**ReadMe Template Worksheet** 

Your ReadMes are the key deliverable that engineers will want to see as part of your job search. Each engineering team will look at different aspects of your ReadMe and repo. Some will go into the code itself and explore. Others will just want to see the showcase code snippets in the main ReadMe. Some will read the full thing to understand your approach, others will skim to specific sections.

It’s crucial that you cover all the different sections below to ensure that you’ve got the information for all engineers that check these out.

We regularly have employer partners discuss the importance of the ReadMes in what they’re looking for and why they interview the grads that they do - so don’t underestimate the importance of writing strong ReadMes!

For any pair or group project, you cannot share a ReadMe. These must be written independently to ensure that the engineers reading this understands **your** specific experience and approach.

It’s a good idea to start your ReadMes during the planning stage as this is the best way to get ahead and save time when it comes to finalising your first draft post-project. As you plan each aspect of your project, note down what you intend to do with screenshots of your plan and anything else you think would be useful, then when you execute this part in your code, you can adjust this part of your ReadMe as needed depending on how your process changed, or if it went as planned then you can leave it as it is.

**Make a copy of this document for each project you use throughout the course and fill in each section. Keep your copy in your Outcomes folder, so the team is able to add feedback into your docs.**

Once the content has been finalised by the Outcomes team, you can then upload these onto your GitHub repository later.

**Things To Consider:**

* That there are **no spelling mistakes in your ReadMe** - if you see a spelling error highlighted below, edit this.
  + Some engineers will reject applicants if their ReadMes are full of mistakes. From their perspective, if your ReadMes are full of mistakes, what is your code like…

* That your **technologies are capitalised correctly** - i.e JavaScript, jQuery, MongoDB

* That your **formatting is consistent** throughout - headers, indentation, full stops in bullets etc

* Any **hyperlink included works**

* That you **include images throughout** - code snippets, pictures of your planning stage, screenshots of the final project.
  + These can be still screenshots or gifs
  + This breaks up the text in your ReadMe and helps to keep the reader engaged

* That your ReadMe **sounds like you** - see this as an opportunity to showcase who you are to the engineering community and prospective employers.
  + Think back to the Personal Brand session and how employers want to **get a sense of who you are**. The content you write should sound as if you’re talking through your experience.

**ReadMe Sections**

**Description**

**Instructions**

*Here, give a short description of the project. It can be a couple of sentences where you discuss the point in time during the course that you completed it, the topic of the project and potentially the tech stack.*

**Insert your Description here:**

The project is a recipe website where users can discover cooking instructions for preparing meals at home. Additionally, users have the option to contribute new recipes, receive reviews, and share their culinary creations with both existing and new users.

**Deployment link**

**Instructions**

*Here include the information on where the deployed project can be found. If login details are needed to access the full project, make sure you include them.*

*If you have not yet deployed your project, you can add this in later.*

**Insert your Deployment link here:**

<https://tasty-pallete-3748066eacad.herokuapp.com/>

**Getting Started/Code Installation**

**Instructions**

*Explain how the reader accesses your code. Include a step by step approach.*

**Insert your Getting Started/Code Installation here:**

* bcrypt
* Axios
* bootstrap
* dotenv
* express
* jsonwebtoken
* mdb-react-ui-kit
* mongoose
* react-bootstrap
* react-router-dom
* sass
* seed
* vite
* @fortawesome/fontawesome-svg-core
* @fortawesome/free-brands-svg-icons
* @fortawesome/react-fontawesome
* @vitejs/plugin-react-swc

**Timeframe & Working Team (Solo/Pair/Group)**

**Instructions**

*Share the timeframe given for the project and whether you worked independently, in a pair, or in a group.*

*If you worked in a pair or group, include the names of the people you collaborated with. As a bonus, you can also provide links to their GitHub repo.*

**Insert your Timeframe & Working Team here:**

We were given 8 days to complete our group project. Initially, we conducted research and crafted a wireframe using Trello. We then utilised Trello to allocate tasks, identifying the areas we wanted to tackle first. As a collective effort, we began by working on the backend server, with everyone contributing together. Subsequently, we divided responsibilities for the frontend server among the team members.

Tamanna <https://github.com/ttamanna1>

Liz Alexander <https://github.com/lizalex7>

Ying Li <https://github.com/yingjod>

**Technologies Used**

**Instructions**

*List every technology you used to complete the project. This can be in one long list, or broken down into categories (Back End, Front End, Development Tools).*

**Insert your Technologies Used here:**

Technologies Used:

* Frontend: React.js, Bootstrap, React Router Dom, MDB React UI Kit
* Backend: Node.js, Express.js, MongoDB, Mongoose
* Styling: SASS, Bootstrap
* Other Tools: Axios, JSON Web Token, Bcrypt, Dotenv
* Build Tools: Vite, SASS Compiler
* Libraries: Font Awesome (Font Awesome SVG Core, Free Brands SVG Icons, React Font Awesome)
* Deployment: Heroku (for hosting the application)"

