

Kitchen Compass

Group B2-G5:

Joey Hobrecker, Laura Keller, Andrea Spagnolo & Kyra van Schaik

INDEX

Abstract.....	3
Introduction.....	4
Our Design Process.....	5
Final Concept.....	21
Discussion.....	27
Sources.....	28
Appendix.....	30

ABSTRACT

Students often find it difficult to cook due to lack of time, knowledge and ingredients leading to food waste and repetitive and bad meals. This is why we created KitchenCompass, an AI driven recipe generator in the form of a chef's hat. It creates recipes using the ingredients the user has at home.

The goal of KitchenCompass is to make cooking easier and more fun, to reduce food waste and to teach students different recipes and cooking techniques.

In this report we will discuss the different parts of our design process. For example, we will go over our brainstorming sessions, our user research, our different prototypes, our user testing and the different iterations of KitchenCompass.

The goal of this report is to show the development of our concept and the steps we took to create the final version.

During the Demo Day we presented a completely functioning prototype used to gather insights and feedback for further improvements and explorations.

INTRODUCTION

Cooking is one of the daily tasks many students can struggle with. When you are still living at home someone else always does the cooking tasks for you. However, what do you do if you live alone and have never cooked before?

Recipe Books are not made for younger generations, it's an outdated medium. Our generation is more technology-based, finding recipes and cooking lessons from online platforms such as Youtube, TikTok or Instagram. Nevertheless these are often "StuntFoods" (just good looking, not good tasting) or they are general recipes, not personalized to your tastes and needs.

Using AI for recipe generation sounds perfect then, but existing tools like ChatGPT are impractical, lack an user-centric design and are too general. That's why we created KitchenCompass, a wearable chefs hat that generates recipes with the ingredients you have at home! It makes cooking easier, better to understand, more fun and sustainable!

Each member of our group joined Artifice to explore how creating a product centered around AI would work and to create something that was usable in everyday life and used AI as a basis. We were perfectly joined together into a group!

So to reiterate, our motivation for creating this product was to create a simple and usable AI product that could solve an everyday problem we all experience. We wanted to create an AI tool that was more than just a search engine, it needed to be an experience!

We noticed how we struggled with cooking, and after some research we saw that we were not the only students with this problem.

Our design process was iterative and aligned with the reflective transformative design process (RTDP).

Key phases in this process were:

- Ideation and topic exploration
- User research
- Prototyping and Testing
- Creating a working and final prototype

OUR DCSIGN PROCESS

Ideation

In the first couple of weeks of the Artifice project we had a continuous decision-making approach. We used mind-mapping to explore different ideas and pitched concepts we came up with to each other. This process enabled us to get to know one another and get closer as a group. We learned about one another's interests as well as each other's soft and hard skills, expertises and interests. This led to us taking on responsibilities based on which areas of expertise suited each member the best. However, we made sure to distribute tasks in a way that we could learn from one another too. The first phase of the our design process was focussed on getting to know each other and exploring concepts.

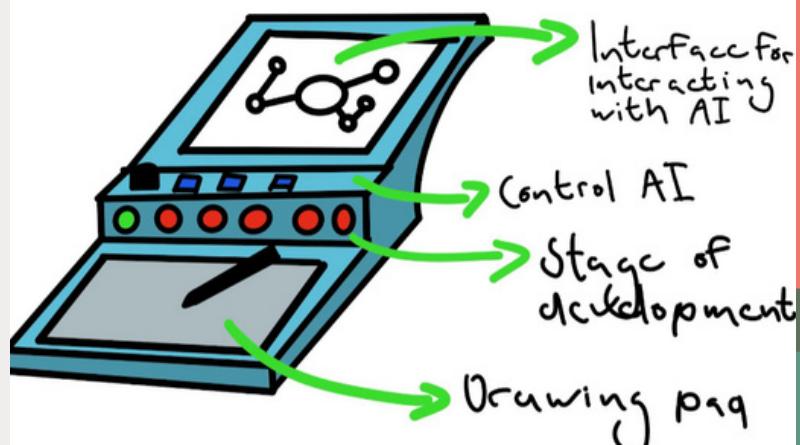


Figure 1 - concept for an AI product of one of our team members

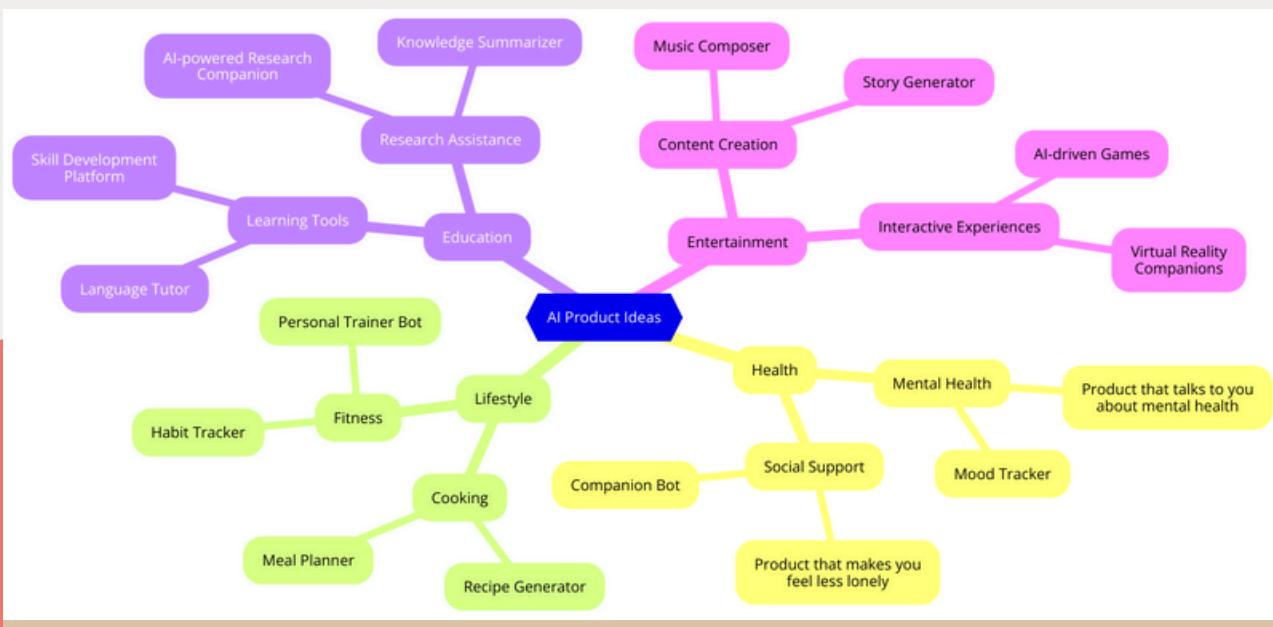


Figure 2 - The Mindmap of our first brainstorming session

Moreover, the PDP and goal setting activity with the AIMION tool workshop was insightful; we were able to get a clearer idea of what our next potential steps would be, as well as the chance to compare our values and visions and set goals as a group. It was really beneficial to us to see how our values and priorities were closely aligned, and therefore goal setting was rather easy.



Figure 3 - Photo of PDP and goal setting AIMION tool workshop results

During our first brainstorming session we found something that interested our whole group; cooking with an AI assistant. We all cooked on a daily basis and saw a lot of room for improvement in our processes.

That's why we decided that the following would be our goal for this project:
"Create a product that guides and helps students through the cooking process."

Before we would really make this the focus of our project, we needed to know for sure that we were not the only students with an "imperfect" cooking process. Besides that, we also need as much information as possible about students their cooking habits and preferences. This brings us to our User Research.

OUR DESIGN PROCESS

User Research

Questionnaire

Like we said before, we were our own target audience and based our first idea on our own experiences. We were aware that this could lead to a product made only for us, and not for students in general. So, we started our research process with an online anonymous questionnaire for our user group. It consisted of both open-ended and closed-ended questions, where we asked them about their cooking habits, their struggles while cooking, where they find inspiration for new recipes and if they would be interested in an app suggesting recipes based on ingredients they have at home.

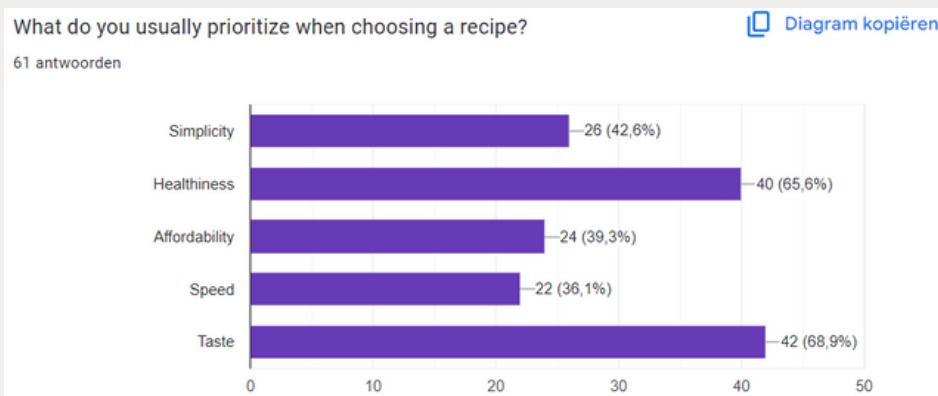


Figure 4 - What do you usually prioritize when choosing a recipe?

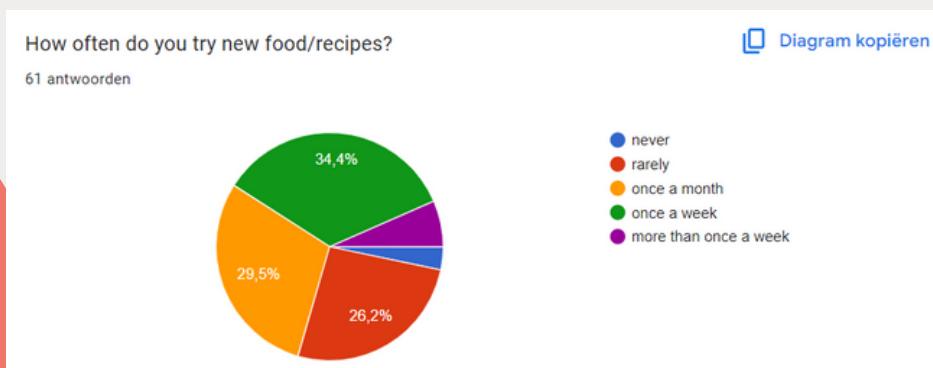


Figure 5 - How often do you try new food/recipes?

61 people filled in our form, gives us a varied and good pool of answers. **56 were students who lived by themselves**, making the results fitting to our research. One of our main take aways was that more than half of our participants try out new recipes rarely to once a month, see figure 5. This showed us that a recipe generator focused to get people out of their comfort zone could be interesting to explore.

We asked our participants what kind of recipe they prefer (written, audio, visuals, multiple answers were possible). **34 participants liked visual recipes, 44 liked written recipes.** This informed our use of written recipes and our exploration of using AI generated images. Another interesting insight we gained was that **54 of our participants were interested in an app suggesting recipes based on ingredients you have at home.** This was a feature we did not focus on in our earlier prototypes, but later in the design process we shifted our concept and this became the main feature of our final product.



Figure 6 - an image of spaghetti bolognese generated by ChatGPT [1]

Would you find it useful if an app suggested recipes based on the items you already have at home?

61 antwoorden

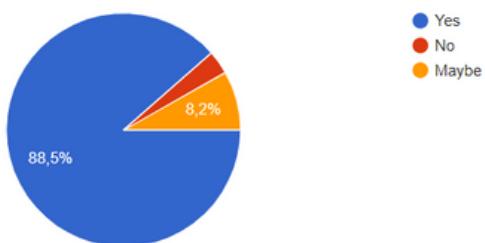


Figure 7 - Would you find it useful if an app suggested recipes based on the items you already have at home?

We only discussed some of the results from the form, but every question played a critical role in shaping our final product. They served as the foundation for our entire process and we made sure that every decision we made aligned with the results.

AI cookbook- Bytes

To familiarize ourselves to cooking with AI generated recipes each group member created and cooked one AI recipe each week and reflected on it during and after the cooking process. This helped us understand how an AI model creates recipes, what is possible and what is not and how to prompt for the best results. We explored how far we could push our AI generated recipes and if the meals were even any good.



Figure 7, the inside of our cookbook

Initially we tested with easily available models like ChatGPT and Microsoft Copilot. Further down in the project, we started using our own prototypes for our weekly testing. This gave us a clear idea of how to improve each iteration. It was easy to see what worked and what not when using the product yourself. It enabled us to identify issues with the interactions and the generated recipes. In the end of the project, we collected all of the recipes and reflections into an AI research Cookbook; Bytes.

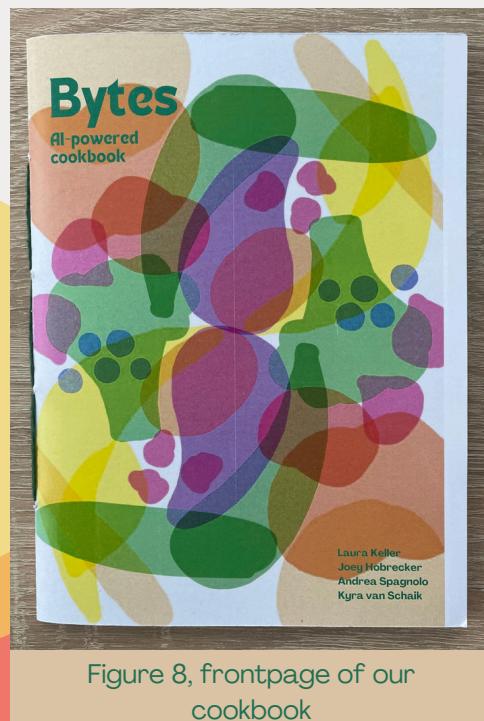


Figure 8, frontpage of our cookbook



Figure 9, back of our cookbook



Figure 10, one of the pages from our cookbook

OUR DESIGN PROCESS

Prototyping & Iterations

Our first iteration consisted out of a bowl with an integrated scale, timer and small printer connected to an app on your phone. A student could enter the requirements for their recipe on their phone and the bowl would guide them through the whole cooking process. It would help measure the amount of food, set timers and print the recipe. We wanted to limit the focus people would have on empirical values since this can lead to several eating disorders, so we decided to use a simple color scale. We created this concept in several brainstorming session and visualized it with the following sketches.

