**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 last project that I created for this course is a kitchen standard check to be implemented in any restaurant and the owner be able to record his fridge temperature and cooked meat as per law in Food Safety.

**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:**

[Kitchen Standard](https://kitchen-standard-checks-bbd60a709900.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
* Neon
* react-bootstrap
* react-router-dom
* sass
* 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:**

I had a 9-day timeline to successfully deliver the project. Beginning with thorough research, I meticulously outlined a wireframe using Trello. Leveraging the platform further, I strategically assigned tasks, prioritising the areas I aimed to address first. The conceptual groundwork was laid out in my mind prior to project commencement. Notably, this project was executed independently.

**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
* Backend: Django, Neon
* 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:**

Technical Requirements

\* \*\*Build a full-stack application\*\* by making your own backend and your own front-end

\* \*\*Use a Python Django API\*\* using Django REST Framework to serve your data from a Postgres database

\* \*\*Consume your API with a separate front-end\*\* built with React

\* \*\*Be a complete product\*\* which most likely means multiple relationships and CRUD functionality for at least a couple of models

\* \*\*Complex Functionality\*\* like integrating a 3rd party API or using a particularly complex React Component would mean that the CRUD and multiple relationships requirement can be relaxed, speak to your instructor if you think this could be you.

\* \*\*Implement thoughtful user stories/wireframes\*\* that are significant enough to help you know which features are core MVP and which you can cut

\* \*\*Have a visually impressive design\*\* to kick your portfolio up a notch and have something to wow future clients & employers. \*\*ALLOW\*\* time for this.

\* \*\*Be deployed online\*\* so it's publicly accessible.

\* A \*\*working app\*\* hosted on the internet

\* A \*\*link to your hosted working app\*\* in the URL section of your Github repo

\* A \*\*git repository hosted on Github\*\*, with a link to your hosted project, and frequent commits dating back to the \_very beginning\_ of the project