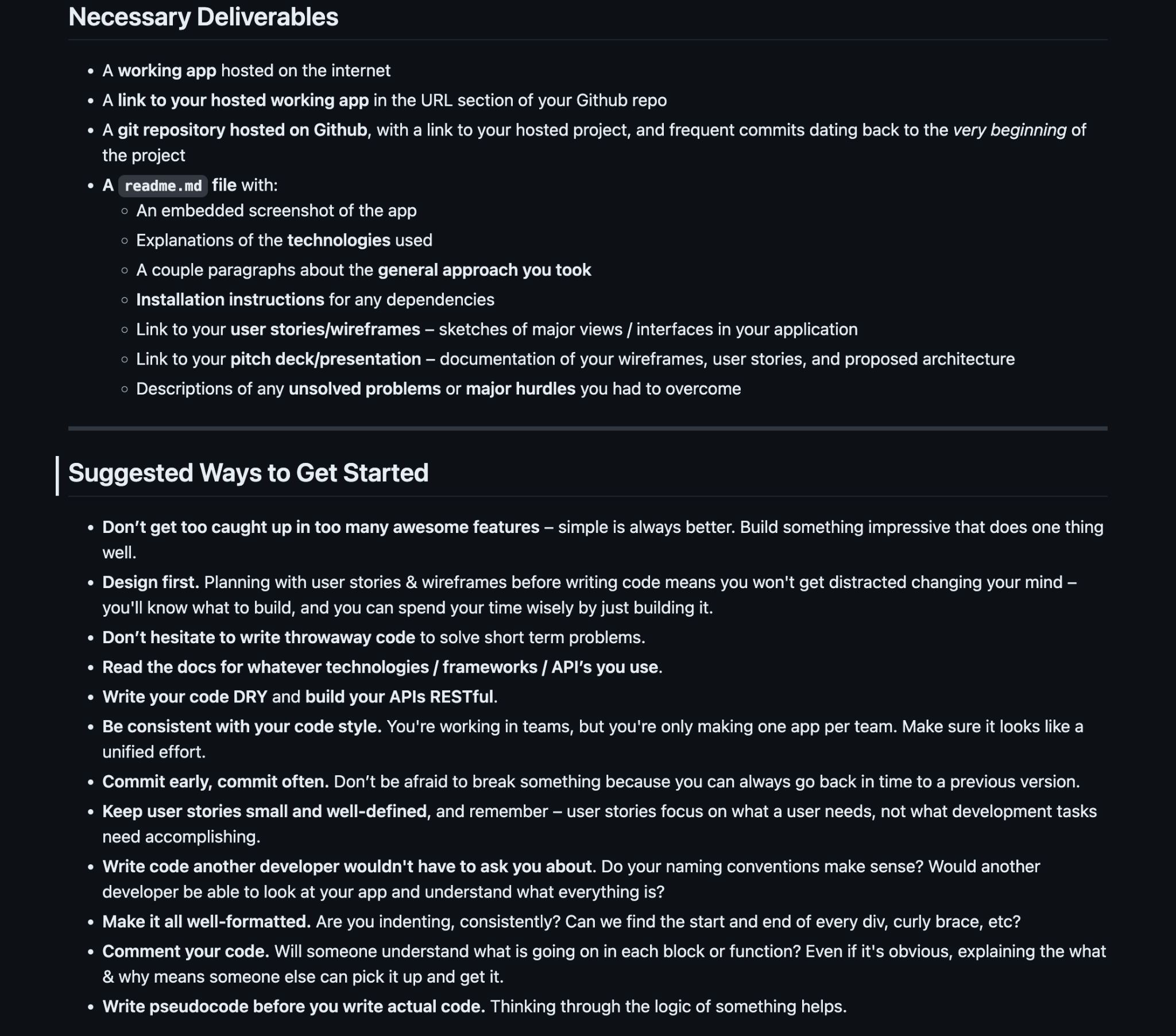
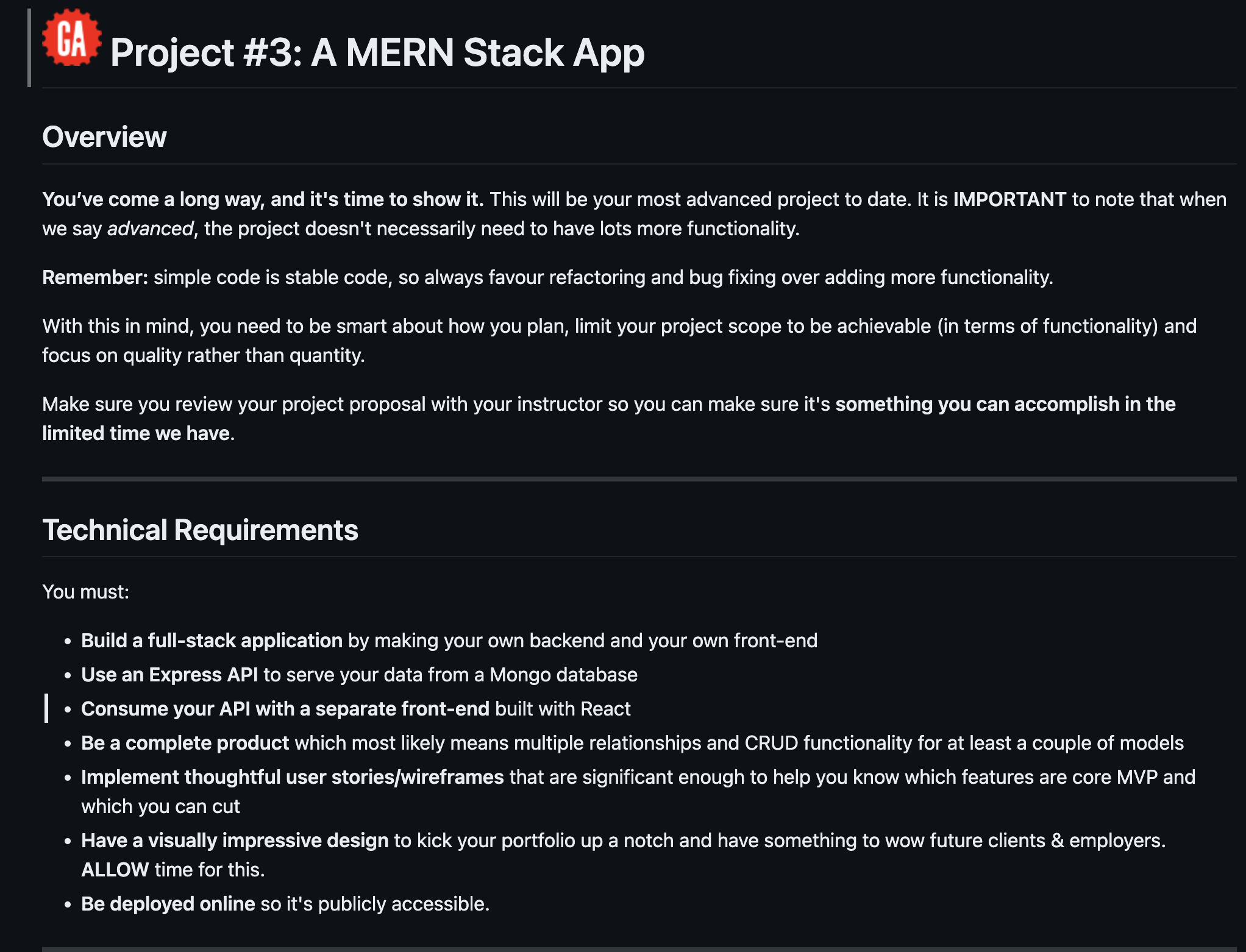
**Brief**

**Instructions**

*Include the brief set by your instructional team here. This sets the context of the project you were working towards and mimics briefs you will be set later in your future roles.*

*This can either be in bullets or in a paragraph.*

**Insert your Brief here:**



**Planning**

**Instructions**

*The planning stage is important, as all projects in your future roles will have detailed plans before any coding happens. It is a great experience to share with potential engineer employers, as this reflects real engineering team practices.*