Sketches

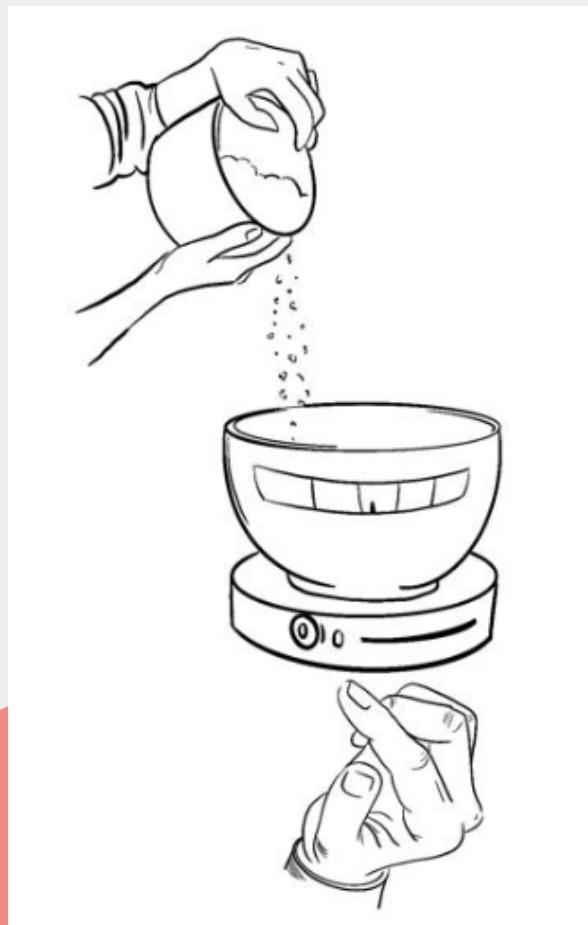


Figure 11, sketch showing possible iteration

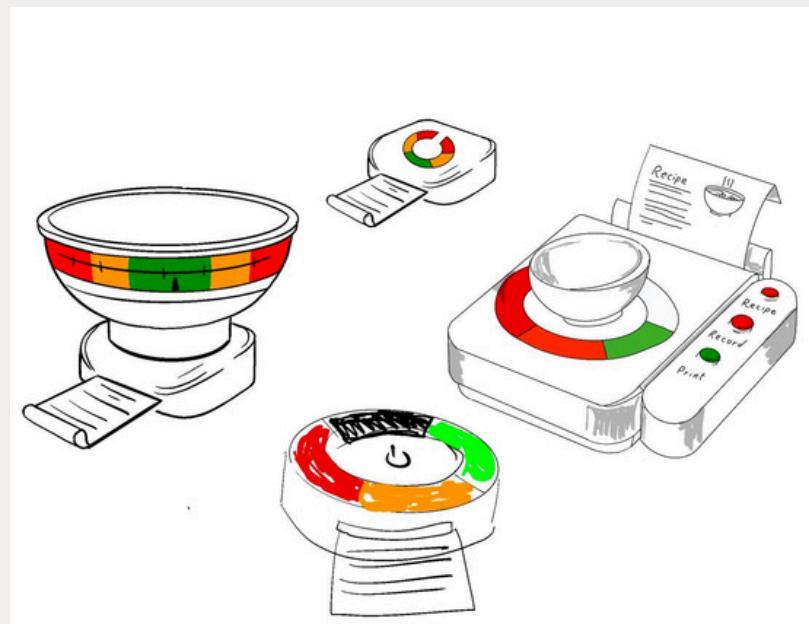


figure 12, sketches of our initial concept

Lego prototyping

We made our concept more tangible in a workshop where we were encouraged to use Lego's for low fidelity prototyping. We ended up with several prototypes made from LEGO. The outcomes were creative, exploratory and inspiring. We were able to visually observe what each one of us had in mind, compare and get inspired by each other's ideas.

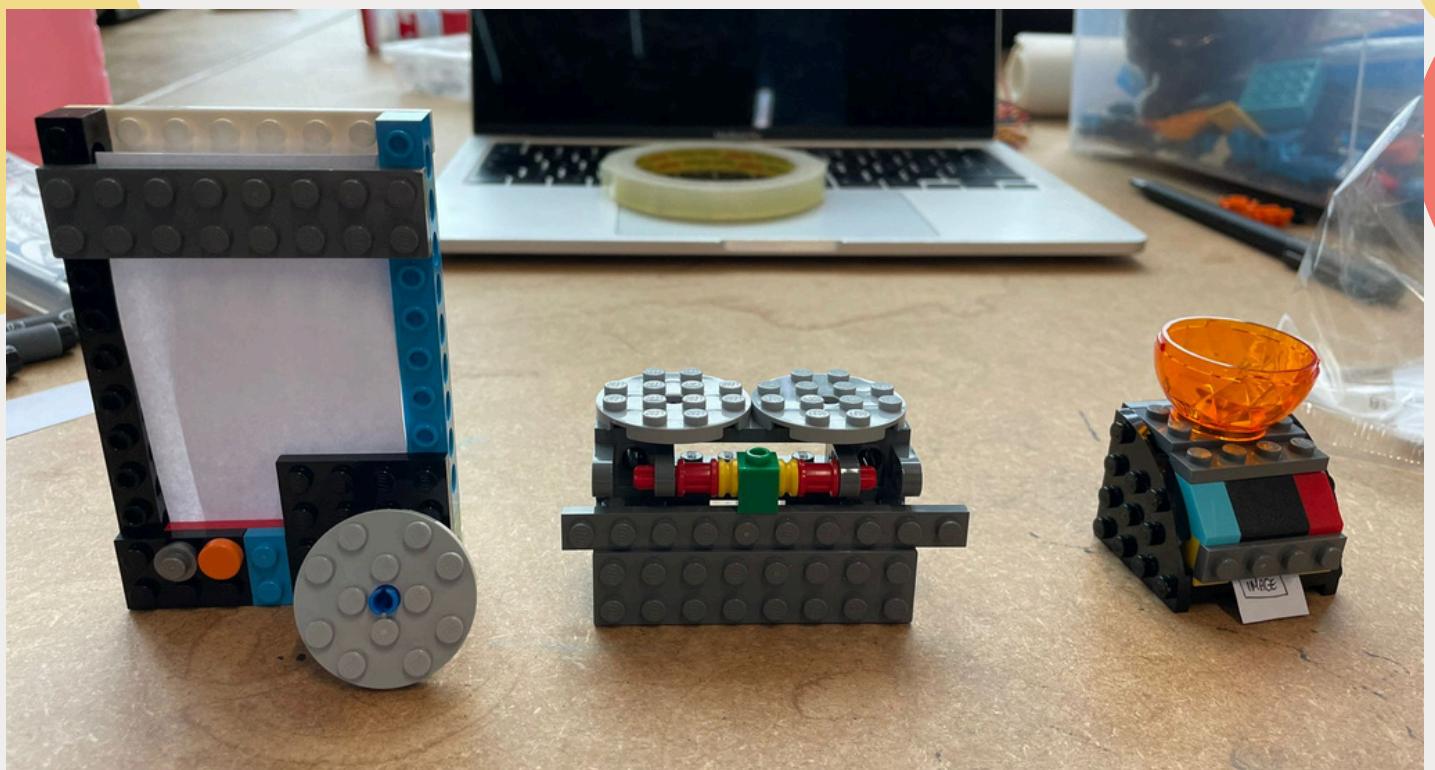


Figure 13 , our Lego prototypes



Figure 14, The idea of having a receipt printer stayed throughout our whole process

Flavor Fusion

Having a good idea of our physical prototype, we started working on the digital part of our product; the phone application and the AI-models. We called these digital prototypes ‘Flavor Fusion’. Every week we tested a new iteration of this application and added new features and improvements based on our own reflections. By trying out the model and recipes, we were able to quickly and reliably improve it. The final version of ‘Flavor Fusion’ was a website where you could generate a recipe based on cultures or general themes.

First, we will explain the ‘cultures’ mode. You started with a map where you could choose one or more countries, the recipe would be based on local dishes from these countries. Then, you could use multiple sliders to specify details like quantity, difficulty, amount of protein/carbohydrates, etc. After this final step, the recipe would be generated. The general mode started asked you for a prompt. This could be everything from ‘I feel like pasta tonight’ to ‘give me something jazzy’. After this, the same sliders appeared to adjust the recipe.



Figure 15, homescreen of Flavor Fusion

This screenshot shows a configuration page for generating a recipe. On the left side, there is a vertical list of sliders with their current settings: 'Number of people eating: 4', 'Cooking time (minutes): 60', 'Difficulty level: Medium', 'Ingredients amount: Medium', 'Protein Content: Medium', 'Carbohydrates Content: Medium', 'Fat Content: Medium', 'Budget: Medium', and 'General Health: Moderate'. Below these sliders is a section titled 'Select dietary restrictions:' with buttons for 'Vegetarian', 'Vegan', 'Gluten-Free', 'Dairy-Free', 'Nut-Free', 'Halal', and 'Kosher'. At the bottom, there is a text input field labeled 'Ingredients you have at home (separated by commas)' with a placeholder '(e.g. flour, sugar, eggs, butter, oil, salt, pepper, garlic, onions, tomatoes, etc.)' and a 'Next' button.

Figure 16, all possible options in our website of Flavor Fusion

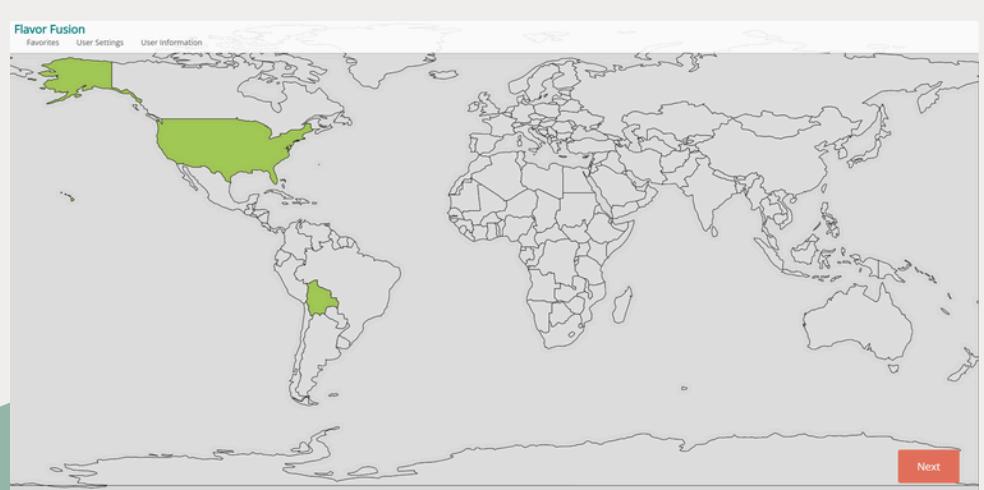


Figure 17, map to choose what culture you want to eat from

The following flowchart depicts the different iterations of 'Flavor Fusion'. Version 4 was used on the Midterm demo day and was accompanied by a cardboard bowl and scale.

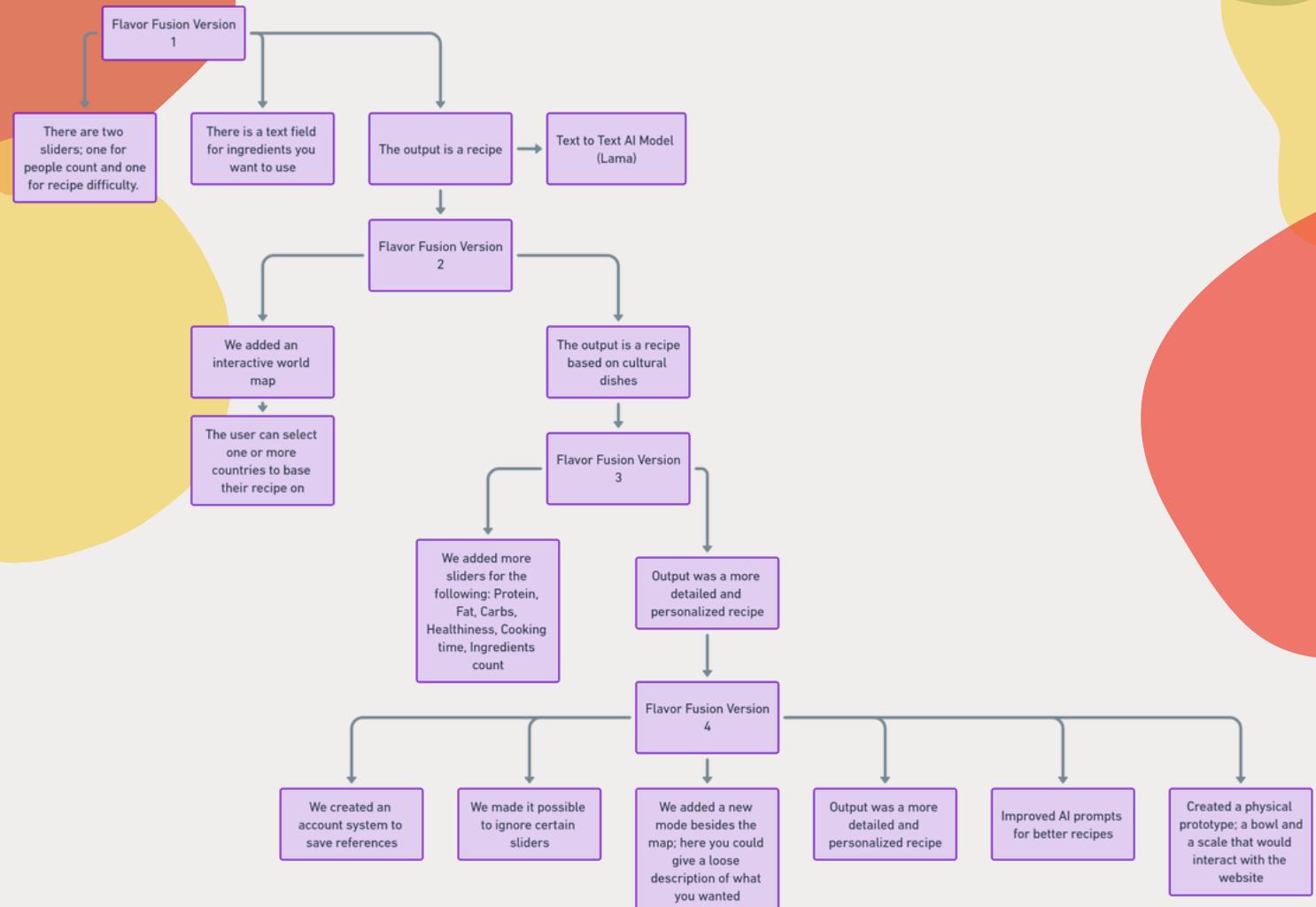


Figure 18, flowchart of Flavor Fusion

Midterm DemoDay Feedback

Version 4 of Flavor Fusion was presented during the midterm Demoday. Although people did enjoy and see a use for our product, feedback indicated that the amount of sliders and variables were overwhelming. Being a product that supposedly helps choose a recipe, having too many features and choices may have a counter effect and overwhelm the user. To avoid the experience being chaotic we came up with the idea of having an app or an account where some options could be filled in at once (duration spent, amount of diners). This would allow the user to focus on other options increasing the overall efficiency of the product. When trying to apply this to our product, wanting to have a simple and compact design hindered the inclusion of an app. Additionally, having all the options as buttons on our physical prototype could also be overwhelming. Therefore, we reduced the amount of options to the most essential ones.

