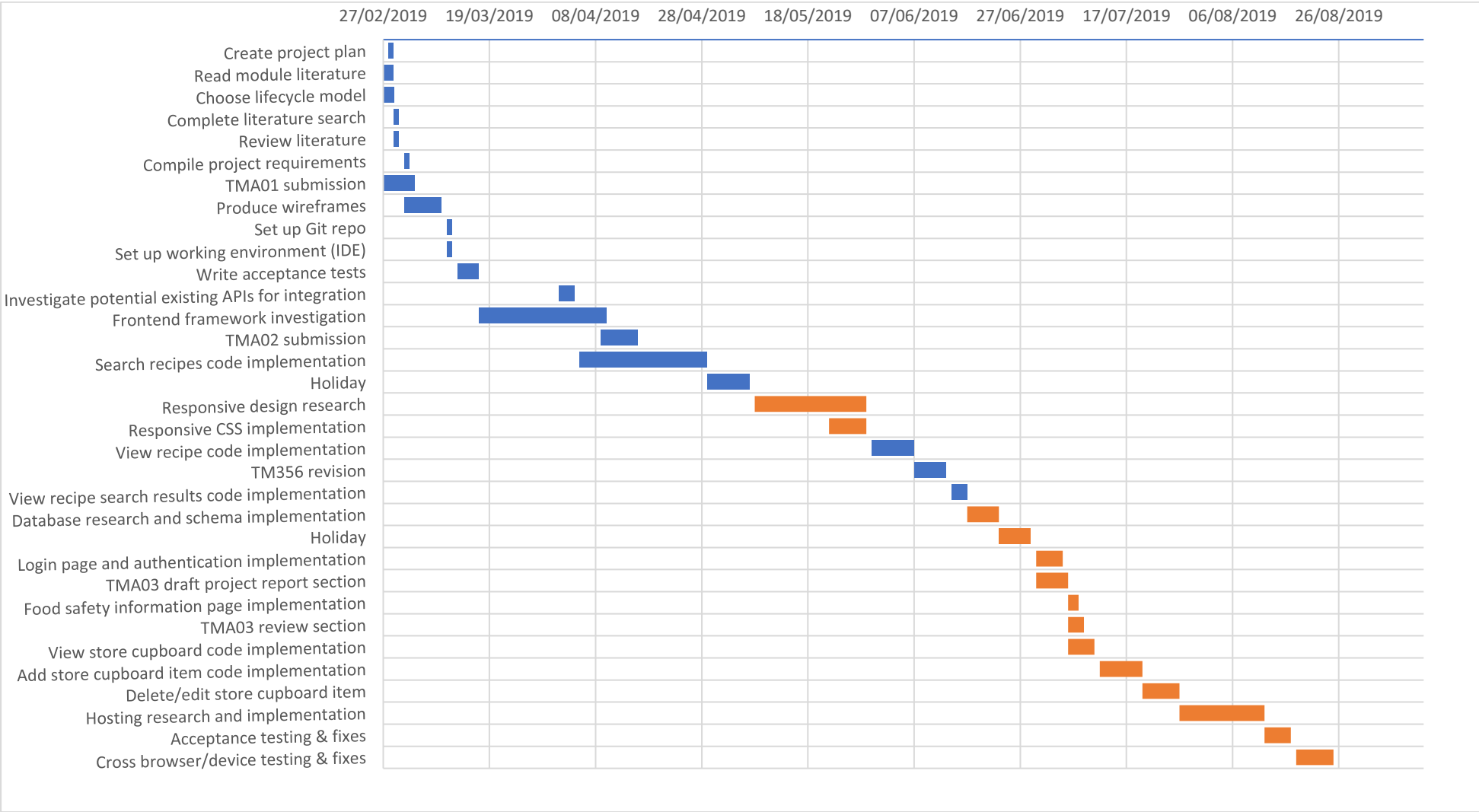


Figure 19 Gantt chart showing current project progress