**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.*

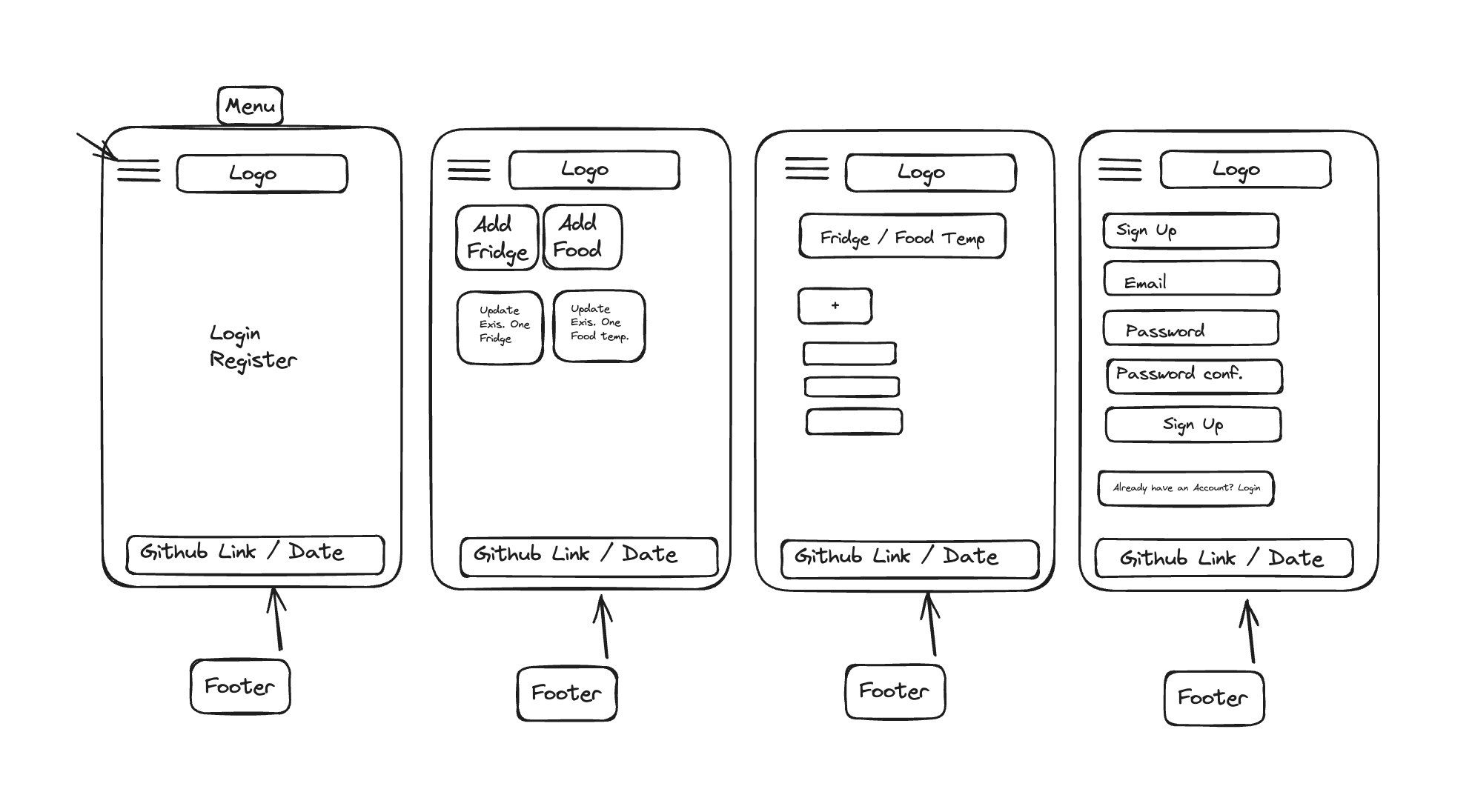
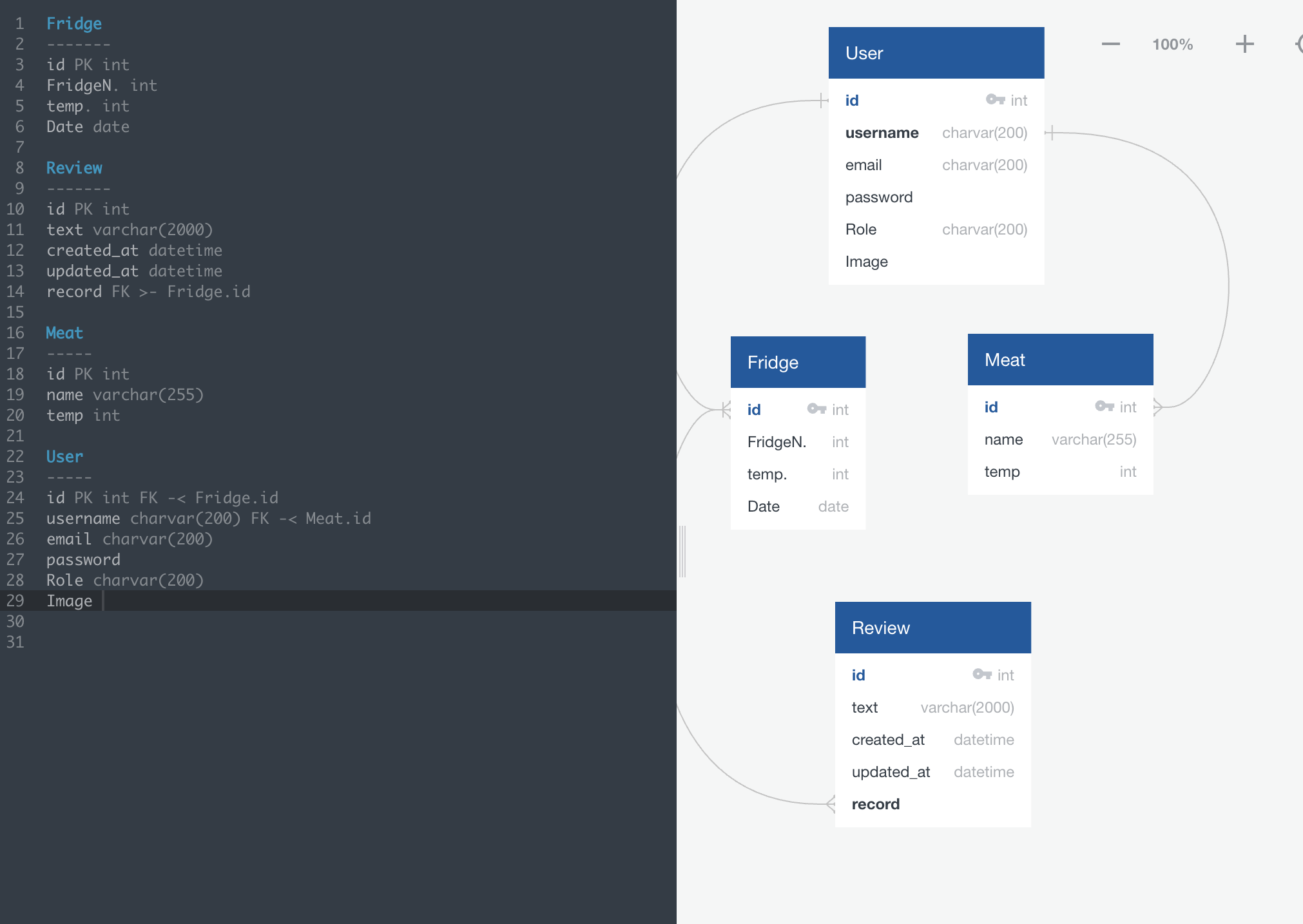
* ***Did you do any sketches****? If so, discuss this and include images.*
* ***Any wireframes of the front end and UI****? You did? Then explain this and include images.*
* ***Any ERDs****? Same here, explain and include images.*
* ***Use a project management tool to plan the sprint****? If so, talk 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 that I created before commencing my project. I had the freedom to design it according to my preferences. My 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.*