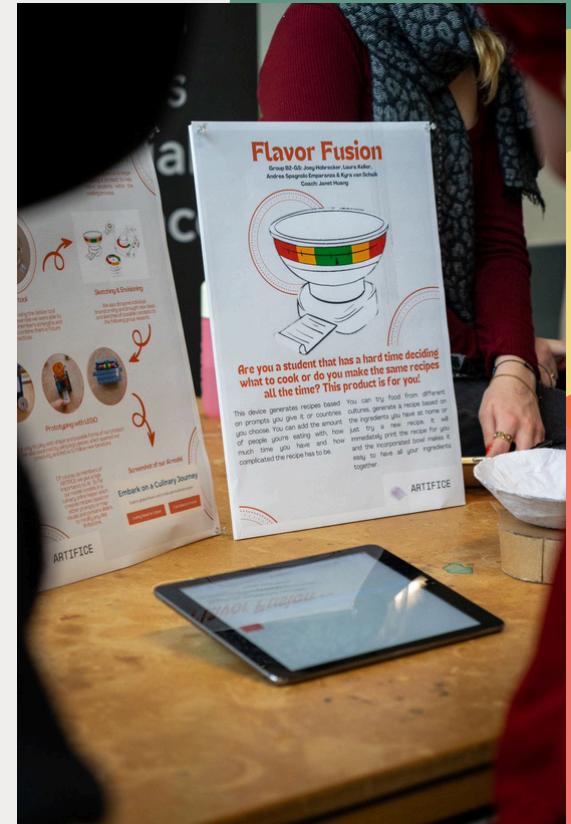


Figure 19, our stand on Midterm

Next to our website, we had a very basic physical prototype. However, for most visitors, the relation between our website and the physical prototype felt too broad. The presented prototype could generate any recipe, from any country with any ingredient you wanted. Consequently, people found our concept too broad and abstract. On top of this the connection between the concept and a bowl that measures your ingredients felt far fetched.

A visitor suggested that the bowl could be used to place ingredients that the user already has at home and recipes would be generated based off of that instead. This would give the physical aspect of our product more of a specific and useful function as well as specify our concept. The decision to change the direction of our concept was furthered when considering the insight from our initial questionnaire and the midterm Demoday proving that many people would be interested in a product that suggests recipes based on ingredients they already possess.



Figure 20 , physical prototype of the bowl

KitchenCompass

Our next iteration was a bowl with an integrated camera; users could put the ingredients they wanted to use in the bowl, the camera would detect movement and take a picture and this picture would be sent to an image-to-text AI model. That model would extract the ingredients and sent this to a text-to-text AI model that would generate a recipe with them. When testing this version we noticed that the camera was positioned too close to the ingredients and could not recognize them reliable. The distance between the components was too small. We tested it again, now with a bigger bowl, but the problem persisted. Additionally, ingredients were overlapping one another, making it difficult for the program to recognize what was put in the bowl. Our next iteration had an arm connected to the bowl, keeping the camera further above the bowl to create more distance between the ingredients and the webcam. However, this made our prototype look really big and unnatural. We sketched multiple scenarios for the webcam, for example, putting the webcam on a stand, somewhere fixed in the kitchen, or on a wearable.



Figure 21 , brainstorming about possible options for our camera

Our final iteration was the chef's hat with the camera. At first, we were unsure whether separating the webcam from the base would be ideal, however some scenarios started growing on us. A wearable would ensure that it would work in every kitchen layout and it seemed more fun and interesting than our previous ideas. We looked at different kind of wearables, like bracelets or an apron, but a hat seemed the most logical since the camera would be at a similar height as your eyes.

Additionally, placing it on your head would add a maximum distance possible for a wearable, solving our previous issue of the components being too close. Besides the distance, having a chef's hat would fit perfectly into the aesthetic of a cooking device.

In the flowchart below you can see the different iterations of KitchenCompass. This is a follow-up of the flowchart of the iterations of Flavor Fusion. The two names indicate the design stages of our product.

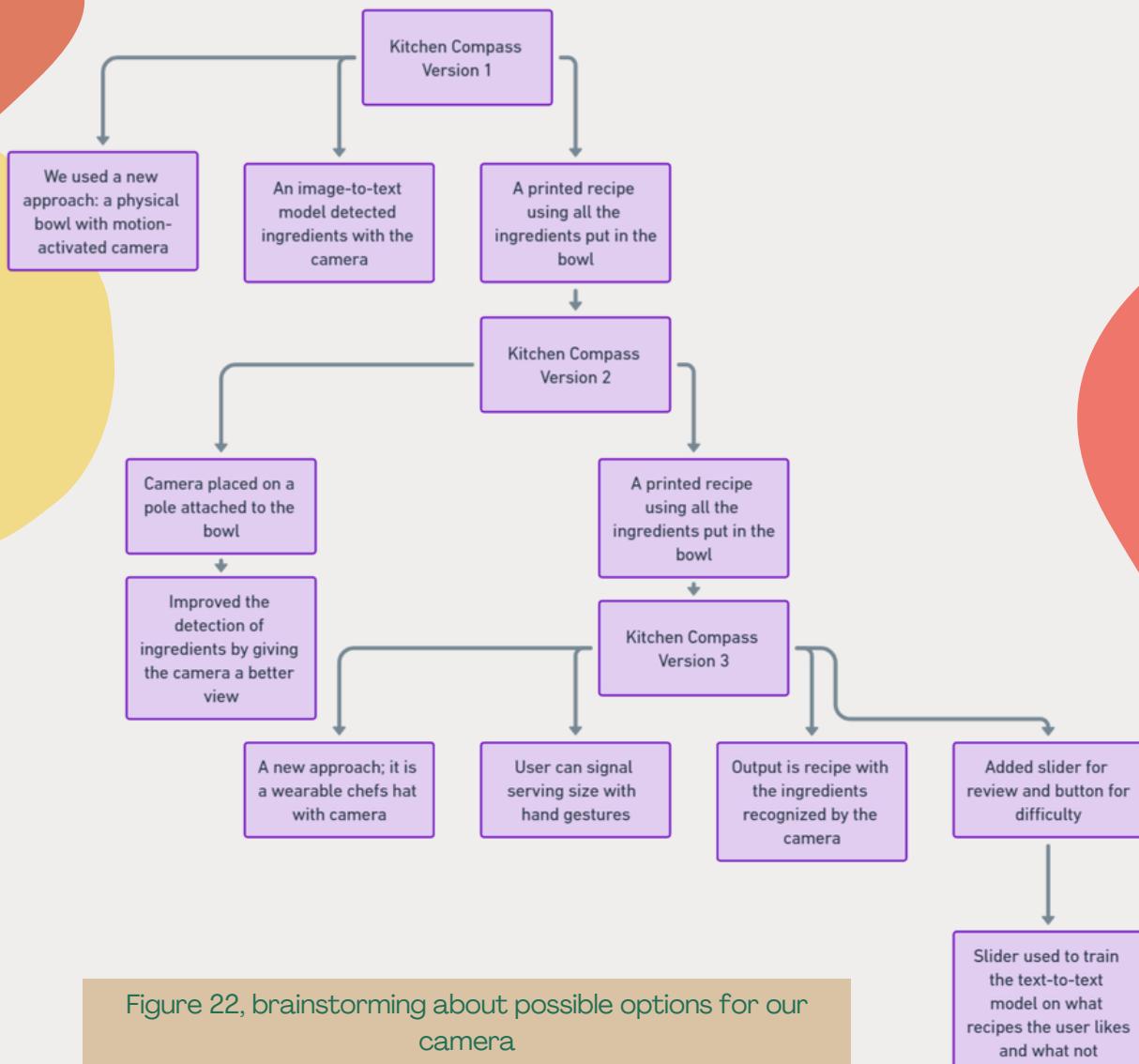


Figure 22, brainstorming about possible options for our camera

Making the Chef's hat

After choosing which wearable scenario fit our product the best, we started designing and brainstorming our chef's hat. Wanting to keep the design of a classic chef's hat, we stuck to the colour white. We utilised a thicker type of fabric so we could ensure that it would securely hold the camera in place. Additionally, the band portion of the hat was made with a double layer adding overall stability and durability. After researching various sewing patterns for chefs hat, we created our own pattern to guide us when sewing and ensure the correct proportions would be used. This consisted of one long rectangular strap and a circle with a cut to its centre. So that our product could be utilised by a wide number of people, we added an adjustable velcro closure making the hat both more comfortable and customisable.

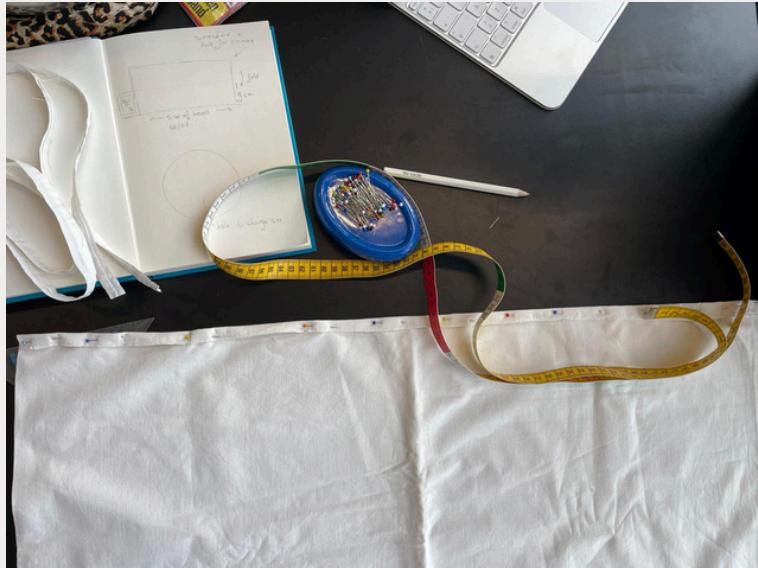


Figure 24, cutting fabric to sew the band of the hat



Figure 23, sewing the hat in the Wearable Senses Lab



Figure 25, sewing the hat in the Wearable Senses Lab

KitchenCompass

In our model with the bowl the camera was triggered by movement, therefore, when an ingredient was placed in the bowl the camera would be activated. The image-to-text model would then analyze and recognize the ingredients on the picture. When implementing the camera into a wearable, new problems arose. Utilizing this capture system on a wearable led to the webcam constantly being triggered, due to the inevitable movement of the user. As a result, we changed our code so that it would only take two pictures, triggered by the hands of the user. All ingredients would therefore be recognized from that one picture.

In this case, using a bowl to store the ingredients also proved to be an issue. As mentioned previously, ingredients overlapping each other made it impossible for the image-to-text model to recognize the ingredients.

We decided that we wanted to prioritize the function of our prototype: making sure it worked properly. Consequently we removed the bowl from our physical design replacing it with a charging station for the wearable.

Modelling and making the Base

We explored different materials for our final high-fidelity prototype. In the end the choice was between wood and 3D filament, because of time restraints we chose to 3D print it.

3D printing our product came with advantages that furthered the fidelity of our prototype. It ensured precision and allowed us to customize separate components when iterations were needed. Being able to make adjustments facilitated and made the iteration process smoother.

On top of this, printing the product provided a cleaner and therefore more professional appearance of the stand. This fit with our goal of creating something that would closely resemble a market-ready product.

Finally, considering the context in which our product would be used, the kitchen, we wanted to ensure that the materials we would utilise would be safe for food preparation, easy to clean and durable. Since it is designed for everyday use, ensuring that these specifications are met felt essential.

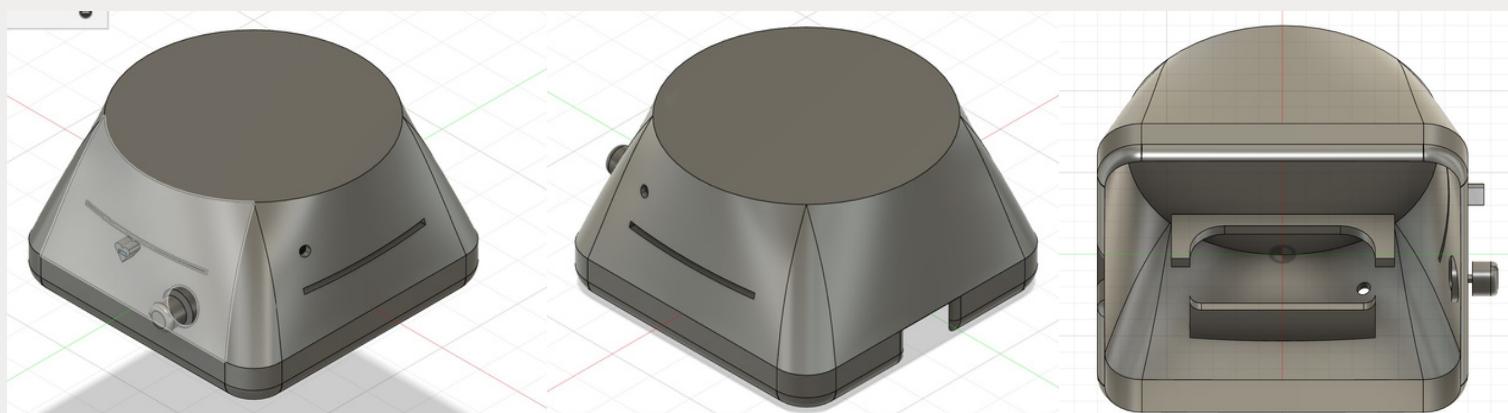


Figure 26, screenshots of final fusion 360 base

After 3D printing it we wanted to make the base as smooth as possible, to achieve the sleek and sophisticated look we wanted aesthetically, focusing on Creativity & Aesthetics. Therefore, after doing some research we found an option to mix wood filler with acetone and create a paste-like texture to cover the base (Figure). After that, we sanded it down (Figure) after we let it dry and then spray painted it on top (Figure).

The final outcome resulted to be smoother and shinier, which matched with our intentions.



Figure 27, process of smoothing the 3D-printed base and additional components



Figure 28, spray-painting the base



Figure 29, improved 3D printed outcome

OUR FINAL DESIGN

Our final concept consists of two components; a wearable chef's hat (Figure 30) with an integrated frontal camera and a 'charging port' for the hat to be placed on (Figure 31). It takes the form of an antenna, with a trapezoidal base below.

The front of this base contains a slider and a button (Figure 36). The slider is used as a rating mechanism; after going through the cooking experience, the user gives feedback. The feedback is then stored and used to mold the AI model to fit the user's preferences. The button is used to signal the level of difficulty the generated recipe should have (one press will generate a beginner's recipe, 3 presses will generate a more advanced one). Additionally, it is utilized during the final stage of the interaction to confirm the given rating. On the right hand side there is a thin horizontal gap for which the printed recipe (Figure 34) will come out, as in the interior there is a receipt printer. They can be stored and reused, giving the user the option to make their own personalized AI cookbook with their favorite recipes.