*Start by explaining the initial steps you took in the project.*

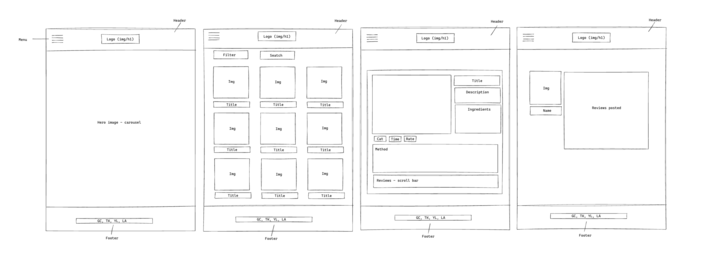
* ***Did you do any sketches****? If so, discuss this and include images.*
* ***Any wireframes of the front end and UI****? You did? Then explain this and include images.*
* ***Any ERDs****? Same here, explain and include images.*
* ***Use a project management tool to plan the sprint****? If so, talk through this - what tool did you use? How you allocated tickets/responsibilities, sprint timeline etc. Also include screenshots of this.*
* ***Any pseudocode****?*
* ***If it was a group or pair project*** *- Discuss who was designated which tasks. This is very important, as engineers want to understand who owned the different code elements when looking at a group project.*

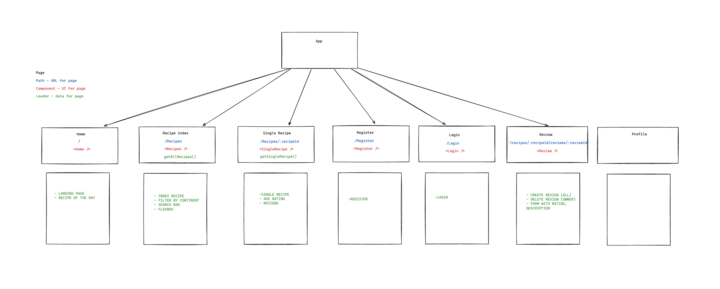
*For each project, review the above bullets and discuss every step you took in the planning stage, including the relevant images.*

*Not every project will include the above, but it’s important to discuss any of the bullets that you did implement.*

**Insert your Planning here:**

This was the wireframe our group and I created before commencing our project. We had the freedom to design it according to our preferences. Our wireframe was well-structured, outlining the various pages and routes we intended to incorporate.





**Build/Code Process**

**Instructions**

*The Build/Code Process will be the longest section of your ReadMe and will be most insightful to the engineers that review them. This is where you will discuss the steps you took to code the project.*

*You want to see your ReadMes as a way to walk the engineers through your approach and problem solving from the start of the project through to the end.*

*You'll need to include a minimum of 3-4 code snippets, highlighting code you're particularly proud of and these code snippets will have descriptions on what you did, how and why to set the context of the snippet you include. These explanations are important for the engineers, as they will want to understand what you did and the reasoning behind the steps you took.*

*You don't need to document every single thing you coded, but walk them through the key sections of the project build.*

*For any group project, you will just focus on your contributions.*

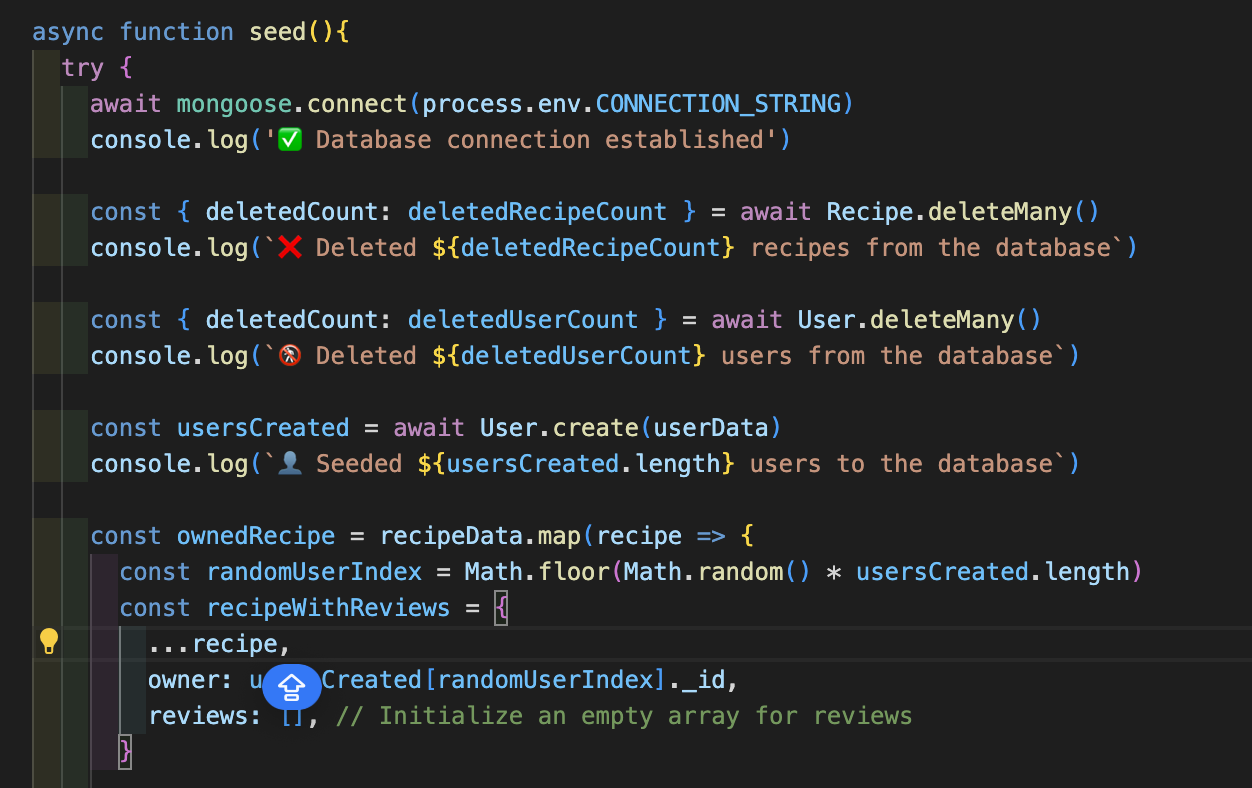
*Some people will document the build/code process by discussing the key stages they worked on. Others will do a day by day guide. It’s entirely up to you how you structure this, as long as you discuss all the key things above.*