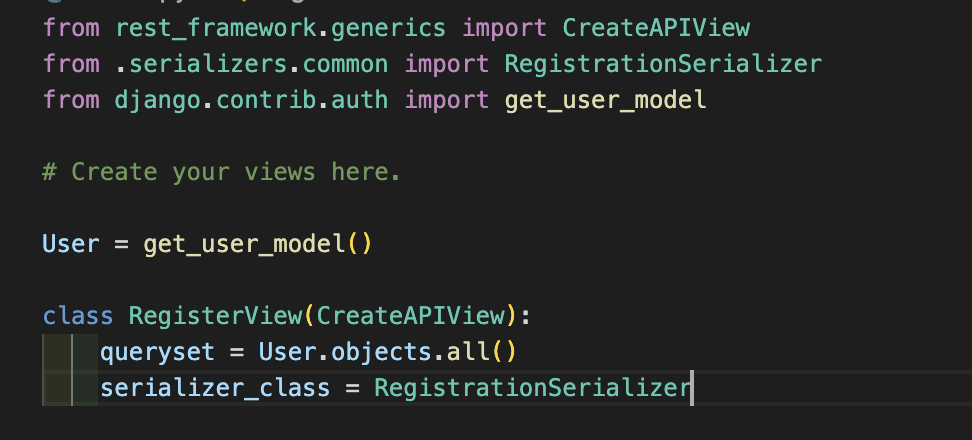
*Some people will document the build/code process by discussing the key stages they worked on. Others will do a day by day guide. It’s entirely up to you how you structure this, as long as you discuss all the key things above.*