Figure 30, chef's hat with camera



Figure 31, chef's hat charging on the antenna

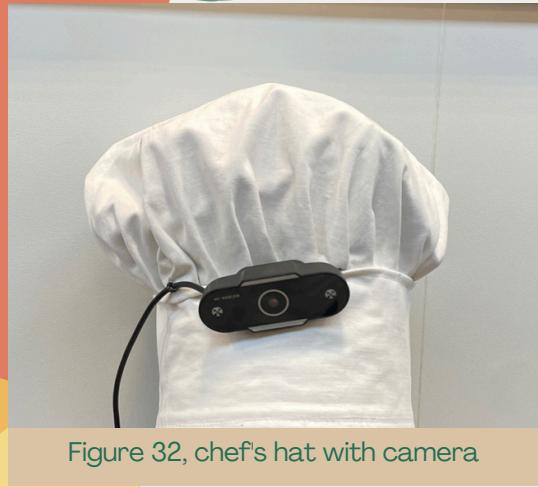


Figure 32, chef's hat with camera



Figure 33, base with antenna



Figure 34, recipe printing from the base

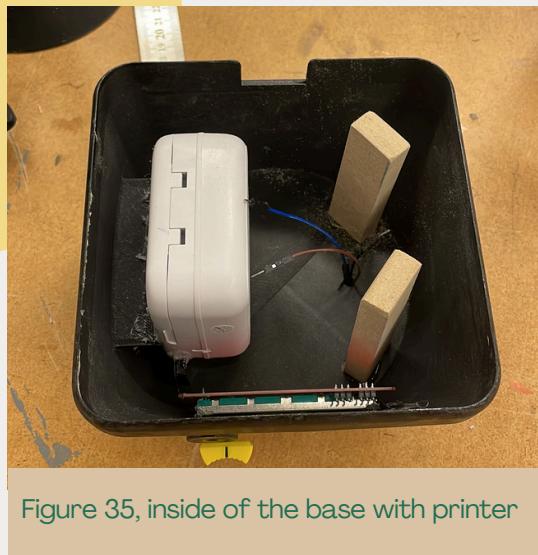


Figure 35, inside of the base with printer



Figure 36, base without the antenna

Our DemoDay stand



Figure 37,
Demoday
stand plan

For Demoday, we wanted to create a hands-on cooking experience so the user gets to interact with our product in the most accurate way as possible. Therefore, we set up a tiled Backdrop hung to the wall, and added other kitchen-like components such as an oven (gray painted box attached to the wall), hanging utensils and cloths, a pan and a bowl with food so that the users could take their pick when testing our chefs hat.

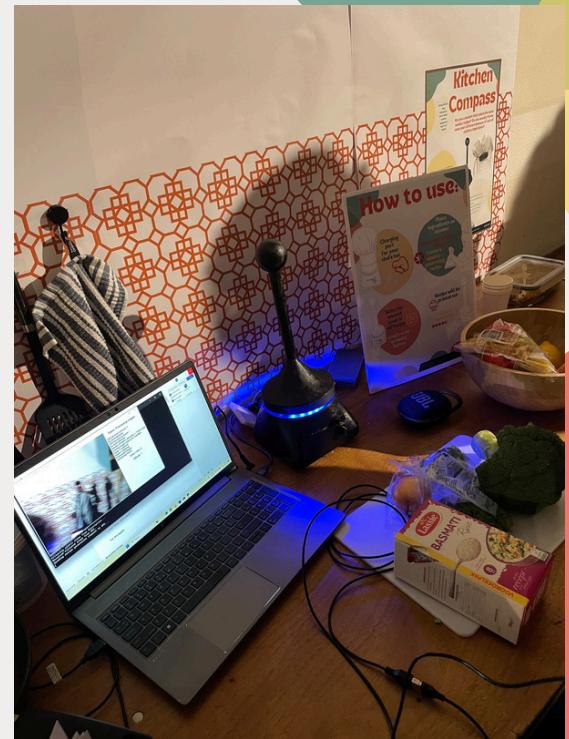


Figure 38, our Demoday stand

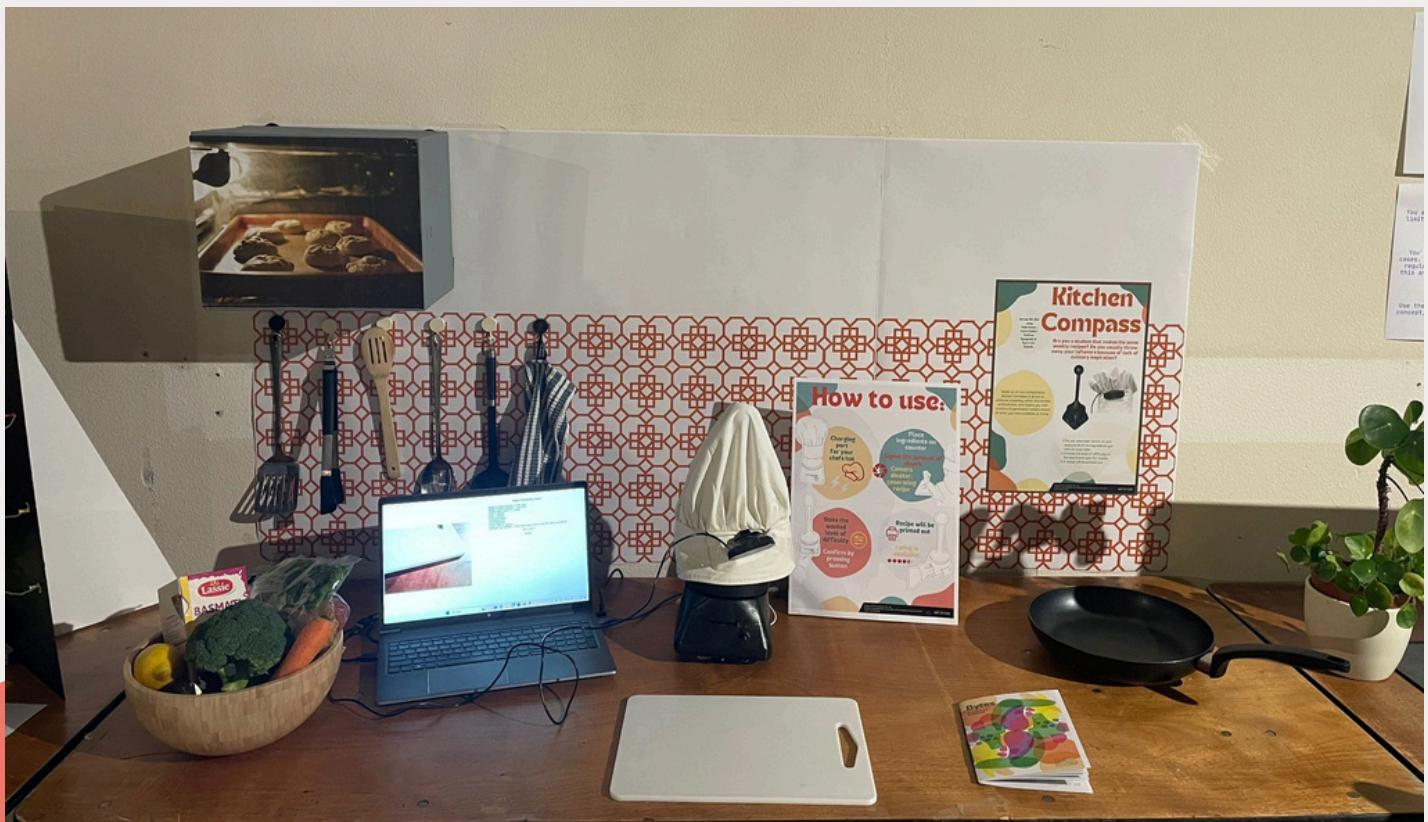


Figure 39, our full Demoday stand

DemoDay Feedback

Demoday was a busy yet insightful day, where we were very successful in having people come closer to our stand, willing to test our product. Thinking ahead of the next potential steps for KitchenCompass, we carefully listened and noted down any feedback that was given to us.

For instance, we were asked to consider any possible options or further iterations within the wearable component of our product; Is a wearable hat the best way to implement the camera? Could we explore another type of wearable that is not in the head and could be more effective? We could also think about having another device that doesn't necessarily have to be a wearable- maybe it could be attached to the user's kitchen counter.

It was noted to us that a good way forward would be to work with different modes, as the fact that our intended concept was to create recipes based on what you have available at home but the moment you set more diners the AI recipe generated gave larger food quantities, also including ingredients which were not initially in the image-to-text prompt.

Having different modes meant exploring the possibility of having a way which generates a recipe solely on what you put on the counter, and a second mode in which you state that you will go to the supermarket and so other ingredients should be included.

Another interesting piece of feedback was the idea of giving the hat more uses, such as instructing you throughout the cooking process and giving tips.

Lastly, the idea of having an app/forum for which you use the camera again to upload your cooked outcome and compare and share recipes with others was another aspect we could move forward with.

KitchenCompass Limitations

In the list of limitations, advice and feedback from the Demoday is included.

1. **Trouble recognizing ingredients:** The image-to-text model had trouble recognizing some basic ingredients and often confused things with each other. For example it would mix up apples and tomatoes as well as avocados with big black olives (since it has no sense of size). For ingredients to be recognized properly, the conditions it was working in had to be perfect. This included variable that may be difficult to control such as proper lighting.
2. **Trouble realizing what were edible ingredients:** Since the image-to-text model could recognize all components visible by the camera, it had difficulty distinguishing between edible and non-edible ingredients and would try integrate all of them into the recipe; for example after seeing a laptop in the screen, it would suggest using it as a cutting board. An other example was when seeing bricks, it gave a non-edible art recipe for cement cookies.
3. **Hard to capture the ingredients:** Due to uncontrollable variable such as height of user, it was a challenge to capture all of the ingredients and consistently had to adjust the angle of the camera. Additionally, since it was a wearable, the constant movement meant the quality of the pictures captured by the camera were limited.
4. **Not enough ingredients at home:** Since the model only captures the specific ingredient and not the quantity of them, there is a possibility of you requiring more than you currently already have. This defeats the purpose of not having to buy additional ingredients and cooking with what you have at home.
5. **Not enough reason for the hat:** After capturing the picture, the hat does not have much of a purpose. This could be an aspect we could improve and iterate.
6. **No way to give more information to the AI-model:** We did not give users the possibility to inform the AI recipe generator of certain important aspects. This includes potential allergies or cooking preferences (for example maybe you would like to use your oven). However this is something we had previously discussed and chosen not to implement to limit things being overwhelming.
7. **Hand controls did not feel natural and worked bad:** Although opinions about this was shared, some felt that the hand controls did not feel natural. Every user at the demo day needed guidance from us making it clear that our prototype may not be as intuitive as we would like it to be. Capturing the hand motion also proved to be difficult for the camera to track and capture.

Future Improvements

After the feedback that was given to us, thinking ahead we made a joined list of possible improvements or next steps that would be applied to our product. This list includes improvements suggested to us on demo day:

- 1. Explore other wearables:** we can explore other wearables with a more stable and clearer vision on the ingredients to replace the chef's hat.
- 2. Mount camera to the wall:** if we mount the camera on the wall it is stable and always looks at the same space; this makes capturing the ingredients way more reliable and the user does not have to worry if everything is in shot. Most student housing have a kitchen that is connected to the wall, so it would work for almost all students.
- 3. Use an app for community and basic settings:** combining our final product with an application on your phone has several benefits. It could be used for variables that don't change often, like allergies or cooking level. We could integrate the camera better by connecting it to the community; it could be used to take a photo of your meal and uploading it to the app for others to see and try.
- 4. Use system like ChatGPT Vision:** the newly released ChatGPT Vision, an AI system that can process video in real time, would be perfect for our product. It could be used to guide the user to the whole cooking process; it could see if you added enough spices and it could help you improve your cutting skills.
- 5. Remove hand controls and replace it with more natural interaction:** the hand controls we used to signal the amount of servings did not feel natural. We need to find ways that feel more natural like talking or gestures (like waving).
- 6. Explore using a place to lay your ingredients down:** the image-to-text AI model had a lot of trouble recognizing the ingredients. If we create more contrast using a placemat it could make it easier to recognizer compared to just the counter. For example we could use a completely white or black background.
- 7. Explore different image-to-text models:** we used a model from Facebook (llama) for ingredient recognition, for future version we need to explore different models like ChatGPT.
- 8. Explore different modes besides cooking with the ingredients you have at home:** there need to be more options for recipe generation besides using the ingredients at home. We could explore different modes like the world map we used in earlier iterations.
- 9. Explore using an image-to-video model like Sora:** we could use the newly realized Sora, a text-to-video model to generate cooking tutorials in the future. Imagine you generate a recipe and get a set by step video guide for cooking it.

GROUP DISCUSSION

Through feedback from demo day and insights gained when looking at user interactions with the product, we were able to identify various limitations and potential improvements we could make if we were to continue working on the project.

Overall our group dynamic was very positive and consistent. Since the beginning we arranged to meet weekly before or after the workshops, as out of the available time slots for this project it was challenging to find a space where we were all free. This led us to have some meeting complications in the closer weeks before Demoday, where we had to end up meeting during lunch hours.

At the beginning of this year, we all set a series of goals and skills we would like to develop within this project individually as designers. However, as time progressed we naturally steered more towards working on the group parts which we were best at/enjoyed the most. Lastly, within a design team it is important to have a good communication and, of course, be engaged and enjoy what we are doing. During this project we not only gained skills as a group and taught things to each other, yet also had fun together while doing it.



Figure 40, Group B5

Sources

1. OpenAI. (2024). DALL-E (Version 2) [AI Image Generation Tool]. Retrieved December 18, 2024, from <https://chatgpt.com/>
2. OpenAI. (2024). ChatGPT (Version 4) [AI language model]. Prompt: "Help me create python and Arduino code. The code should do the following: 1. To an Arduino a button is connected and a LED, with this button you can decide the difficulty of the recipe and the LED strip reacts to this. During the whole code the LEDs give feedback to the user, include this also include space for sound effects. 2. Track the amount of fingers of a person, use the amount of fingers that are put up as the amount of servings, when fingers are put up for 3 seconds take two screenshots. 3. Give these screenshots to an AI model, I will fill the model in, the image-to-text model filters the ingredients and gives these to a text-to-text model that uses the difficulty and the ingredients to generate a recipe, I will also include the correct model afterwards. 4. In the end, there is a slider connected to the Arduino and the LEDs need to react to this slider. The LEDs and sounds will be used as feedback for users, so include as much as possible." Retrieved December 18, 2024, from <https://chat.openai.com>

3. OpenAI. (2024). ChatGPT (Version 4) [AI language model]. Prompt: "Help me create code for three files: Sketch, Html and CSS. You need to give me the code for both complete files. We are creating a cooking AI recipe generator. Create a main page it's called Flavor Fusion. On the main page you have a bar at the top for favorites user settings and user information. In favorites the favorite recipes of the user are stored. In user settings the user can toggle slider on and off. In user information there is favorites information; this is used to improve the model over time. Each time a user favorites a recipe it will be put in here and used for future recipe generation; the text to text model needs to learn what kind of recipes the user likes here you can also find user tools here the user can toggle on and off what kind of appliances he has for example like a microwave and an oven. On the mainscreen you can choose to cook with map or cook with prompt. First i will explain cook with map. Cook with map wil bring you to an interactive map of the world. You can click on one or mulitple countries on this map. The text to text model will create a dish from this countries or a fusion of dishes from multiple countries. When you have chosen a country you come to a screen with the following sliders: Number of people eatingCooking timeDifficulty levelIngredients amountProtein contentCarbohydrates contentFat contetBudgetGeneral healthy. Here you can also selecht dietary restrions and can add ingredeints you have at home you want to use. Then you can generate the recipe this is given to an text to text model i will fill it in which specifi one. After this you see the recipe and you have a back to home button a retry button a button to save it a button to print it abutton to add it to your favorites and a button to change certain parts of the recipe if you for example dont want to use that specific ingredients make all these buttons work. The other mode is just the same only the map is replaced with a text field were you can put in the general vibe you want your food to be. " Retrieved December 18, 2024, from <https://chat.openai.com>

APPENDIX

1. Our Google Forms Questionnaire
2. The filled-in ERB form
3. The code for Flavor Fusion
4. The code for KitchenCompass
5. Our AI cookbook

Google Forms

Cooking project questionnaire

61 antwoorden

Analyse publiceren

Personal information

How old are you?