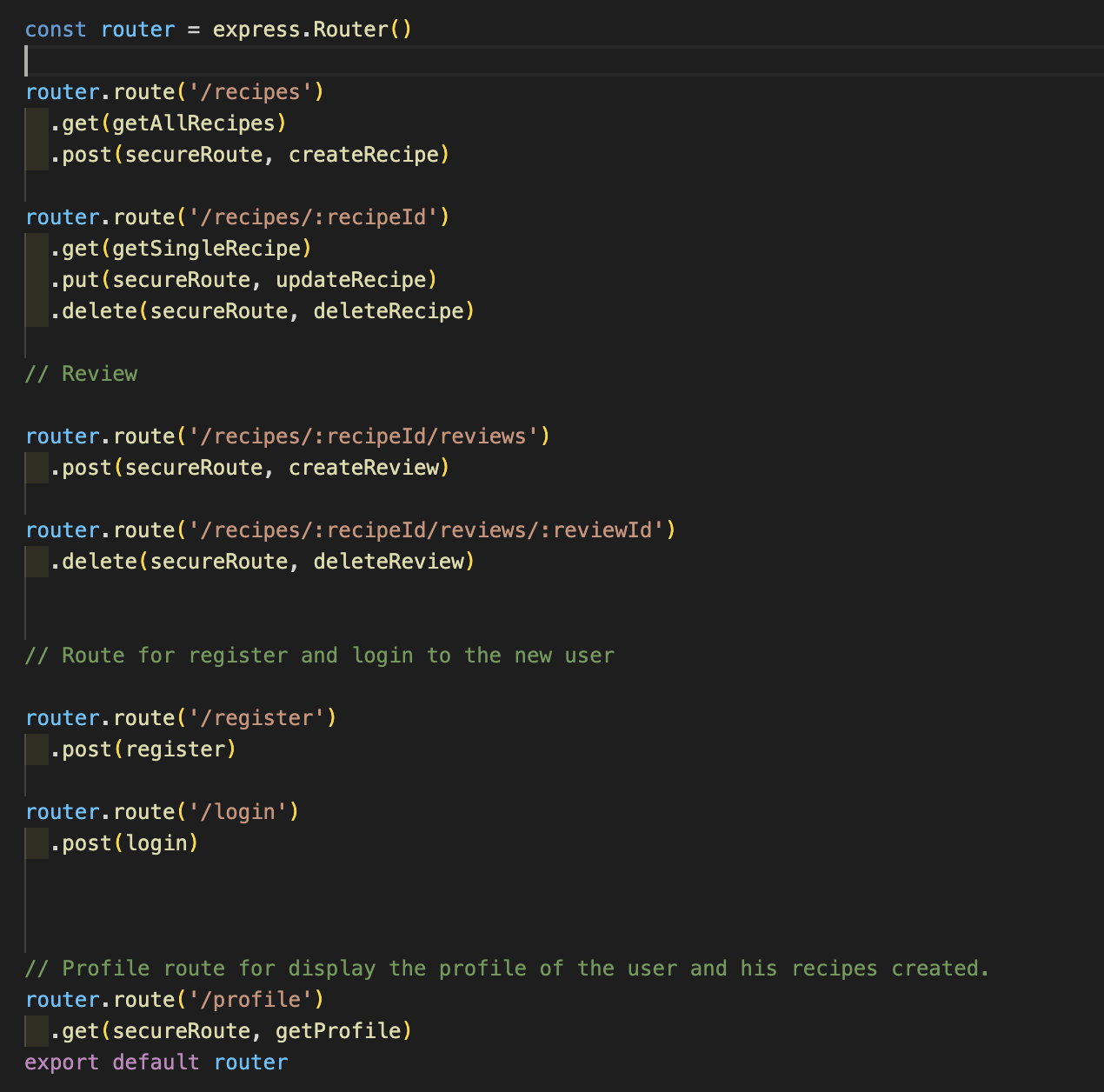
**Insert your Build/Code Process here:**

Day 1 – We began by creating the wireframe and conducting research to gather names and recipes.

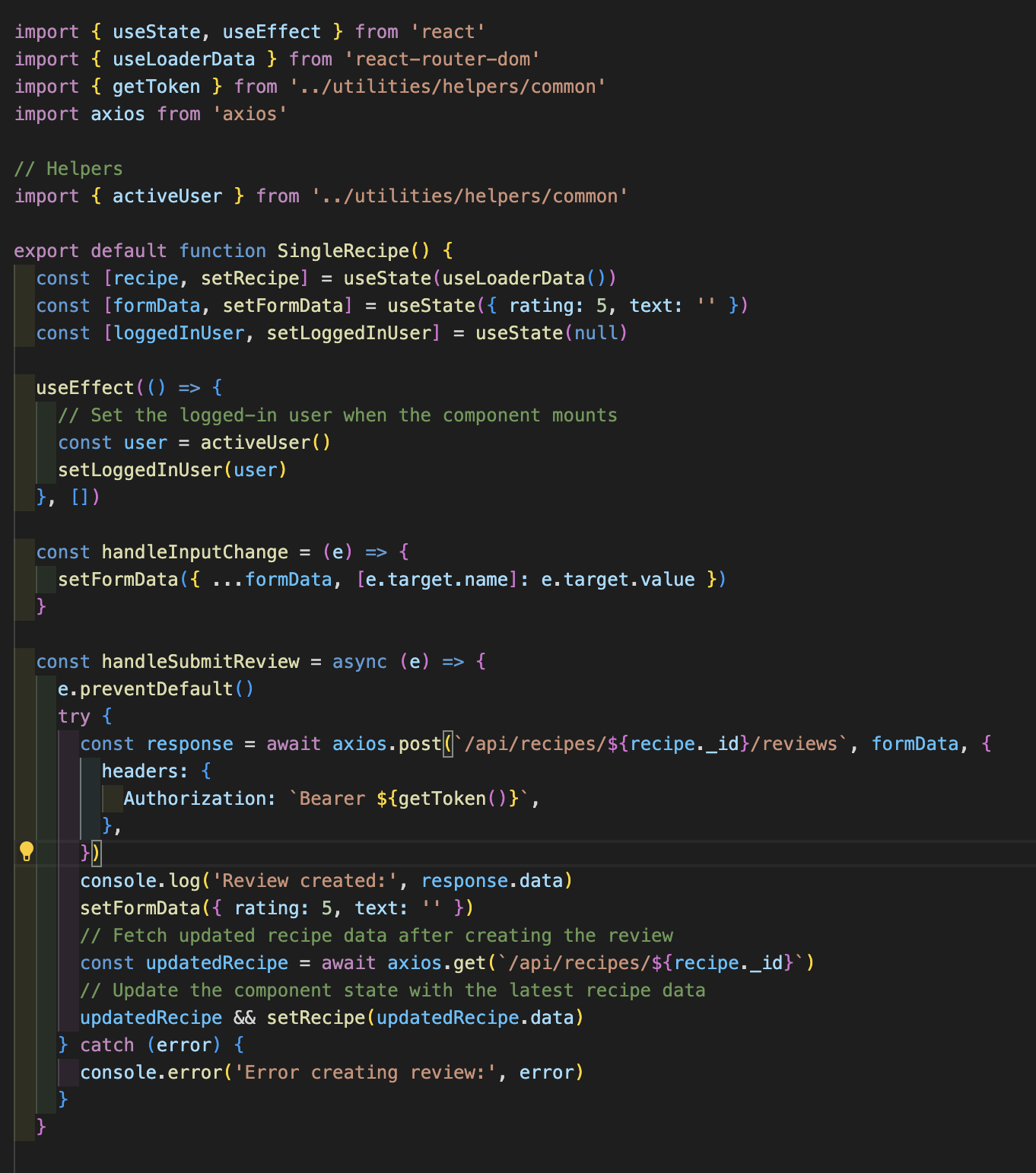
Day 2 – We initiated the coding for the backend server of our project. This included working on the seeds and establishing the routes for the website.