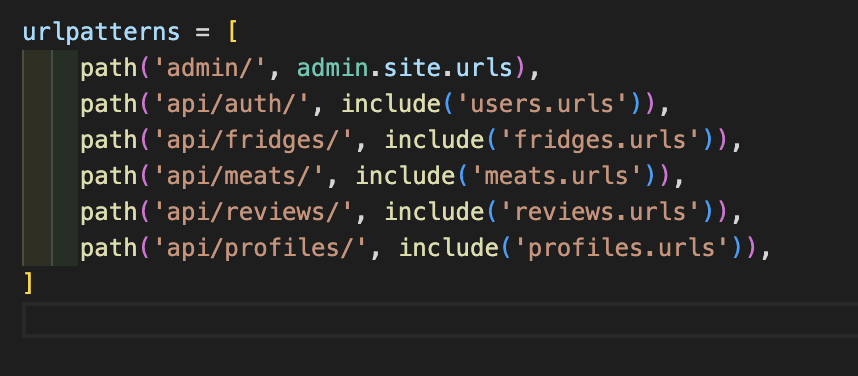
**Insert your Build/Code Process here:**

Day 1 – I started to search for the logo and finished my wireframe, as I already had the idea of what I wanted to do. I would like to make a site that it’s able to hold information from a coffee shop or kitchen.

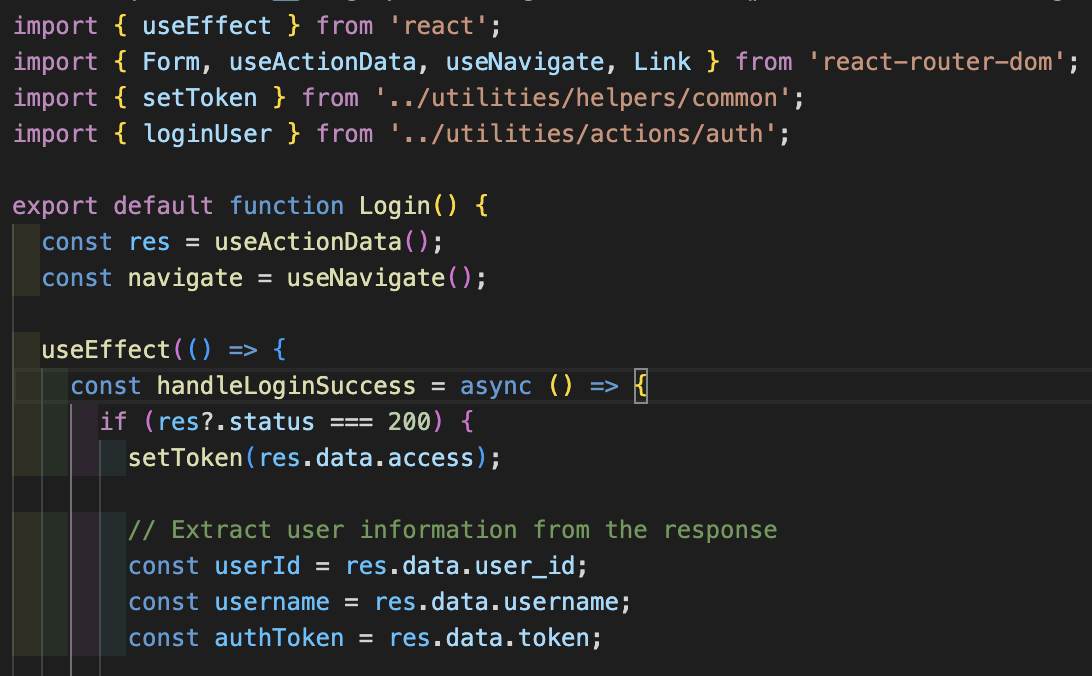
Day 2 – I initiated the coding from the backend server of my project. This included working on the Neon and establishing the routes for the website.