61 antwoorden

Age Group	Percentage
16 or younger	75.4%
17	23%
18	14.8%
19	42.6%
20	26.2%
21	11.5%
22 or older	8.3%

Kopiëren

Are you a student?

61 antwoorden

Status	Percentage
yes	95.1%
no	4.9%

Kopiëren

Do you enjoy what you cook (does it actually taste good?)

61 antwoorden

Response	Percentage
yes	75.4%
sometimes	23%
no	1.3%

Kopiëren

Cooking Habits

How often do you cook?

61 antwoorden

Frequency	Percentage
every day	44.3%
3-4 times	31.1%
1-2 times	14.8%
rarely	3.3%
never	8.2%

Kopiëren

What is your typical cooking time for a meal?

61 antwoorden

Time	Percentage
Less than 15 minutes	47.5%
15-30 minutes	37.7%
30-60 minutes	8.2%
1-1 1/2 hours	3.3%
1 1/2 - 2 hours	1.6%
More than 2 hours	1.6%

Kopiëren

Do you live by yourself?

61 antwoorden

Situation	Percentage
yes and I have roommates	44.3%
yes and I live alone	39.3%
no, I live at my parents house	8.2%

Kopiëren

Do you meal prep? (meal prep = making food for multiple days, so you don't have to cook every day)

61 antwoorden

Response	Percentage
yes	39.3%
rarely	34.4%
sometimes	26.2%

Kopiëren

How often do you try new food/recipes?

61 antwoorden

Frequency	Percentage
never	26.2%
rarely	29.5%
once a month	34.4%
once a week	13.1%

Kopiëren

Ingredients and Pantry

Which of these best describes your cooking skill level?

61 antwoorden

Level	Percentage
Beginner	18%
Intermediate	68.9%
Advanced	13.1%

Kopiëren

How much do you rely on pre-packaged or ready-made meals?

61 antwoorden

Reliance	Percentage
often	49.2%
occasionally	32.8%
rarely	13.1%
never	4.9%

Kopiëren

Personal questions

Kopiëren

Google Forms

<p>Do you track your nutritional intake (e.g., calories, protein, fat, etc.)?</p> <p>61 antwoorden</p> <p>● Yes, regularly ● No, occasionally ● No, never</p>	<p>Is there anything else that occasionally goes wrong or is challenging while cooking?</p> <p>20 antwoorden</p> <ul style="list-style-type: none"> Having an unreliable oven i don't measure ingredients Pas vraiment No sometimes i eat the food before finishing the cooking process. Fire :(No Kitchen is small and has few fires getting the ingredients and making the time to cook Impatient, only cook at max heat Shits messy nope Burn Sometimes when I'm cutting smth I drop my knife on my foot and somehow it just stays in vertically...that's the only thing that goes wrong though. sometimes I forget to put something in that you can't put in later like certain spices I sometimes overflow my pan with excess ingredients, I have trouble with cooking vegetables specifically, I sometimes have miscalculated with the microwave/oven and overcooked things... burning things When nobody lets me cook - no
<p>What do you usually prioritize when choosing a recipe?</p> <p>61 antwoorden</p> <p>Simplicity Healthiness Affordability Speed Taste</p>	
<p>How confident are you in measuring ingredients accurately when cooking?</p> <p>61 antwoorden</p> <p>1 2 3 4 5</p>	
<p>Where do you find new recipes?</p> <p>61 antwoorden</p> <p>Social media/websites YouTube cooking channels Cookbooks Recommendations from... I don't follow recipes most of the time I just cook Albert Heijn app AH AH verspakkelt</p>	<p>How do you usually time your cooking process (e.g., boiling, baking, frying)?</p> <p>61 antwoorden</p> <p>I use a timer (e.g., on my phone or microwave) I estimate based on experience I just check the food periodically</p>
<p>If you have to follow a recipe, what type of recipe would you prefer?</p> <p>61 antwoorden</p> <p>visual (photos - videos) written instructions audio anything is fine I think a mix of all three would be great physical with someone that gives me personal instructions I would be interested in a smart bowl with a built-in scale and timer for cooking assistance</p>	<p>Do you find it difficult to multitask while cooking (e.g., managing timing, chopping ingredients, etc.)?</p> <p>61 antwoorden</p> <p>● Yes, it's challenging ● Sometimes, depending on the recipe ● No, I'm comfortable multitasking</p>
<p>Would you be interested in a smart bowl with a built-in scale and timer for cooking assistance?</p> <p>61 antwoorden</p> <p>● Yes ● No ● Maybe</p>	<p>Which parts of the cooking process do you find most challenging?</p> <p>61 antwoorden</p> <p>Measuring ingredients Timing different parts of the meal Prepping ingredients (chopping, etc.) Cooking techniques (e.g., sautéing, grilling) Cleanup afterward Advanced cooking techniques (tempering an...)</p>

Google Forms

Would you find it useful if an app suggested recipes based on the items you already have at home? [Kopiëren](#)

61 antwoorden

Antwoord	Aantal	Percantage
Yes	53	88.5%
No	5	8.2%
Maybe	3	3.3%

Do you have anything else to add? [10 antwoorden](#)

=
no
« Couteau Fourchette »
Also think about people who go to the gym and need to know calories and macros
Nop
nope
So excited for your design!!!!
I don't know, maybe look into ethics if you are doing your project on food suggestions for people and maybe you can combine the food thing with grocery things like that AI doesn't only help with preparing food but also with cost estimations
no

Thank you for answering all our questions :)

Deze content is niet gemaakt van goedkeurd door Google. - [Servicevoorwaarden](#) - [Privacybeleid](#)
Does this form look suspicious? [Report](#)

Google Formulieren

Link to our Google Forms:
<https://forms.gle/U2yVvVdsax24hYqw5>

Ethical Review Form (Version 1.6)

This Ethical Review Form should be completed for every research study that involves human participants or personally identifiable personal data and should be submitted to ethics@tue.nl. For more information about how this process works please click [here](#).

Part 1: General Study Information

1	Project title	DFP008 Artifice: Aesthetics and Ethics of Artificial Intelligence Students enrolled in the course
2	Name of the researcher/student	Janet Huang, Stephan Wensveen, Mathias Funk,
3	Email of the researcher/student	Aaqib Saeed, Steven Houben, Jesse Benjamin, Mihnea Calota, Wessel Nieuwenhuys, Peng-kai Hung, Anna Merl, and Martijn van Loon
4	Supervisor(s) name(s) <i>Additional explanation:</i> Please write down the name of your direct supervisor. You can mention several supervisors if appropriate, but at least one supervisor should be mentioned.	y.c.huang@tue.nl, S.A.G.Wensveen@tue.nl,
5	Supervisor(s) email address(es) <i>Additional explanation:</i> Please give the email address of the supervisor(s) mentioned in question 4.	M.Funk@tue.nl, a.saeed@tue.nl, s.houben@tue.nl, j.j.benjamin@tue.nl, m.s.calota1@tue.nl, w.w.nieuwenhuys@tue.nl, p.hung@tue.nl, a.m.merl@tue.nl, m.p.m.v.loon@tue.nl Industrial Design
6	Department	<input type="checkbox"/> Yes, Bachelor. Course: DPB210, DPB220,
7	Are you a student and is this application for educational purposes?	<input checked="" type="checkbox"/> DPB240, DPB 230 (premasters) <input type="checkbox"/> Yes, Master. Course: DPM115 <input checked="" type="checkbox"/> No We will use various methods (described in #part 4.1) to collect data from people on campus and online
8	Research location <i>Additional explanation:</i> Where will the data collection take place? On campus, in a company, in public space, etc.	October 1, 2024 (as soon as approved by ERB committee)
9	Start date data collection <i>Additional explanation:</i> Please state when your data collection will start. Please note that you do not have to provide information about your complete (PhD) project, but only on this particular sub-study that you are submitting for approval in this form.	
10	End date data collection	July 31, 2025
11	Does your project receive external funding (e.g., NWO, relevant for special regulations from funders)?	<input type="checkbox"/> Yes. Name Funder: <input checked="" type="checkbox"/> No
12	Which internal and external parties are involved in the study? Think about sharing data or information between TU/e and other universities, commercial companies, hospitals, etc. <i>Additional explanation:</i> Describe all internal and external parties that are involved in the study or project, including:	Students and staff members from the TU/e will be involved in some of the studies, as participants, coaches, and experts. <ul style="list-style-type: none"> • human participants (e.g., people being interviewed, people participating in online surveys, patients, etc.); • researchers or research groups at the TU/e who participate in the study; • (Researchers at) other universities/institutions that provide data/services, help analyzing the data, etc.; • (commercial) partners, companies, government bodies, municipalities, consultancy firms, hospitals or care institutions

Ethical Review Form

	<p>that provide data (e.g., contact details of participants, data for further analysis).</p> <p>Indicate which role each party plays: who defines the means and purposes in the study, who will supply the data (external parties?), who will process/handle the data, who will be able to access the data during and after research (only researchers at TU/e or also others)?</p>	
13	Have any special agreements already been made with an external party, such as a Non-Disclosure Agreement (NDA) or a data sharing agreement?	<input type="checkbox"/> Yes, namely: <input checked="" type="checkbox"/> No
14	<p>Has your proposal already been approved by an external Ethical Review Board or Medical Ethical Review Board?</p> <p><i>Additional explanation: For example, when you are collaborating with another university and the project has been approved by their Ethical Review Board, or when you received a WMO-waiver from a Medical Ethical Review Board.</i></p>	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No
15	If yes: Please provide the name, date of approval and contact details of the ERB. Please also include the registered number for your project approval. Additionally, please send in the Ethical Review Form upon which ethical approval was granted together with this form.	
16	<p>Have you already performed a Data Protection Impact Assessment (DPIA) for this or a very similar project?</p> <p>Please read the information below: a DPIA is not the same as a regular privacy impact assessment. More detailed questions on privacy will follow in the section below.</p> <p><i>Additional explanation: A Data Protection Impact Assessment (DPIA) is a formal document that must be drafted under the guidelines of the General Data Protection Regulation (GDPR) if you process personal data that are likely to result in high privacy risks for participants. Think of research with vulnerable people, high-risk medical research, The Dutch DPA (Autoriteit Persoonsgegevens) and our website provides more information about a DPIA</i></p>	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No If yes: Please provide details about the DPIA here and send in the DPIA documentation together with this form.

Part 2: Medical study

1	<p>Does the study have a medical scientific research question or claim?</p> <p><i>Additional explanation: Medical/scientific research is research which is carried out with the aim of finding answers to a question in the field of illness and health (etiology, pathogenesis, signs/symptoms, diagnosis, prevention, outcome or treatment of illness), by systematically collecting and analyzing data. The research is carried out with the intention of contributing to medical knowledge which can also be applied to populations outside of the direct research population. If your research contains questions about health and health related parameters (such as well-being, vitality, feelings of anxiety or stress) but your research question is not primarily medical, then you can answer 'no' to this question.</i></p>	<input type="checkbox"/> Yes* <input checked="" type="checkbox"/> No *If yes or in doubt, please contact Susan Hommerson via s.m.hommerson@tue.nl
---	---	--

Part 3: Use of (medical) devices in the study

1	<p>Does your research include a device?</p> <p><i>Additional explanation: A device is a complete piece of physical hardware that is used to compute or support computer functions</i></p>	<input checked="" type="checkbox"/> Yes, not self-made <input checked="" type="checkbox"/> Yes, self-made <input type="checkbox"/> No
---	---	---

Ethical Review Form

	<i>within a larger system. Devices can be divided into input-, output-, storage-, internet of things-, or mobile device.</i>	
2	Please describe your device or link to an online description of the device	Students' prototypes will have various forms. Some students will make prototypes what are primarily operating on the laptop computers and mobile phones. Some students will also make physical prototypes with microcontrollers (e.g., Arduinos) and actuators (e.g., Screens, LED lights, sounds, vibrations, or haptic feedback) to provide users information and feedback on certain system behaviors. <input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
3	Will you use a device that is 'CE' certified for unintended use (meaning you will use existing CE certified devices for other things than they were originally intended for) or use a device that is not 'CE' certified? <i>Additional explanation: You can find more information about CE certification on https://ec.europa.eu/growth/single-market/ce-marking_en.</i>	<input type="checkbox"/> Yes <input checked="" type="checkbox"/> No
4	If yes: Do you use a device or software that has a medical purpose such as diagnosis, prevention, monitoring, prediction, prognosis, treatment or alleviation of disease or injury?	<input type="checkbox"/> Yes, my device or software currently has a medical purpose <input type="checkbox"/> Yes, my device or software could have a medical purpose in the near future <input checked="" type="checkbox"/> No <input type="checkbox"/> I'm not sure

Part 4: Information about the study

1	What are your main research questions? <i>Additional explanation: You need to provide at least one clear research question.</i>	In the course, students will apply several theories and methods to develop interactive design concepts that can allow museum visitors to experience the artificial intelligence's applications. We identify the following five topics for students' explorations. Students will select one that fits their interests and formulate a project topic with 2 to 4 other students. 1. AI for Creativity and Designing. AI technology and systems have the potential to amplify and augment humans or other living creatures' capabilities. Instead of merely passively accepting machine-generated recommendations, we should explore how AI's generative functions and massive materials can aid creators in designing and developing new ideas or things. For instance, how can an AI system support a designer's inspiration process? How can these systems make their own decisions to explore or exploit visual ideas for contributing meaningful materials? How can AI be seamlessly integrated into prototyping and making process? We would like to explore the role of AI in creativity and designing. This is not restricted to design alone, but it can also be applied to physical
---	--	---