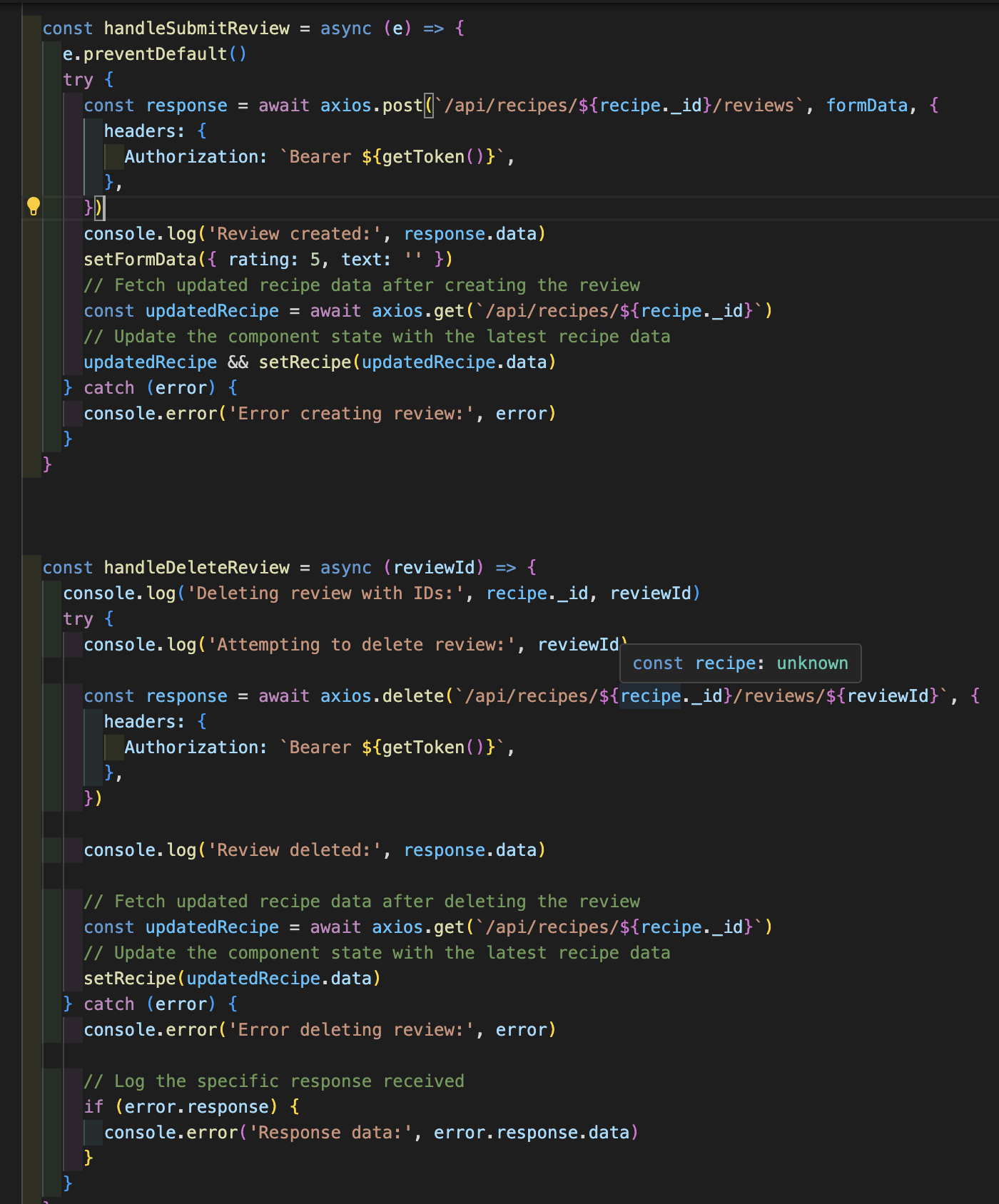
+ develop and add more technical details