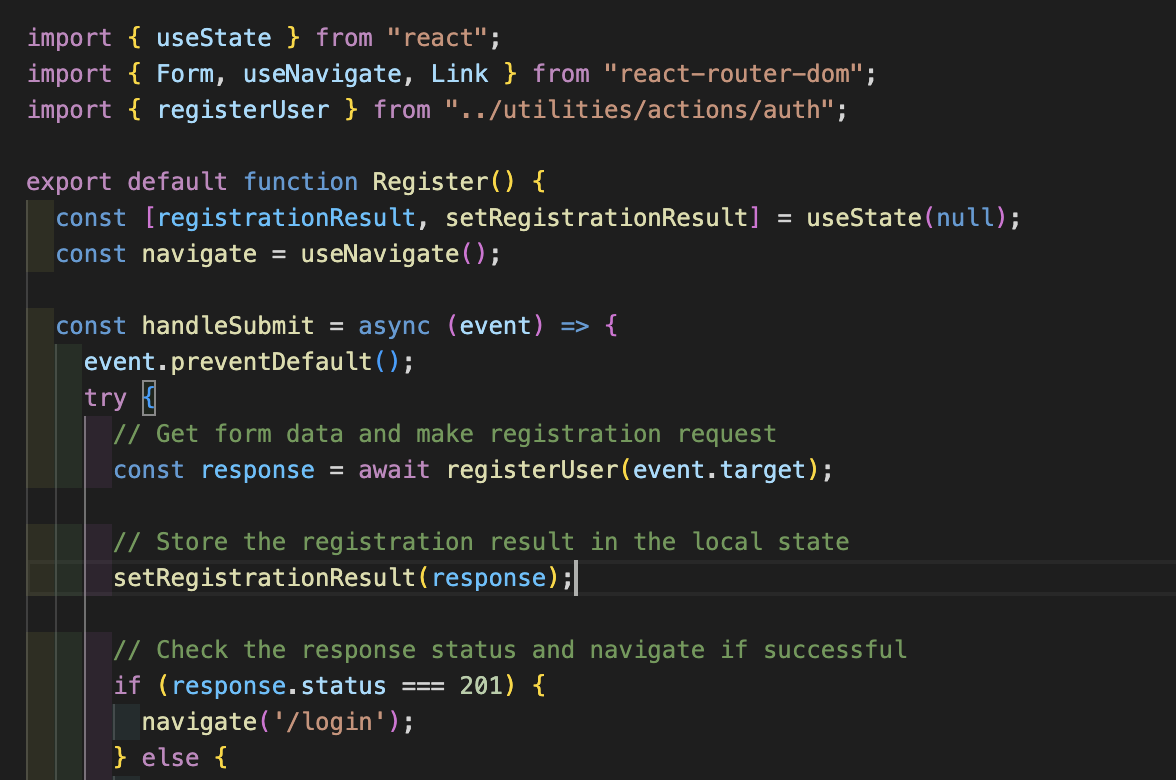
I don’t chosen to have loads of backserver informations, as my important focus was the frontend to be able to display the temperatures and reviews that the user would add to the page.





Day 3 –Once the backend server was meticulously structured to ensure seamless functionality, I seamlessly transitioned my focus towards the frontend development phase. Beginning on the inaugural day, I start to implement the login and registration functionalities, meticulously addressing all aspects of authentication necessary to fortify the robustness of my project.





Day 4 - As I embarked on the development of my project, I first contemplated the layout and content organisation for my profile page. After deliberating on which sections to include, I began executing the code to enable the display of fridge temperature on the screen. Concurrently, I initiated the construction of my profile page, aiming to showcase comprehensive information extracted from various sources within the project.