Ethical Review Form

	<p>prototyping, creative coding, or digital craftsmanship.</p> <p>2. AI for Homecare. As the world's population ages, the challenge of providing effective healthcare intensifies. Hospitals face understaffing issues, and the increase in elderly patients is pushing healthcare systems to their limits. The need to transition healthcare towards home is more pressing than ever. This shift, while necessary, brings challenges like implementing data-enabled healthcare practices and monitoring technologies, increased workloads for clinicians and the need for patients to adapt to new technologies. This project explores how AI can revolutionize home healthcare for the chronically ill. You can think about patients suffering from heart conditions or diabetes, who still want to achieve a good quality of life and avoid hospitalization. You could for example think about integrating systems into the patients' homes. A focus on the caregiver's side could also be an interesting angle. You are encouraged to integrate AI seamlessly into home healthcare, ensuring the population receives the compassionate and efficient care they deserve. This project isn't just about technological innovation; it's about improving lives and shaping the future of home healthcare.</p> <p>3. Data as a More-than-human Material. Data is fluid and can mean different things in different contexts and different moments. Assigning meaning to data through self-reporting can greatly improve the value that we distill from it. How can we make interactions between people and their (self-reported) data more fluent and meaningful? Where and when is data created, and where does it go when it dies? What does data that is born today tell us 30 years from now?</p> <p>4. Bio Intelligence and Posthuman Design. Sensors and datasets are frequently used to enhance human experience. In an age where climate change is taking a bigger role in our everyday lives than ever before, we are looking for ways to bring other voices into the mix, such as the lives of objects and non-human living things. These non-human actors might hold important information about creating balance on earth. How can we empower the non-human things around us to sense for their own purpose? How can we better understand our more-than-human counterparts through AI?</p>
2	Description of the research method

Ethical Review Form

	<p><i>Additional explanation: For example, interview, survey, experiment, user-test, Randomized Experiment, focus groups, pilot study, observation, etc.</i></p>	<p>and evaluate the design concepts. In order to guide students to collect the data appropriately, we select the following common methods as choices for students to select and use in their projects.</p> <ol style="list-style-type: none"> 1. User Surveys: Conduct surveys to understand visitors' prior knowledge, interests, and expectations related to AI. This will help in tailoring the exhibition content to their needs. 2. Contextual Inquiry: Observe visitors as they interact with early exhibition prototypes to identify pain points, usability issues, and areas for improvement in real-world settings. 3. Card Sorting: Use card sorting exercises to determine the most intuitive and logical ways to organize exhibition content, ensuring that it aligns with visitors' mental models. 4. Focus Groups: Host focus group discussions with diverse groups of potential visitors to gather feedback on exhibition concepts, design ideas, and interactive elements. 5. Usability Testing: Conduct usability tests with interactive exhibits and prototypes to evaluate the effectiveness of user interfaces, navigation, and the overall visitor experience. 6. A/B Testing: Implement A/B testing to compare different versions of interactive exhibits and gather quantitative data on user preferences and engagement. 7. Expert Evaluation: Enlist AI experts to assess the accuracy and clarity of the information presented in the exhibition to ensure that it is both informative and credible. 8. Behavioral Analytics: Use visitor tracking and analytics tools to collect data on how visitors move through the exhibition, which exhibits they engage with the most, and how long they spend at each exhibit. 9. Post-Visit Surveys: After the visit, gather feedback from visitors to understand their overall satisfaction, what they learned, and any suggestions for improvements. 10. Longitudinal Studies: Conduct follow-up studies to assess the long-term impact of the exhibition on visitors' knowledge and attitudes towards artificial intelligence.
3	<p>Description of the research population, in- and exclusion criteria</p> <p><i>Additional explanation: Please describe which persons are eligible for your study. What criteria are used to select participants in your study, and what criteria are used to exclude possible participants? For example: We will randomly select participants from the JSF participant database with good vision and older than 18 years.</i></p>	<p>Participants will be 16 years or older and be fluent in English or Dutch because the conversational prototype will be implemented in either one of the languages. They will also have normal capabilities of hearing and eyesight to be able to perceive the non-verbal signals from the prototypes. Student groups will use methods (in #2) to collect</p>
4	<p>Description of the measurements and/or stimuli/treatments</p>	<p>qualitative and quantitative data.</p>

Ethical Review Form

	<p><u>Additional explanation:</u> Think about your outcome measures and the variables you will be collecting and describe them in a way such that another person understands what the participant will experience.</p> <p>For example: Participants will perform task A and see pictures from database B, and we measure validated Scale 1.</p>	<p>For the interviews, participants will be asked demographic questions (such as their age, gender, and occupation) and questions about their interaction with the prototypes.</p> <p>The study and procedure will be reviewed by the project coach/mentor to ensure that they adhere to this study proposal regarding both the aforementioned topics and the collection of non-sensitive data only. The only exception to non-sensitive data regards the student groups that will be conducting interviews. In this case, personal data (e.g., name, email) will be temporarily stored to arrange the interviews, and video or audio recordings can be made to allow later transcription. Those files will be deleted right after the transcription was made.</p>
5	<p>Describe and justify the number of participants you need for this study. Also justify the number of observations you need, taking into account the risks and benefits.</p> <p><u>Additional explanation:</u> Think about if you need 3 or 30 participants for example, and why? Do they need to provide their input once, or several times, and why?</p>	<p>Sample size is based on feasibility as this is an educational study. Students are expected to collect data from five to 10 persons each, or more if feasible and recommended by their mentors (student groups are formed by 3-5 students). Normally, a quantitative method will require 20 or more participants; a qualitative method might include 10 or more users. Students might also combine both quantitative and qualitative methods in their study.</p> <p>Nowadays, AI applications are becoming popular</p>
6	<p>Explain why your research is societally important. What benefits and harm to society may result from the study?</p> <p><u>Additional explanation:</u> What benefit will the results of your study have to society in general?</p>	<p>and are capable to provide momentary advice based on users' requests. Social robots, voice user interfaces, and chatbots are such examples. However, the existing AI designs often fail to understand the social contexts to behave appropriately. People are also afraid of the threaten impacts AI will bring to their works, lives, and society.</p> <p>In our squad, students will embrace the vision of human-centred AI in which AI augments rather than replaces human intelligence, we aim for humans and AI to cooperate, leverage complementary strengths, anticipate the potential harms to individuals or society, and mitigate them through careful designs. For such products or systems to be successful, the design should explain itself so that humans can understand AI's reasoning, be aware of its limitations, and wisely incorporate AI's support in their activities, especially for making important decisions. Furthermore, designers also need to understand the sociocultural context of the ethical impacts in the real world.</p>
7	<p>Describe the way participants will be recruited</p> <p><u>Additional explanation:</u> How will you recruit participants for your study? For example, by using flyers, personal network, panels, etc.</p>	<p>1. User Surveys:</p>

Ethical Review Form

	<p>2. Use personal networks to recruit participants that match the required criteria</p> <p>Contextual Inquiry:</p> <ul style="list-style-type: none"> o On-site recruitment: Approach participants and ask if they would be willing to participate in observations and interviews as they interact with exhibition prototypes. o Pre-scheduled appointments: Invite selected participants from your survey responses or focus groups to participate in more in-depth contextual inquiries. <p>3. Card Sorting:</p> <ul style="list-style-type: none"> o Use personal networks to recruit participants that match the required criteria <p>4. Focus Groups:</p> <ul style="list-style-type: none"> o Pre-screened participants: Select individuals from your survey responses or on-site recruitment who meet specific criteria (e.g., age, familiarity with AI) to participate in focus groups. o Online recruitment: Use the museum's online channels to invite interested individuals to join focus group sessions. <p>5. Usability Testing:</p> <p>Pre-scheduled appointments: Invite individuals from previous research phases who expressed interest in usability testing to participate in more structured sessions.</p> <p>6. A/B Testing:</p> <ul style="list-style-type: none"> o Random sampling: Implement A/B testing with a random sample of museum visitors who are interacting with the exhibition. Ensure the sample size is large enough for statistically meaningful results. <p>7. Expert Evaluation:</p> <ul style="list-style-type: none"> o Reach out to AI experts and researchers in academia or industry who can provide expert evaluation without the need for participant recruitment. <p>8. Behavioral Analytics:</p> <ul style="list-style-type: none"> o Utilize tracking technology to collect data on participant behaviors without the need for direct participant recruitment. <p>9. Longitudinal Studies:</p> <ul style="list-style-type: none"> o Pre-scheduled appointments: Recruit participants from previous research phases who are willing to participate in follow-up studies to assess long-term impacts.
8	<p>Provide a brief statement of the risks you expect for the participants or others involved in the study and explain. Also take into consideration any personal data you may gather and associated privacy issues.</p> <p>1. Privacy Concerns:</p> <ul style="list-style-type: none"> • Risk: Participants may be concerned about the collection of personal data during their interactions with interactive exhibits or during surveys.

Ethical Review Form

<p><i>Additional explanation: Risks for the participants can be anything from risk of data breach to risk of safety or well-being. Describe these possible risks and describe the way these risks are mitigated.</i></p>	<ul style="list-style-type: none"> • Explanation: To mitigate this risk, ensure that any data collected is anonymized and used solely for research purposes. Clearly communicate to participants how their data will be handled, stored, and protected. <p>2. Physical Safety:</p> <ul style="list-style-type: none"> • Risk: Participants may encounter physical hazards while interacting with interactive exhibits. • Explanation: Students need to have their prototypes tested by colleagues in D-Search lab. Students will follow a thorough safety assessment procedure to minimize potential risks and improve their prototypes, such as using soft surfaces and rounded edges to avoid injuries, or stabilizing the electronic circus to ensure robustness and safety, etc. <p>3. Informed Consent and Data Security:</p> <ul style="list-style-type: none"> • Risk: Participants may have concerns about the security of their personal information and whether their consent is truly informed. • Explanation: Ensure that informed consent procedures are thorough and that participants fully understand what is expected of them. Safeguard all collected data through encryption and secure storage methods described in part 7. <p>4. Participant Fatigue:</p> <ul style="list-style-type: none"> • Risk: Participants may become fatigued or disengaged during lengthy research sessions, affecting the quality of data collected. • Explanation: Keep research sessions concise and engaging. For instance, we will advise students to design the testing not longer than 30 minutes. If the study needs to take a longer time, students will provide breaks and/or non-monetary incentives to maintain participant motivation and focus.
--	--

Ethical Review Form

Part 5: Self-assessment checklist			
<i>Note: answers in the blue boxes indicate that your research is eligible for fast-track approval</i>		Yes	No
1a	Does the study involve human material? (e.g., surgery waste material derived from non-commercial organizations such as hospitals)		v
1b	Will blood or other (bio)samples be obtained from participants? (e.g., hair, sweat, urine or other bodily fluids or secretions, also external imaging of the body)		v
2	Will the participants give their consent – on a voluntary basis – either digitally or on paper? Or have they given consent in the past for the purpose of education or for re-use in line with the current research question?	V	
3	Are the participants, outside the context of the research, in a dependent or subordinate position to the investigator? <i>Additional explanation: Think about doing research on your own students or on your own employees. When there is a dependency or power imbalance between you and the research participants, you need to answer 'yes' to this question.</i>		V
4	Does the study involve participants who are particularly vulnerable or unable to give informed consent? (e.g., children (<16 years of age), people with learning difficulties, patients, people receiving counselling, people living in care or nursing homes, people recruited through self-help groups)		V
5	Will participating in the research be burdensome? (e.g., requiring participants to wear a device 24/7 for several weeks, to fill in questionnaires for hours, to travel long distances to a research location, to be interviewed multiple times)?		V
6	May the research procedure cause harm or discomfort to the participant in any way? (e.g., causing pain or more than mild discomfort, stress, anxiety or by administering drinks, foods, drugs, or showing explicit visual material)		V
7	Will financial inducement (other than reasonable expenses and compensation for time) be offered to participants? <i>Additional explanation: For an explanation of what is considered a reasonable compensation, see the topic participant fees from the HTI group</i>		V
8a	Will it be necessary for participants to take part in the study without their knowledge and consent at the time? (e.g., covert observation of people) If yes: Will you be observing people without their knowledge in public space? (e.g. on the street, at a bus-stop)		V
8b	Will the study involve actively deceiving the participants? (e.g., will participants be deliberately falsely informed, will information be withheld from them, or will they be misled in such a way that they are likely to object or show unease when debriefed about the study) Will participants be asked to discuss or report sexual experiences, religion, alcohol or drug use, suicidal thoughts, or other topics that are highly personal or intimate?		V
9			V
10			V
11	boxes in part 5. Describe how you safeguard any potential risk for the research participant.		

Ethical Review Form

Part 6: Self-assessment on privacy

The following questions (1-10) concern privacy issues, as laid down in the General Data Protection Regulation (GDPR). The Data Stewards and – if necessary – privacy team of TU/e will assess these questions. In some cases, more information is required to assess the privacy risks. If this is the case, you will be notified that the Data Stewards team will contact you.

The GDPR defines ‘personal data’ as any information relating to an identified or identifiable natural person (‘data subject’). Personal data also includes data that indirectly reveals something about a natural person. Personal data can lead to the physical, physiological, genetic, mental, economic, cultural or social identity of a natural person. There are two main categories of personal data: regular personal data and special category personal data.

If you are not sure whether some of these questions below should be answered with a Yes or No, please contact a Data Steward first through rdmsupport@tue.nl.

Note: answers in the blue boxes indicate that your research is eligible for fast-track approval

		Yes	No
1	Will the study involve discussion/collection/processing of regular personal data, or will you collect and (temporarily) store video or voice recordings for the purpose of conducting interviews? <i>Additional explanation: For example, name, address, phone number, email address, IP address, gender, age, video or interview recordings? If you are not sure whether your data contains personal data, please contact the Data Stewards Team (rdmsupport@tue.nl).</i>	V	
1A	If yes: Please describe which regular personal data you will collect in this study?	The students will collect participants' contact information (age, nationality, occupation) to analyze the demographic data. They will also collect (name, email address, and phone number) to interviews participants for clarifying some findings or conduct follow-up studies for investigating their long-term experiences of visiting the exhibition more than once.	
2	Will the study involve discussion/collection/processing of special category personal data or other sensitive data? <i>Additional explanation: Examples of special category personal data are race, religion, health information, political views, genetic or biometric data for the unique identification of a person, sexual preference, etc. Health information concerns personal data of the physical or mental health of persons, including the provision of health care. Examples of other sensitive data is information such as communication data, financial records or credit scores, camera surveillance data, location/GPS data, internet-of-things data, employee monitoring, observing or influencing behaviour, criminal records, data of vulnerable persons (children, people with disabilities, refugees), BSN number etc. Please be aware that the use of special category personal data in research requires extra security measurements in order to safeguard the privacy of data subjects and to comply with the GDPR. Processing of this special category data is prohibited, except for specific purposes and under certain circumstances. If you need to process special category data, please consult the data stewards at rdmsupport@tue.nl.</i>		V
2A	If yes: Please describe which special-category personal data and/or sensitive data you will collect in this study?		
<i>If you answered yes to either question 1 or 2, please answer the questions below. If you answered no to both questions, you can skip this part and continue onto part 6. Also, if an answer to any of the following questions is 'yes', please contact a Data Steward at rdmsupport@tue.nl</i>			
3	Will your project involve the processing of personal data on a large scale? <i>Additional explanation: In general, any processing that involves more than 10.000 data subjects should be considered "large scale". However, if the data of approximately 1000 persons (or more) are involved, the data processing may still be considered large scale. In that case, besides the number of persons involved in the study, one should also assess (i) the amount of data collected from these persons taking into account the type/risk level of the personal data, (ii) the</i>	Yes	No
3	Will your project involve the processing of personal data on a large scale? <i>Additional explanation: In general, any processing that involves more than 10.000 data subjects should be considered "large scale". However, if the data of approximately 1000 persons (or more) are involved, the data processing may still be considered large scale. In that case, besides the number of persons involved in the study, one should also assess (i) the amount of data collected from these persons taking into account the type/risk level of the personal data, (ii) the</i>	V	