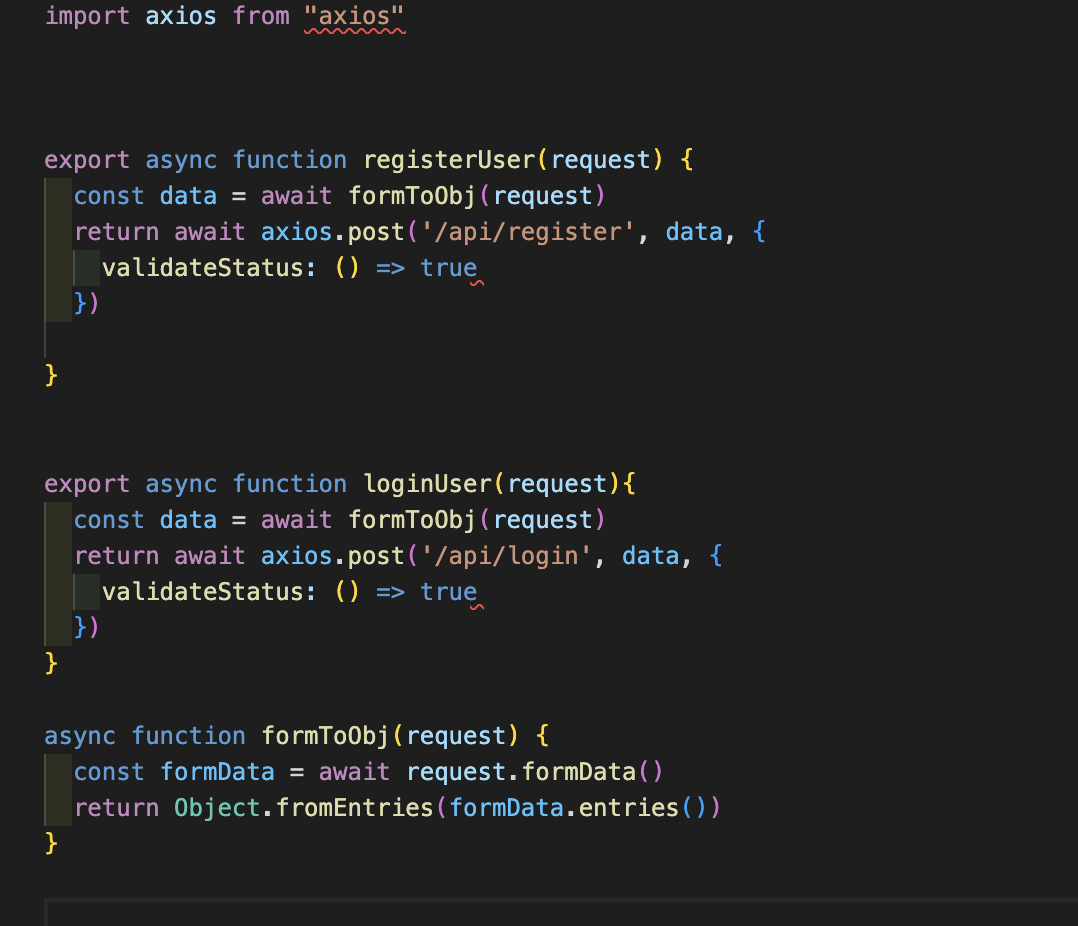


Day 3 – With the backend server well-structured, we transitioned to developing the frontend and began assigning tasks and responsibilities among the team. On the first day, my responsibility was to create the single Recipe page, where only the clicked recipe would be displayed.

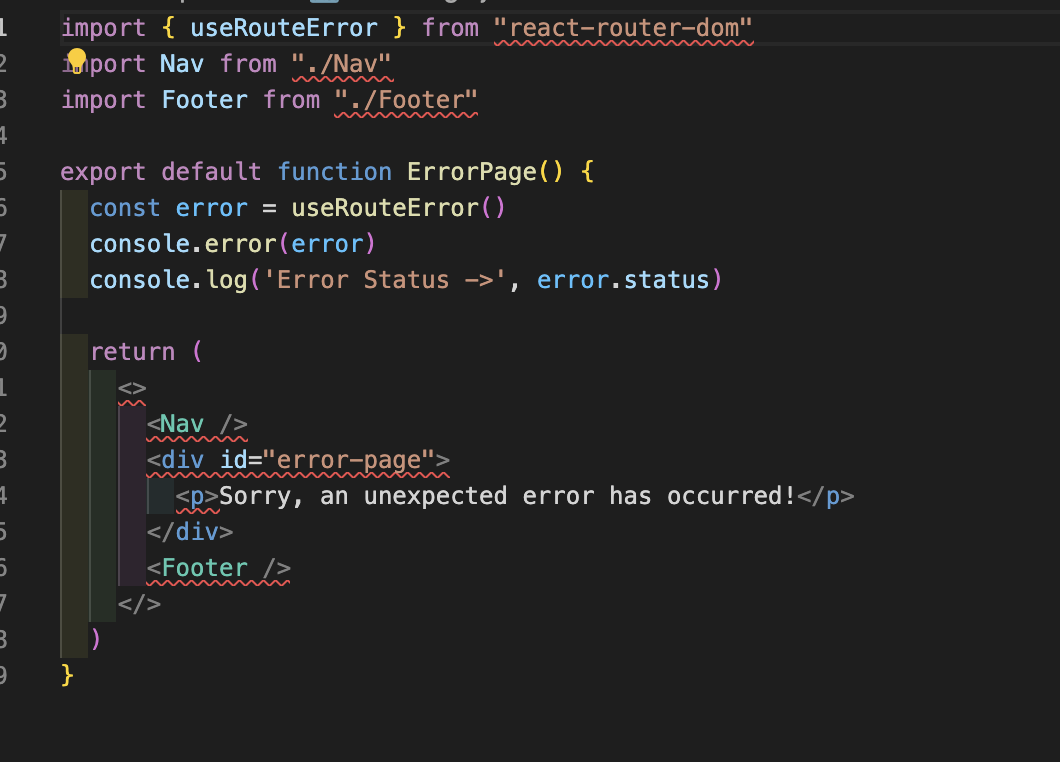


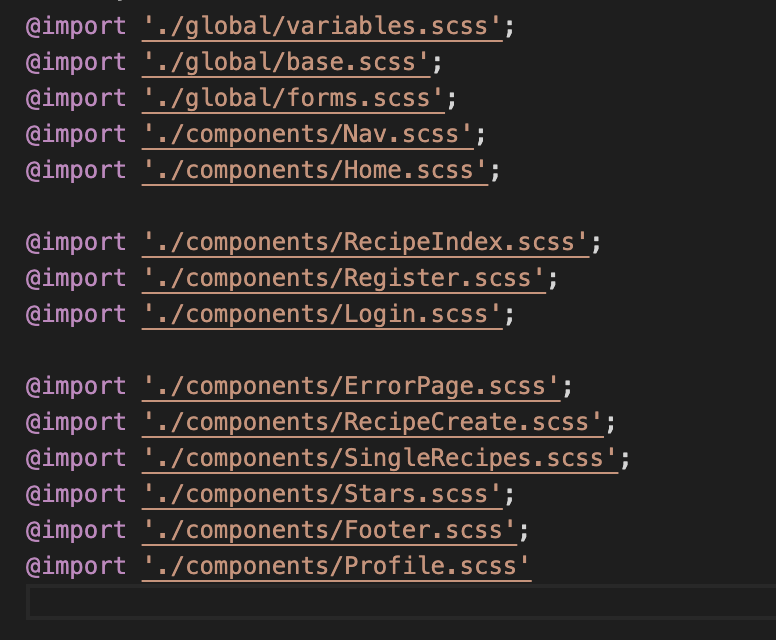
In the section below, we've outlined the structure for the reviews on the single recipe page.