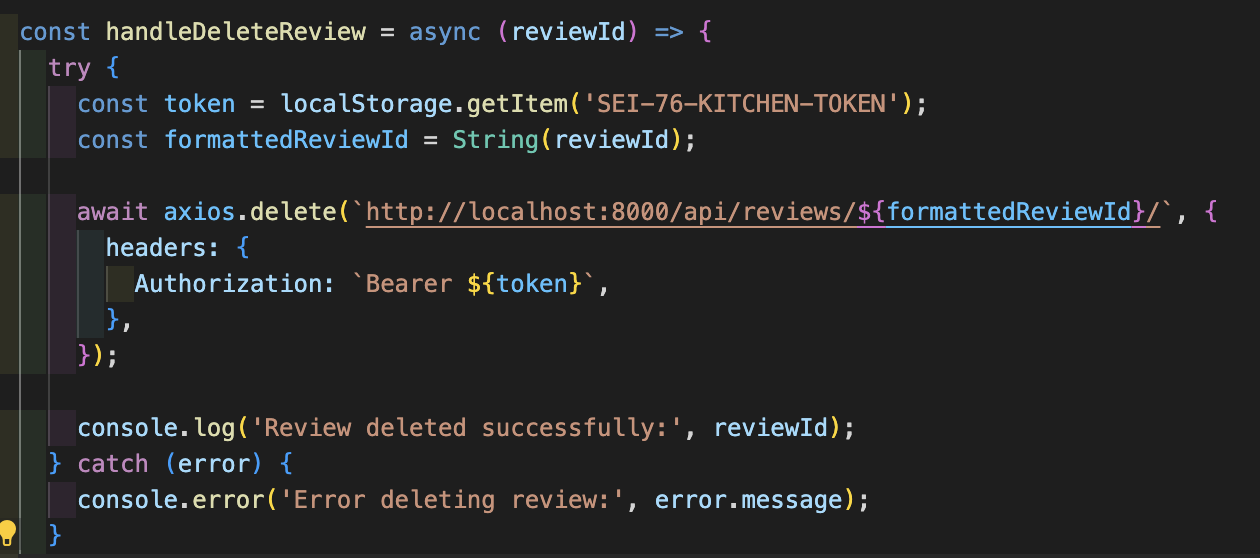
I created the Fridges forms that I had to use at my profile page, with all the information that was important for the user insert.



Day 5 - As I progressed towards achieving my Minimum Viable Product (MVP), I prioritised the completion of all CRUD (Create, Read, Update, Delete) functionalities within my project. To this end, I commenced the development of the handle delete review feature, thereby ensuring the inclusion of all essential components. This entailed orchestrating the seamless removal of reviews, marking a pivotal step in achieving project completeness.

Delving into the technical intricacies, I meticulously gathered the necessary information required for users to delete reviews securely. Employing token-based authentication mechanisms, I established a robust framework to verify user identities, thereby fortifying the authentication process.

Furthermore, I introduced a meaty form tailored specifically for utilisation within the profile page, aiming to streamline the user experience. Leveraging the useEffect hook, I optimised the timing settings to minimise waiting periods, ensuring swift and responsive interactions within the profile environment.



After this, I started to work at the update button that I wanted to display on the page, as I wanted the user to be able to update the meat records.