Ethical Review Form

	<p>duration of the data processing, (iii) the geographic scope or extent of the processing. For example, if you would collect and process data across several European countries with 10+ socio-economic data items of 1200 individual persons for several years in a row, that is likely “large-scale processing”. Other examples of a large-scale processing activity are:</p> <ul style="list-style-type: none"> • Monitoring driving behavior of road users on Dutch highways • Collecting data of Covid patients • A hospital that processes patient data as part of its usual operations • A transport company that processes travel information of people who travel by public transport in a certain city. For example, by tracking them through travel maps. 	
4	<p>Does this processing activity involve the use of new or innovative technologies?</p> <p><i>Examples of a new technology: combining fingerprints and facial recognition for physical access control, the use of bodycams in public spaces, the use of new technical methods in conducting research such as AI. This question also refers to new technologies that have not been deployed by TU/e so far.</i></p>	V
5	<p>Does your study involve systematic (c.q. automated) monitoring of persons?</p> <p><i>Additional explanation: Consider data processing activities that have the purpose of observing, monitoring or controlling individuals, for example in circumstances where the individuals are not aware by whom their personal data is collected and how it is used. Examples of such activities are using camera systems to monitor driving behavior on highways, monitoring email inactivity or employee phone use, certain applications of machine learning and artificial intelligence.</i></p>	V
6	<p>Does the study involve collaborations (with third parties) in which data are shared or exchanged in order to link or combine data?</p> <p><i>Additional explanation: This may often apply in a collaboration between the university and a commercial party, contract research, etc. It is important to assess this for all data in the entire project, not just your own data. An important consideration in this situation is whether the person whose data is involved could have expected that data from these different databases or sources of information were to be combined. For example, it is less likely for data subjects to expect that databases from different parties will be combined and the results are used for different purposes than one could reasonably expect; this may apply for example in a collaboration between the university and a commercial party.</i></p>	V
7	<p>Will the study include data processing activities that prevent data subjects from exercising their rights or using a service or contract?</p> <p><i>Additional explanation: Examples include processing operations carried out in public places that people cannot avoid (train station, airport, shopping mall, public university premises, etc.) or processing operations whose purpose is to allow or not allow data subjects to use a service or enter into a contract (examples: by refusing to pay a benefit, not being able to apply for a loan, etc.).</i></p>	V
8	<p>Will the study process personal data to score, rank or profile persons?</p> <p><i>Additional explanation: Examples: monitoring (highway) roads to give road users a “score” based on their detected driving behavior, a bank assessing its customers based on their creditworthiness, or an organization building behavioral and marketing profiles based on use of their website or navigating their website.</i></p>	V
8	<p>Does your data processing include activities that involves composing “blacklists” – and, in particular, in relation to sensitive or special category data, such as communication data, financial records or credit scores, genetic data, biometric data, health data, camera surveillance data, location/GPS data, internet-of-things data, employee monitoring, observing or influencing behaviour, etc.</p> <p><i>Additional explanation: This situation will not be a common occurrence in research, but you may indirectly be involved in this. In general, this typically concerns processing operations involving personal data relating to criminal convictions and offences, data relating to unlawful acts, data concerning unlawful or annoying behaviour or data concerning bad payment behaviour by companies or individuals are processed and shared with third parties (blacklists or warning lists, as used, for example, by insurers, hospitality companies, shopping companies, telecom providers as well as blacklists relating to unlawful behavior of employees, for example in the healthcare sector or by employment agencies, etc.).</i></p>	V
9	<p>Will personal data be transferred or shared outside the EU/EEA?</p> <p>EU data protection rules apply to the European Economic Area (EEA), which includes all EU countries and non-EU countries Iceland, Liechtenstein and Norway.</p> <p><i>Additional explanation: The GDPR has drafted additional requirements for transfers data outside of the EU/EEA. Typically, additional safeguards must be implemented to protect the personal data of residents in the European Union. For example, if you collaborate with an American, Indian or Chinese university or other third party outside the EU/EEA, you must first check whether this is allowed and under which conditions this is allowed. Another typical example is storage of data on American providers of cloud (storage) services. Please contact the data stewards first to discuss this.</i></p>	V
10	<p>Will any raw or anonymized personal data or any other sensitive data or research results from the project possibly be transferred to a high-risk country*?</p> <p><i>*High risk countries: China, Russia, Iran, Turkey, and North Korea.</i></p>	V

Ethical Review Form

	<i>If personal data or other potentially sensitive data is exchanged with one of these countries, or if part of the data processing takes place in one of these countries: an advice from the Data Protection Officer, the kennisveiligheidsteam (Knowledge Security team), and the CISO (Chief Information Security Officer) is ALWAYS required.</i>		
--	---	--	--

Part 7: Data processing, storing and archiving

1a	<p>Is consent your legal basis for processing the personal data in your study?</p> <p><i>Additional explanation: What is a legal basis? One of main principles in the GDPR is to ensure that personal data is processed lawfully, fairly, and transparently. To comply with this principle, the processing of personal data also requires that you have a valid legal basis for the personal data processing activity.</i></p> <p><i>In research projects, the legal basis is often but not always consent. However, it is possible that it is not clear or not possible to establish whether to use consent as a legal basis.</i></p> <p><i>Some examples where consent may not be applicable as legal basis are covert research, data collection in public spaces, secondary data analysis of existing data, data that are transferred to you by a third party, consent is not possible or would require disproportionate effort, etc. In that case, please indicate which legal basis you think that applies or (preferably) contact a data steward first.</i></p>	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
1b	<p>If yes: Please briefly explain how you will obtain consent from participants and send in your draft of the information letter and consent statement together with this form. You can download a suitable template here.</p>	Before starting the survey or testing, the researcher will give the participant a consent form and explain the study. Students will use TU/e's template to make a form with their study details.
1c	<p>If no: Please briefly explain on which legal basis - other than consent - you will process the personal data in your study.</p>	
2	<p>In which way will you collect and process the (personal) data?</p> <p><i>Additional explanation: Please describe which software (e.g., LimeSurvey, Atlas Ti, Qualtrics), tools (electronic lab journals, information management systems, etc.), technologies, apps or devices (Internet-of-Things, Fitbit, etc.), techniques (monitoring, interview, survey), special data environments (e.g., Living Lab), etc. you will use to collect or process data?</i></p>	For all collected data, the responsible researcher will de-identify it in the first place. All data will be encoded with unique identifiers. Each identifier consists of the capital letter "P" and numeric index "i"—denoting the i-th participant. Personal data such as name and email address will be collected for coordination purposes, which will not be linked to the study data collection through the unique identifier and only be accessible by the student team. This data will be deleted right after the experiments and will not be used in data analysis and publications. All data will be saved in the Data Foundry
3	<p>Where will the data and in particular the personal data be stored during and after completion of the study? If you have already uploaded your Data Management Plan, you can refer to your Data Management Plan.</p> <p><i>Additional explanation: Please address the following:</i></p> <ul style="list-style-type: none"> • <i>Where will you store your data during the study and after you have completed the study? University supported-storage facilities are SURFdrive, SURF Research Drive, Ceph, departmental drives (this includes BE Project Drive), and the TU/e instance of Microsoft OneDrive. For most personal data, the use of SURF Research Drive, departmental drives (including BE Project Drive) and SURFdrive is required.</i> 	or/and the Microsoft OneDrive or/and Teams.
4	Which security measures are applied?	During the study, data is stored and access is only provided to student groups and the

Ethical Review Form

	<p><i>Additional explanation: Please address these questions:</i></p> <ul style="list-style-type: none"> • Is access to your (personal) data restricted? If yes, how (access control, password protection, etc.)? • Who will have access to the data during and after completion of the study? • Will you anonymize or pseudonymize the data? • Is Bitlocker (Windows), FileVault (Mac) or similar hard-drive encryption active on your laptop? • What will you do with the data after your project has come to results? Do you need to keep all data? <p>How long will you store the data after completion of the project, or can/will (part of) the data be destroyed?</p> <p>Will you or your supervisor want to keep the data for new or future research/reuse? Will you share (raw) data with others? If yes, how and how do you ensure that this is secure?</p> <ul style="list-style-type: none"> • If access restrictions are required during and after the study, please explain how this is arranged. 	<p>course lecturers. Data will be stored anonymously, e.g., P1.</p> <p>The recording form the interview will be deleted immediately after the transcriptions are made. Because we plan to publish students' works as research papers, all data will be retained at least for three years.</p>
--	---	---

Part 8: Closures and Signatures

1	<p>Enclosures (tick if applicable):</p> <p><input type="checkbox"/> Informed consent form;</p> <p><input type="checkbox"/> Informed consent form for other agencies when the research is conducted at a location (such as a school);</p> <p><input type="checkbox"/> Text used for ads (to find participants);</p> <p><input type="checkbox"/> Text used for debriefings;</p> <p><input type="checkbox"/> Approval other research ethics committee;</p> <p><input type="checkbox"/> The survey the participants need to complete, or a description of other measurements;</p> <p><input type="checkbox"/> Any other information which might be relevant for decision making by ERB;</p> <p><input type="checkbox"/> Data Protection Impact Assessment checked by the privacy officer</p> <p><input type="checkbox"/> Data Management Plan checked by a data steward</p>	
2	<p>Signature(s)</p> <p>Signature(s) of researcher(s)</p> <p>Date:</p> <p>Signature research supervisor (if applicable)</p> <p>Date:</p>	<p>October 1st, 2024</p> <p>Janet Huang</p> 

Code for Flavour Fusion

sketch code [3]

```
// sketch.js

let worldMap;
let selectedCountries = [];
let countryShapes = [];
let state = 'login'; // States: 'login', 'mainMenu', 'selection', 'vibesInput', 'sliders', 'result', 'account', 'favorites', 'userInfo'
let nextButton;
let peopleSlider, timeSlider, difficultySlider, ingredientsSlider;
let proteinSlider, carbSlider, fatSlider;
let budgetSlider, healthSlider;
let peopleLabel, timeLabel, difficultyLabel, ingredientsLabel;
let proteinLabel, carbLabel, fatLabel;
let budgetLabel, healthLabel;
let generateButton;
let recipeDiv;
let dietaryRestrictions = [];
let dietaryButtons = [];
let ingredientsInput;
let ingredientsInputLabel;
let loadingDiv;
let vibesInput;
let vibesInputLabel;
let resetButton;
let lastPrompt = ""; // To store the last used prompt
let lastRecipe = ""; // To store the last generated recipe

// Variables for main menu buttons
let startMapButton, startPromptButton, accountButton, favoritesButton, userInfoButton;

// Variables for account customization
let accountSettings = {
  variables: {
    people: true,
    time: true,
    difficulty: true,
    ingredientsAmount: true,
    protein: true,
    carb: true,
    fat: true,
    budget: true,
    generalHealth: true,
    dietaryRestrictions: true,
    ingredientsAtHome: true
  },
  dietaryRestrictions: [],
  availableTools: ['Oven', 'BBQ', 'Microwave', 'Blender', 'Stove'],
};

// Favorites list and their summaries
let favorites = [];
let favoriteSummaries = [];

// User tools
let userTools = ['Oven', 'BBQ', 'Microwave', 'Blender', 'Stove'];

// Tooltip for country names
let tooltip;

// Login variables
let users = {} // For demonstration purposes, using an object to store users
let currentUser = null;

function preload() {
  // Load the GeoJSON world map data
  worldMap = loadJSON('https://raw.githubusercontent.com/johan/world.geo.json/master/countries.geo.json');
}

function setup() {
  // Create a canvas (hidden by default)
  let canvas = createCanvas(windowWidth, windowHeight);
  canvas.parent('canvas-container');
  select('#canvas-container').hide();

  parseWorldMap();
}

// Setup main menu buttons
startMapButton = select('#start-map-button');
startPromptButton = select('#start-prompt-button');
accountButton = select('#account-button');
favoritesButton = select('#favorites-button');
userInfoButton = select('#user-info-button');
```

```
startMapButton.mousePressed(startWithMap);
startPromptButton.mousePressed(startWithPrompt);
accountButton.mousePressed(goToAccount);
favoritesButton.mousePressed(showFavorites);
userInfoButton.mousePressed(showUserInfo);

// Make the logo clickable to go back to home
let homeButton = select('#home-button');
homeButton.mousePressed(() => {
  // Hide all other containers
  hideAllContainers();
  // Show main menu
  select('#main-menu-container').show();
  state = 'mainMenu';
});

// Ensure navigation buttons work from any state
accountButton.mousePressed(() => {
  hideAllContainers();
  goToAccount();
});
favoritesButton.mousePressed(() => {
  hideAllContainers();
  showFavorites();
});
userInfoButton.mousePressed(() => {
  hideAllContainers();
  showUserInfo();
});

// Tooltip for country names
tooltip = createDiv("");
tooltip.addClass('country-name-tooltip');
tooltip.hide();

// Setup login screen
setupLoginScreen();
}

function windowResized() {
  resizeCanvas(windowWidth, windowHeight);
}

function draw() {
  if (state === 'selection') {
    background(220);
    drawCountries();
  }
}

function hideAllContainers() {
  select('#login-container').hide();
  select('#main-menu-container').hide();
  select('#canvas-container').hide();
  select('#interface-container').hide();
  select('#recipe-container').hide();
  select('#favorites-container').hide();
  select('#user-info-container').hide();
  select('header').hide();
}

function parseWorldMap() {
  // Parse the GeoJSON data and project coordinates
  for (let i = 0; i < worldMap.features.length; i++) {
    let country = worldMap.features[i];
    let countryName = country.properties.name;
    let geometry = country.geometry;
    let coordinates = [];

    if (geometry.type === 'Polygon') {
      coordinates = geometry.coordinates;
    } else if (geometry.type === 'MultiPolygon') {
      coordinates = geometry.coordinates.flat();
    }

    let countryPolygons = [];
    for (let j = 0; j < coordinates.length; j++) {
      let coords = coordinates[j];
      let vertices = [];
      for (let k = 0; k < coords.length; k++) {
        let lon = coords[k][0];
        let lat = coords[k][1];
        let x = map(lon, -180, 180, 0, width);
        let y = map(lat, 90, -90, 0, height);
        vertices.push(createVector(x, y));
      }
      countryPolygons.push(vertices);
    }

    countryShapes.push({
      name: countryName,
      polygons: countryPolygons
    });
  }
}
```