Day 3 - We began working on authentication and added some styling to our page. + add more technical details



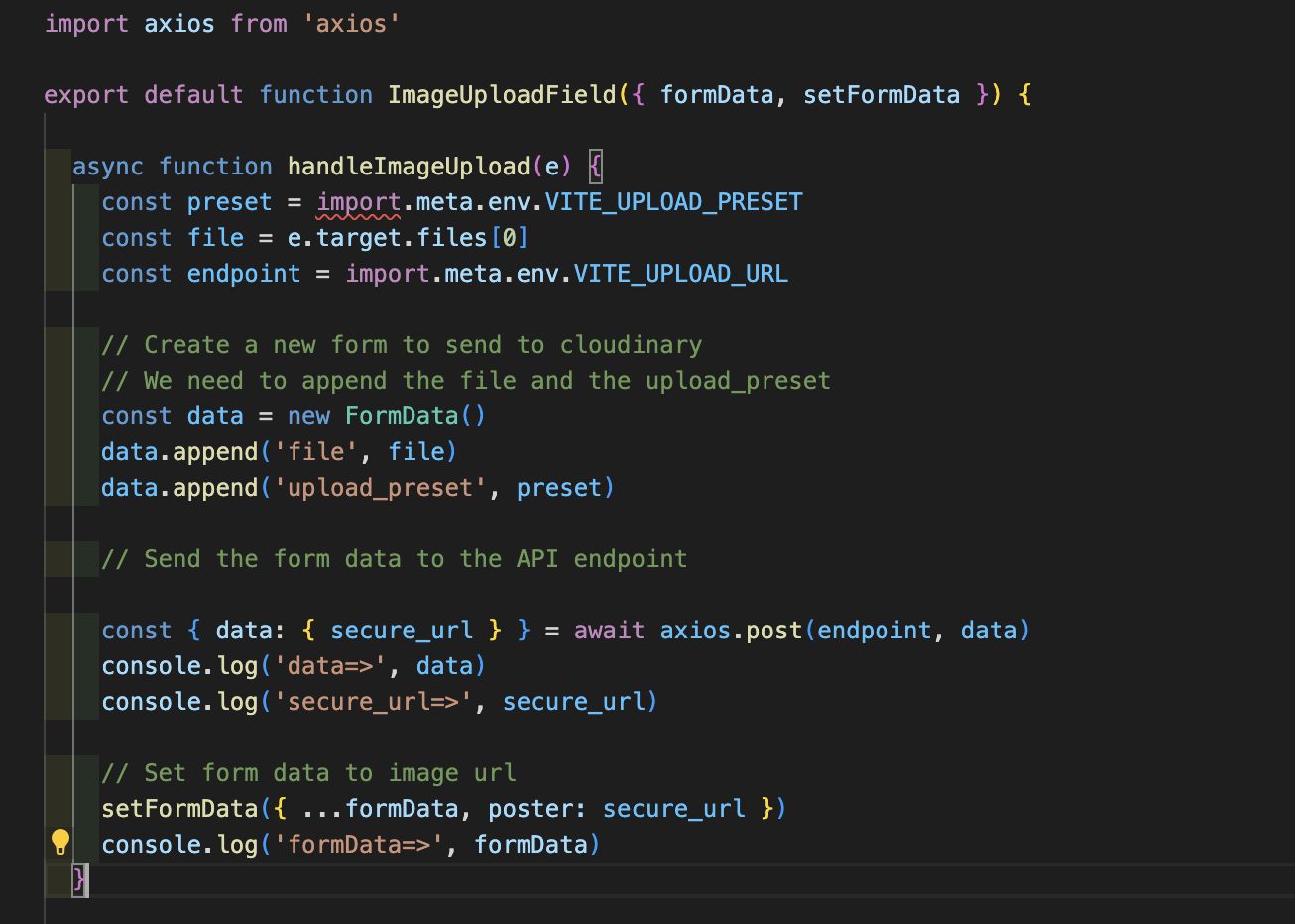
Day 4 - We designed the error page for our site and focused on enhancing the styling, completing pages that were already functional but lacked the desired aesthetic. **+ add more details**



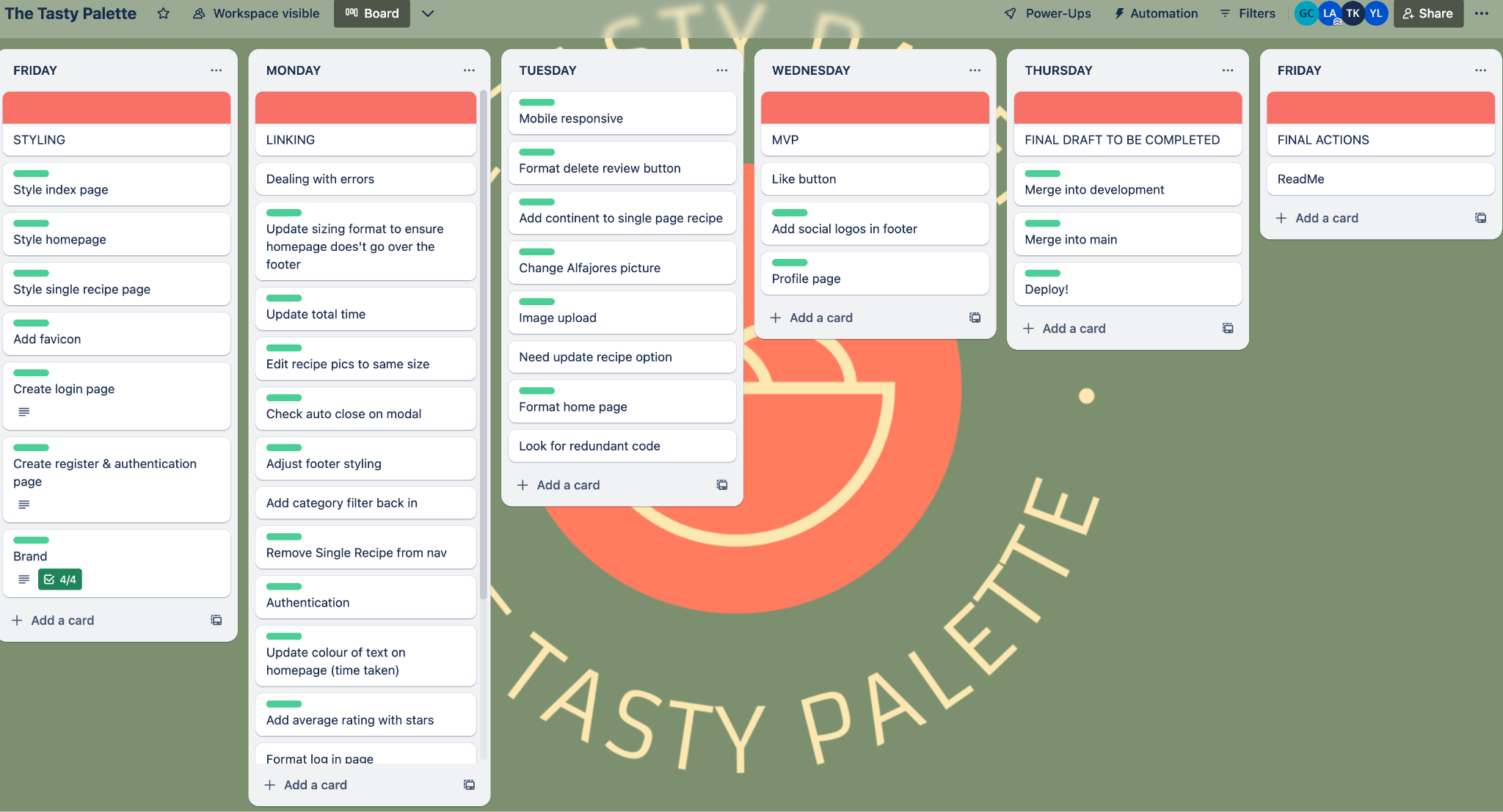


Day 5 - We achieved our Minimum Viable Product (MVP). At this point, our focus was on refining the user experience by eliminating errors in our pages and completing any remaining items. **+ add more technical details about your process.**

Day 6 - We encountered an issue with the upload link for adding pictures to the project. As a group, we resolved the problem, and soon after, all pages were fully functional again. >> **again you need to explain a bit more your technical process.**



Day 7 - We completed our project and successfully deployed it.



