Sprint Minutes:

Sprint 1 of 5

The idea is:

- App where we can input shopping. Updated and stored in inventory.
- Then we can generate recipe based on available ingredie3nts.
- And generate shopping list based on that. Or we can add stuff that aren't low or aren't in the inventory – adjustable shopping list.
- Automatically update inventory once we select the recipe we want to use.
- 3 separate stock for cupboard, fridge, freezer. Able to click on those and you can see what's inside.
- Linked to SQL database. -> by default we will fill with random quantities and ingredients = starting stock.
- Able to update or delete ingredients.
- Anna K to try out a demo on Figma to brainstorm and ensure we are only on the same page for how we want it to look and function.
- Extra: Log in page to elevate code. -> another database to store user info in SQL.
- Basic Functions:
  - JS, HTML, CSS – button clicking, webpages,
  - Python
  - SQL: DOT TO CREATE
  - Fridge stock table
  - Freezer stock table
  - Cupboard stock table
  - Ingredient
  - Quantity
  - Type/Description
  - Min needed for stock
  - Sell by date
  - Storage location
- Extra:
  - User info table
  - Edamame API – filter recipes through that.

    - Code will need to filter through available ingredients and then go through API for recipe.
    - User can specify main protein or thing: ‘what do you want tonight’, ‘chicken’, ‘ok here’s a recipe for chicken wings’
    - Order proteins by sell by date:
        - Chicken
        - Beef
        - Tofu
        - Fish
        - Pork
        - Egg
    - Extra: Surprise me button!
    - Extra: option for high protein meals or low calorie etc
    - Function to control range for example: where if less than 1 protein in fridge suggest another protein to buy
    - When recipe is suggested have button for ‘Yes, I cooked this’ so that stock can update.
    - There should be a function where we have alerts when things go below certain stock level e.g. milk
    - Python functions to make calls to API:
    - Function to connect to API: Amy
    - Python function to add, delete, update etc the SQL:
    - Python function to add: Vanessa
    - Python function to delete: Karen
    - Python function to update: Lauren
    - Lauren has done HTML and CSS
    - We can create a logo and brand identity.
    - Vanessa created repo on GitHub
    - First function: let’s generate a random recipe first.
    - Python function to add to inventory
    - START with SQL stock database
- Reconnect: Sunday 20th at 4pm or we may adjust for USA time
- Lauren has made a merged activity log so we can monitor our progress – it's pinned on slack.

Sprint 2 of 5:

- Put files in .gitignore.
- API connection is working but Amy still wants to completely finish it.
- Will need to put in exceptions / insertions
- Clarify – should we be using lots of classes? Should everything we write now be a class?
- Go over class recording showing previous cohort project to check if they used lots of classes?
- For HW2 we will tackle it next week but we may assign Qs to each other or just each take a stab at the questions and put together a whole complete file.
- Make sure we are filling in the activity logs!
- Add, update, delete item functions are complete.
- Next function we want to do is iterate through what we have through stock and find and generate random recipe.
- Base it on proteins – ask user to select protein from list of what's in stock. Query will find proteins, display, and then search the API. Start with a list of 10. Then select an item. Then we see what we need to buy to complete the recipe and what we already have.
- We may need a unit converter – everything in the API can be in grams.... We may just change our database to show only grams – check where the word decimal is coming from when show all.
- Code: look at proteins and select = Lauren S
- Search API and show list of 10 = Lauren A
- Search stock based on the recipe ingredients = Dorothy
- Function shows true and false on whether we have items= Dorothy
- and if false then add to a shopping list. = Vanessa
- Code to present shopping list = Anna
- When recipe is used – update stock levels = Karen

Sprint 3 of 5:

- Some over spill from spring 2 – Karen has Wed planned for this
- Need to check the DB (GRAMS!) - Dot!!
- We should all try giving the code a look over – Lauren A says she will check it after our meeting
- How do we get the code to run – why is PyCharm mean to us
- Need to include Test file – unit tests – Lauren S says she will try and get some basics in. Like a separate directory of files? Lauren S will share some resources on this in slack also.
- Need to tidy it up
- This sprint will be focused on getting the project to run and understanding what each of us have done.
- We should do a group retrospective – also what happened to our SWOT analysis? - another Trello for this so we can all put our SWOTs in there – Dot focus on making this diagram
- Run file (ap.py) - you'll see website, once you click on generate recipe, you'll see new page ...etc
- Debugging focus this sprint. Remember to comment out codes to check what functions are having issues.
- Lauren S idea: Put code that isn't used in separate file to showcase our thinking process and how we changed over time!
- Error handling – Vanessa
- Presentation slides start – Dot
- Screen record pages - Anna
- Update meeting – Wednesday after class

Sprint 4 of 5:

Notes unavailable at this time.

Sprint 5 of 5:

- Trying to fix and debug - all hands on deck. Try to front load work this week as many people have busy weekends coming up!
- Fetching protein data is a bit fidgety – if it isn't working for you, check your SQL table as Vanessa updated it to include a protein label.
- Make sure shopping list function works and can produce a txt file – Anna to make this shown in the HTML
- Transaction to roll back or commit shopping list function – Dorothy to do
- Data units adjust - Dorothy
- More data to be added to database – Vanessa
- What can we do with the shopping list – Karen