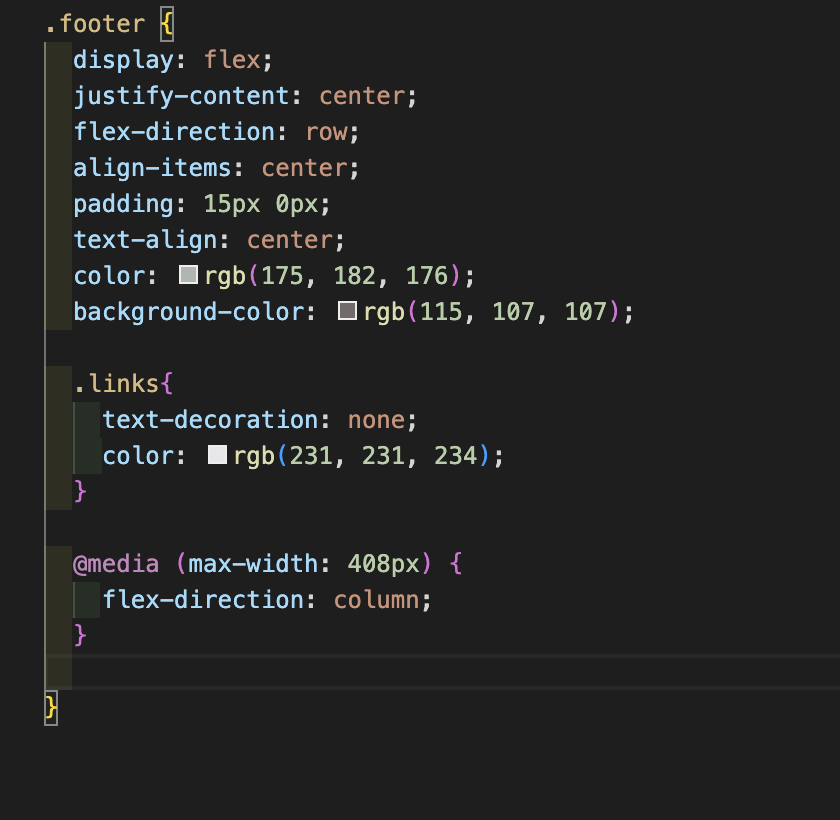


Day 5 - I achieved my Minimum Viable Product (MVP). At this point, my focus was on refining the user experience by eliminating errors in my pages and completing any remaining items.

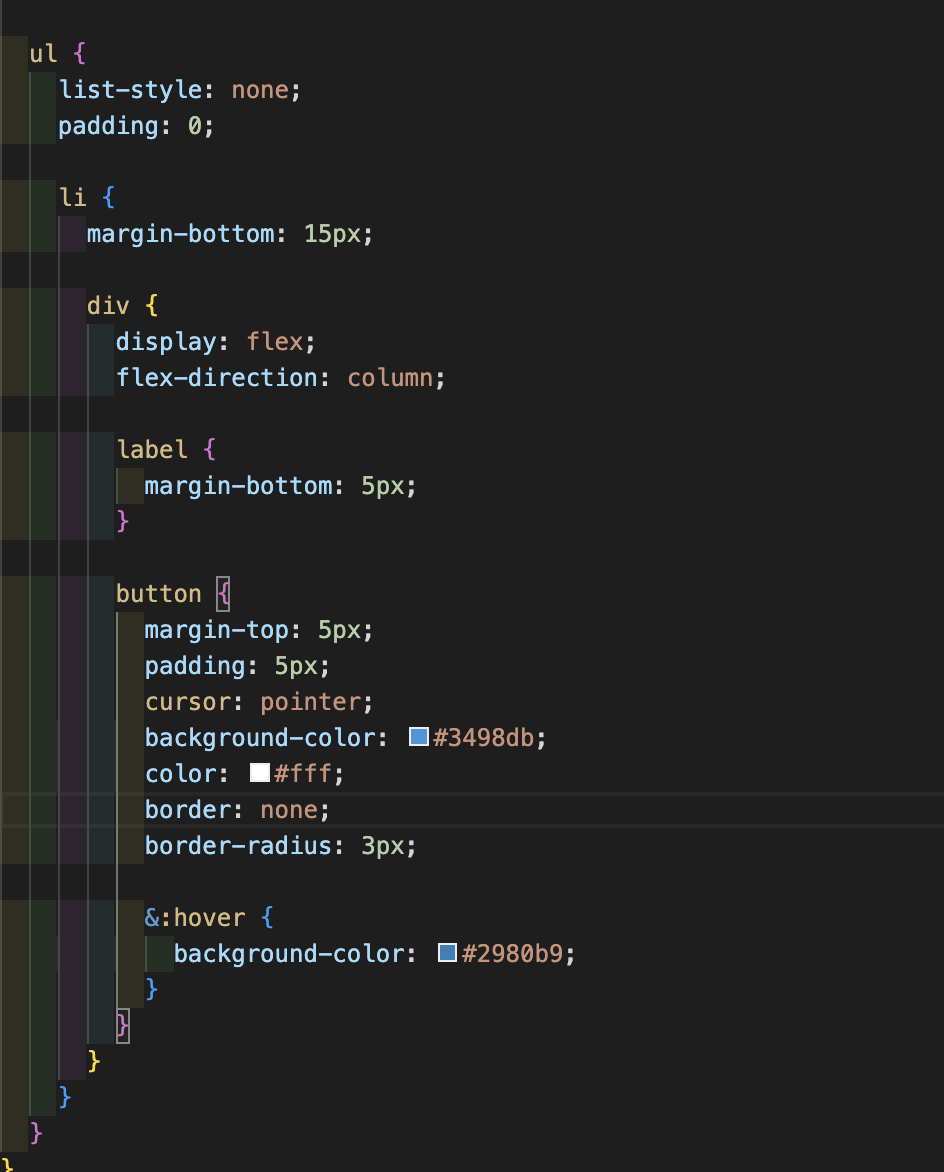
I started to insert some console.logs to look at any issues that I may face, it turned out that all was running as it should and I started to work into the style.