**Challenges**

**Instructions**

*Challenges are great for showing your learning journey and problem solving, and this is a section that many engineers will check out. Every day of your engineering career you’ll encounter challenges, this is part of your growth and development. It’s the challenges you encounter that helps you become a stronger and more competent engineer.*

*Here you will detail any particular challenges you encountered as you were coding the project.*

*Questions to answer here:*

* *What technical challenges did you come across?*
* *Why were these challenges?*
* *What problem solving did you do to rectify them?*
* *Team dynamics/ Project management*
* *Tools/Tech you used*

**Insert your Challenges here:**

I believe the most significant challenge for us was working in a group, as none of us had prior experience in such a setting. Once we organised ourselves and realised that tackling difficult problems as a group made things easier, the project flowed more smoothly.

My personal challenge was implementing the review feature—being able to display it on the page and enabling only the user who wrote it to delete it. Initially, I thought it would be simpler, but after encountering a few issues with the code, everything started functioning as intended.

**Wins**

**Instructions**

*The Wins section is your opportunity to highlight the aspects of your project you are most proud of. See this as your chance to showcase these parts of your projects to the engineers reading your ReadMes.*

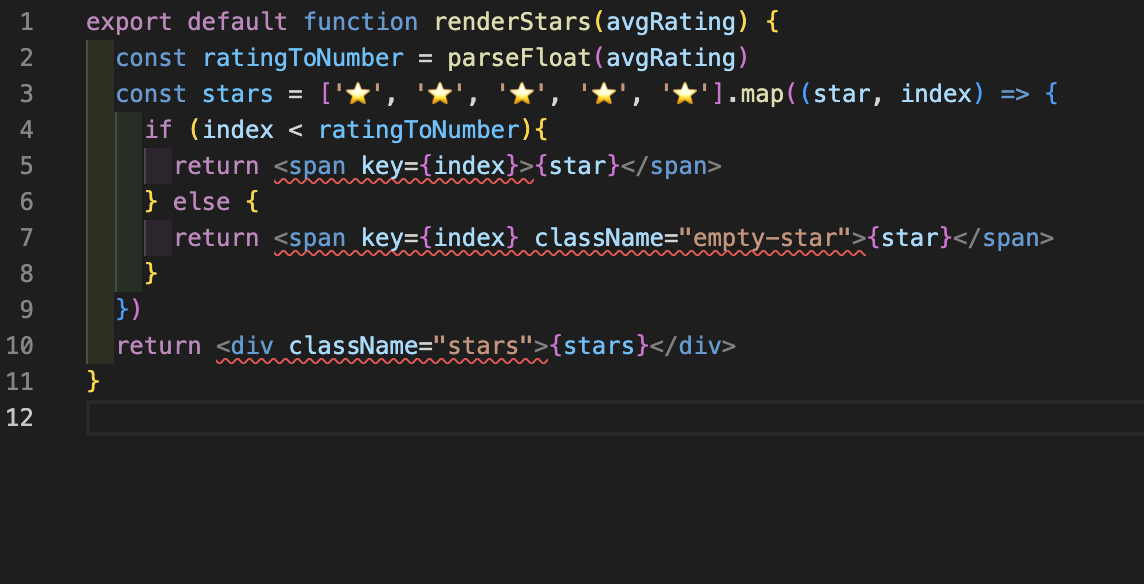
*Things you could discuss here:*

* *Interesting problem solving you did*
* *Strong sections of code*
* *Collaboration with other team members*
* *Visual design of the project*

**Insert your Wins here:**

I believe this project, involving both backend and frontend servers for the first time, was highly rewarding. I successfully incorporated all the ideas I had in mind, thanks to the collaboration within the group.

The most significant accomplishment for me was the review feature, as mentioned earlier. Implementing it required extracting data from another file to generate random reviews and calculate an average, which added a unique and challenging aspect to the project.



**Key Learnings/Takeaways**

**Instructions**

*This section is one of the other most important parts of your ReadMe from an engineers’ perspective and helps to differentiate each of you from your classmates and team members.*

*Engineers love to understand what you learn from each project and how it has shaped you as an engineer.*

*See this as your opportunity to show the engineers how your skills grew during each project sprint.*

*Things you could discuss here:*

* *What Technologies/Tools do you now feel more confident with? Tell them specifically what you learnt about these.*
* *What engineering processes did you become more comfortable with? Standups? Pair programming? Project management? Tell them what you learnt from these processes?*

**Insert your Key Learnings/Takeaways here:**

* MongoDB proved to be an excellent tool for our project, serving as the foundation for our database and review schema.
* Express.js played a crucial role in constructing a well-organised server-side application.
* Working with React was a delightful experience, allowing us to shape our client-side exactly as we envisioned.

**Bugs**

**Instructions**

*If you have any bugs in your project, it’s important that you flag them in your ReadMe. This helps the engineers reviewing your projects to understand that you are aware that there are issues - if you don’t flag these, then they won’t have that visibility that you know these problems are in your code and it can result in them not having a full understanding of your technical knowledge.*

*In either sentences or bullets, explain what the bugs are.*

*If you have no bugs, you can leave this section blank.*

**Insert your Bugs here:**

The most significant bug I encountered was related to the review functionality, which initially wasn't working. Apart from this issue, our website is currently free of any other bugs. >> Do you know how to fix it? If so, explain it.

**Future Improvements**

**Instructions**

*It’s common to get to the end of your project and have ideas on what you would do if you have more time, as well as how you might improve it.*

*If you do, you should detail this here. It’s great to give that context on potential future improvements, to share your creative or technical ideas with the engineers reading your ReadMes.*

*In either sentences or bullets, explain what your future improvements would be.*

**Insert your Future Improvements here:**

Enable users to modify their password, username, or email.

Implement a liking and disliking feature for dishes.

Provide users with the capability to add videos.

Incorporate a one-click sharing functionality for users.