Day 6 - Commenced styling the project interface, beginning with the creation of the footer section.

Selected the colour palette to be incorporated throughout the project, prioritising hues that maintain a subtle visual impact for potential users. The chosen colours were carefully curated to ensure a harmonious and user-friendly experience.



Then I moved to my profile page. I want this page to display the records side by side, meat and fridge, and develop the technical process. It looked exactly how I planned in my mind.



Day 7 - I have my project done and ready for deployment.

**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 had a great experience with this project. The challenge that took me more time to resolve was the token authentication that I was using in my project. I tried to use the same approach as my previous projects, using the token to guarantee the access to the user, but, in this project, I had to use the access Token to reach this, otherwise, the authentication will be returned as failed.

After resolving this problem, all the issues stopped and I could start to display my data at the profile page. I had also an issue with the load data for my page, probably because I used a freesource database holder, then after a research, I saw that it has to be used useeffect to resolve this issue, I changed my code, and all came as should be.

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:**

Throughout my project journey, I encountered various challenges and achieved significant wins. One notable triumph was overcoming the hurdle of implementing token authentication. Initially, I relied on conventional methods, but this project demanded the utilisation of access tokens instead of the usual authorization mechanism. After diligently resolving this issue, all subsequent obstacles ceased, empowering me to seamlessly showcase my data on the profile page.

Additionally, I triumphed over the challenge of loading data efficiently for my page, which was impeded by utilising a freesource database holder. Through thorough research, I identified the need to leverage the useEffect function to address this issue. Implementing this solution involved meticulous code modifications, ultimately leading to the seamless functionality of the page.

On a more personal level, I tackled the challenge of implementing the review feature, which involved displaying reviews on the page and enabling only the respective user to delete them. Despite initially underestimating the complexity of this task, I encountered several hurdles during the coding process. However, through persistent troubleshooting and refinement of the code, I successfully achieved the desired functionality, marking a significant personal triumph in 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:**

* Neon proved to be a free database storage tool for my project.
* Django played a crucial role in constructing a well-organised server-side application.
* Working with React was a delightful experience, allowing me to shape my client-side exactly as I 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, my website is currently free of any other bugs.

Also had a bug for displaying the name of the user, it's related to the beck server that I filled up, but, as I ran out of time, I couldn’t finish this implementation.

**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 text saying that the password or email don’t match.

Provide users with the capability to add pictures.

Display in separate pages, Meat Temperature and Fridge Temperature.