Code for Flavour Fusion

sketch code part 2 [3]

```
function drawCountries() {
let isAnyCountryHovered = false;
// Draw each country on the map
translate(width / 2 - width / 2, height / 2 - height / 2); // Center the map
for (let i = 0; i < countryShapes.length; i++) {
let country = countryShapes[i];
let isSelected = selectedCountries.includes(country.name);
let isHovered = false;

// Check if mouse is over the country
for (let j = 0; j < country.polygons.length; j++) {
let vertices = country.polygons[j];
if (collidePointPoly(mouseX, mouseY, vertices)) {
isHovered = true;
isAnyCountryHovered = true;
}
}

// Draw the country polygons
for (let j = 0; j < country.polygons.length; j++) {
let vertices = country.polygons[j];
if (isSelected) {
fill('#A3C858'); // Fresh Green for selected countries
} else {
fill('#E0E0E0'); // Light gray for unselected countries
}
stroke(0);
if (isHovered) {
strokeWeight(2); // Thicker outline when hovered
} else {
strokeWeight(1);
}
beginShape();
for (let k = 0; k < vertices.length; k++) {
vertex(vertices[k].x, vertices[k].y);
}
endShape(CLOSE);
}

// Display country name when hovered
if (isHovered) {
tooltip.html('<span style="color:black;">' + country.name + '</span>');
tooltip.style('left', mouseX + 'px');
tooltip.style('top', (mouseY - 40) + 'px');
tooltip.show();
}
}

if (!isAnyCountryHovered) {
tooltip.hide();
}

}

function mouseClicked() {
if (state === 'selection') {
// Handle country selection
for (let i = 0; i < countryShapes.length; i++) {
let country = countryShapes[i];
for (let j = 0; j < country.polygons.length; j++) {
let vertices = country.polygons[j];
if (collidePointPoly(mouseX, mouseY, vertices)) {
let index = selectedCountries.indexOf(country.name);
if (index === -1) {
selectedCountries.push(country.name);
} else {
selectedCountries.splice(index, 1);
}
return;
}
}
}
}

function startWithMap() {
state = 'selection';
hideAllContainers();
select('header').show();
select('#canvas-container').show();

// Reset selected countries
selectedCountries = [];

// Create 'Next' button
nextButton = createButton('Next');
nextButton.parent('canvas-container');
nextButton.addClass('action-button');
nextButton.id('next-button');
nextButton.mousePressed(goToSliders);
}

function startWithPrompt() {
state = 'vibesInput';
hideAllContainers();
select('header').show();
setupVibesInput();
}

function setupVibesInput() {
let interfaceContainer = select('#interface-container');
interfaceContainer.show();
interfaceContainer.html('');

vibesInputLabel = createDiv('Describe what kind of food you want right now:');
vibesInputLabel.parent(interfaceContainer);
vibesInputLabel.style('font-weight', '600');

vibesInput = createInput('');
vibesInput.parent(interfaceContainer);
vibesInput.style('width', '100%');

// Next button
nextButton = createButton('Next');
nextButton.parent(interfaceContainer);
nextButton.addClass('action-button');
nextButton.mousePressed(goToSlidersFromVibes);
}

function goToAccount() {
state = 'account';
hideAllContainers();
select('header').show();

let interfaceContainer = select('#interface-container');
interfaceContainer.show();
interfaceContainer.html('');

let title = createDiv('User Settings');
title.parent(interfaceContainer);
title.style('font-size', '24px');
title.style('font-weight', 'bold');

// Create topics and variables
let topics = {
'Preferences': ['people', 'time', 'difficulty', 'ingredientsAmount', 'budget', 'generalHealth', 'dietaryRestrictions', 'ingredientsAtHome'],
'Nutritional Goals': ['protein', 'carb', 'fat']
};

let variableNames = {
'people': 'Number of People Slider',
'time': 'Cooking Time Slider',
'difficulty': 'Difficulty Slider',
'ingredientsAmount': 'Ingredients Amount Slider',
'protein': 'Protein Slider',
'carb': 'Carbohydrates Slider',
'fat': 'Fat Slider',
'budget': 'Budget Slider',
'generalHealth': 'General Health Slider',
'dietaryRestrictions': 'Dietary Restrictions',
'ingredientsAtHome': 'Ingredients at Home'
};

for (let topic in topics) {
let topicDiv = createDiv(topic);
topicDiv.parent(interfaceContainer);
topicDiv.style('font-size', '20px');
topicDiv.style('font-weight', 'bold');
topicDiv.style('margin-top', '20px');

let variables = topics[topic];

for (let i = 0; i < variables.length; i++) {
let varKey = variables[i];
let varName = variableNames[varKey];

let varContainer = createDiv();
varContainer.parent(interfaceContainer);
varContainer.addClass('variable-container');

let varLabel = createSpan(varName);
varLabel.parent(varContainer);
varLabel.addClass('variable-name');
if (accountSettings.variables[varKey]) {
varLabel.addClass('included');
} else {
varLabel.addClass('excluded');
}

let toggleButton = createButton(accountSettings.variables[varKey] ? 'Exclude' : 'Include');
toggleButton.parent(varContainer);
toggleButton.addClass('toggle-button');
}
}
}
}
```

Code for Flavour Fusion

sketch code part 3 [3]

```
toggleButton.mousePressed(() => {
  accountSettings.variables[varKey] = !accountSettings.variables[varKey];
  if (accountSettings.variables[varKey]) {
    varLabel.removeClass('excluded');
    varLabel.addClass('included');
    toggleButton.html('Exclude');
  } else {
    varLabel.removeClass('included');
    varLabel.addClass('excluded');
    toggleButton.html('Include');
  }
  saveUserData();
})
}

function goToSliders() {
if (selectedCountries.length === 0) {
  alert('Please select at least one country.');
  return;
}
state = 'sliders';
select('#canvas-container').hide();
nextButton.hide();
setupSliders();
}

function goToSlidersFromVibes() {
if (vibesInput.value().trim() === "") {
  alert('Please enter a description.');
  return;
}
state = 'sliders';
select('#interface-container').hide();
setupSliders();
}

function setupSliders() {
// Hide the map and clear canvas
clear();
background(255);

// Create a container for the interface elements
let interfaceContainer = select('#interface-container');
interfaceContainer.show();
interfaceContainer.html("") // Clear any existing content

// Centered container
let slidersContainer = createDiv();
slidersContainer.parent(interfaceContainer);
slidersContainer.style('max-width', '600px');
slidersContainer.style('margin', '0 auto');

if (accountSettings.variables.people) {
  let peopleContainer = createDiv();
  peopleContainer.parent(slidersContainer);
  peopleContainer.addClass('input-container');

  peopleLabel = createSpan('Number of people eating: 4');
  peopleLabel.parent(peopleContainer);
  peopleLabel.style('font-weight', '600');

  peopleSlider = createSlider(1, 20, 4, 1);
  peopleSlider.parent(peopleContainer);
  peopleSlider.style('width', '100%');
  peopleSlider.input(updateLabels);
}

if (accountSettings.variables.time) {
  let timeContainer = createDiv();
  timeContainer.parent(slidersContainer);
  timeContainer.addClass('input-container');

  timeLabel = createSpan('Cooking time (minutes): 60');
  timeLabel.parent(timeContainer);
  timeLabel.style('font-weight', '600');

  timeSlider = createSlider(10, 240, 60, 10);
  timeSlider.parent(timeContainer);
  timeSlider.style('width', '100%');
  timeSlider.input(updateLabels);
}

if (accountSettings.variables.difficulty) {
  let difficultyContainer = createDiv();
  difficultyContainer.parent(slidersContainer);
  difficultyContainer.addClass('input-container');

  difficultyLabel = createSpan('Difficulty level: Medium');
  difficultyLabel.parent(difficultyContainer);
  difficultyLabel.style('font-weight', '600');

  difficultySlider = createSlider(1, 5, 1);
  difficultySlider.parent(difficultyContainer);
  difficultySlider.style('width', '100%');
  difficultySlider.input(updateLabels);
}

if (accountSettings.variables.protein) {
  let proteinContainer = createDiv();
  proteinContainer.parent(slidersContainer);
  proteinContainer.addClass('input-container');

  proteinLabel = createSpan('Protein Content: Medium');
  proteinLabel.parent(proteinContainer);
  proteinLabel.style('font-weight', '600');

  proteinSlider = createSlider(1, 5, 3, 1);
  proteinSlider.parent(proteinContainer);
  proteinSlider.style('width', '100%');
  proteinSlider.input(updateLabels);
}

if (accountSettings.variables.carb) {
  let carbContainer = createDiv();
  carbContainer.parent(slidersContainer);
  carbContainer.addClass('input-container');

  carbLabel = createSpan('Carbohydrates Content: Medium');
  carbLabel.parent(carbContainer);
  carbLabel.style('font-weight', '600');

  carbSlider = createSlider(1, 5, 3, 1);
  carbSlider.parent(carbContainer);
  carbSlider.style('width', '100%');
  carbSlider.input(updateLabels);
}

if (accountSettings.variables.fat) {
  let fatContainer = createDiv();
  fatContainer.parent(slidersContainer);
  fatContainer.addClass('input-container');

  fatLabel = createSpan('Fat Content: Medium');
  fatLabel.parent(fatContainer);
  fatLabel.style('font-weight', '600');

  fatSlider = createSlider(1, 5, 3, 1);
  fatSlider.parent(fatContainer);
  fatSlider.style('width', '100%');
  fatSlider.input(updateLabels);
}

if (accountSettings.variables.budget) {
  let budgetContainer = createDiv();
  budgetContainer.parent(slidersContainer);
  budgetContainer.addClass('input-container');

  budgetLabel = createSpan('Budget: Medium');
  budgetLabel.parent(budgetContainer);
  budgetLabel.style('font-weight', '600');

  budgetSlider = createSlider(1, 5, 3, 1);
  budgetSlider.parent(budgetContainer);
  budgetSlider.style('width', '100%');
  budgetSlider.input(updateLabels);
}

if (accountSettings.variables.generalHealth) {
  let healthContainer = createDiv();
  healthContainer.parent(slidersContainer);
  healthContainer.addClass('input-container');

  healthLabel = createSpan('General Health: Moderate');
  healthLabel.parent(healthContainer);
  healthLabel.style('font-weight', '600');

  healthSlider = createSlider(1, 5, 3, 1);
  healthSlider.parent(healthContainer);
  healthSlider.style('width', '100%');
  healthSlider.input(updateLabels);
}
```

Code for Flavour Fusion

sketch code part 4 [3]

```
if (accountSettings.variables.dietaryRestrictions) {
// Dietary Restrictions
let dietaryOptions = ['Vegetarian', 'Vegan', 'Gluten-Free', 'Dairy-Free', 'Nut-Free', 'Halal', 'Kosher'];
let dietaryLabel = createDiv('Select dietary restrictions:');
dietaryLabel.parent(slidersContainer);
dietaryLabel.style('font-weight', '600');

let dietaryButtonContainer = createDiv();
dietaryButtonContainer.parent(slidersContainer);
dietaryButtonContainer.addClass('dietary-buttons');
dietaryButtonContainer.style('text-align', 'center');

for (let i = 0; i < dietaryOptions.length; i++) {
let btn = createButton(dietaryOptions[i]);
btn.addClass('diet-button');
btn.parent(dietaryButtonContainer);
if (accountSettings.dietaryRestrictions.includes(dietaryOptions[i])) {
btn.addClass('active');
}
btn.mousePressed(() => toggleDietaryRestriction(dietaryOptions[i], btn));
dietaryButtons.push(btn);
}

if (accountSettings.variables.ingredientsAtHome) {
// Ingredients Input
ingredientsInputLabel = createDiv('Ingredients you have at home (separated by commas)');
ingredientsInputLabel.parent(slidersContainer);
ingredientsInputLabel.style('font-weight', '600');

ingredientsInput = createInput('');
ingredientsInput.parent(slidersContainer);
ingredientsInput.style('width', '100%');
}

// Get Inspired by Favorites
if (favorites.length > 0) {
let favoritesLabel = createDiv('Get Inspired by Favorites:');
favoritesLabel.parent(slidersContainer);
favoritesLabel.style('font-weight', '600');

let inspireButton = createButton('Inspire Me');
inspireButton.parent(slidersContainer);
inspireButton.addClass('diet-button');
if (accountSettings.inspiredByFavorites) {
inspireButton.addClass('active');
}
inspireButton.mousePressed(() => {
accountSettings.inspiredByFavorites = !accountSettings.inspiredByFavorites;
if (accountSettings.inspiredByFavorites) {
inspireButton.addClass('active');
} else {
inspireButton.removeClass('active');
}
saveUserData();
});
}

// Generate Recipe Button
generateButton = createButton('Generate Recipe');
generateButton.parent(interfaceContainer);
generateButton.addClass('action-button');
generateButton.mousePressed(() => generateRecipe());
}

function updateLabels() {
// Update labels based on slider values
let levels = ['Very Low', 'Low', 'Medium', 'High', 'Very High'];
let difficultyLevels = ['Really Easy', 'Easy', 'Medium', 'Hard', 'Really Hard'];
let ingredientsLevels = ['Very Few', 'Few', 'Medium', 'Many', 'A Lot'];
let healthLevels = ['Very Unhealthy', 'Unhealthy', 'Moderate', 'Healthy', 'Very Healthy'];

if (accountSettings.variables.people && peopleSlider) {
peopleLabel.html('Number of people eating: ' + peopleSlider.value());
}
if (accountSettings.variables.time && timeSlider) {
timeLabel.html('Cooking time (minutes): ' + timeSlider.value() + ' minutes');
}
if (accountSettings.variables.difficulty && difficultySlider) {
let difficultyIndex = difficultySlider.value() - 1;
difficultyLabel.html('Difficulty level: ' + difficultyLevels[difficultyIndex]);
}
if (accountSettings.variables.ingredientsAmount && ingredientsSlider) {
let ingredientsIndex = ingredientsSlider.value() - 1;
ingredientsLabel.html('Ingredients amount: ' + ingredientsLevels[ingredientsIndex]);
}
if (accountSettings.variables.protein && proteinSlider) {
let proteinIndex = proteinSlider.value() - 1;
proteinLabel.html('Protein Content: ' + levels[proteinIndex]);
}
}

if (accountSettings.variables.carb && carbSlider) {
let carbIndex = carbSlider.value() - 1;
carbLabel.html('Carbohydrates Content: ' + levels[carbIndex]);
}
if (accountSettings.variables.fat && fatSlider) {
let fatIndex = fatSlider.value() - 1;
fatLabel.html('Fat Content: ' + levels[fatIndex]);
}
if (accountSettings.variables.budget && budgetSlider) {
let budgetIndex = budgetSlider.value() - 1;
budgetLabel.html('Budget: ' + levels[budgetIndex]);
}
if (accountSettings.variables.generalHealth && healthSlider) {
let healthIndex = healthSlider.value() - 1;
healthLabel.html('General Health: ' + healthLevels[healthIndex]);
}

function toggleDietaryRestriction(option, button) {
let index = accountSettings.dietaryRestrictions.indexOf(option);
if (index === -1) {
accountSettings.dietaryRestrictions.push(option);
button.addClass('active');
} else {
accountSettings.dietaryRestrictions.splice(index, 1);
button.removeClass('active');
}
saveUserData();
}

function generateRecipe() {
// Collect data and create the prompt
let prompt = '';

if (state === 'sliders' && selectedCountries.length > 0) {
let selectedCountriesList = selectedCountries.join(', ');
prompt = `Generate a recipe for dinner that creates a local dish from the selected country or when multiple countries are selected creates a combination of local dishes: ${selectedCountriesList}.`;
} else if (state === 'sliders' && vibesInput) {
let vibesDescription = vibesInput.value();
prompt = `Generate a recipe that matches the following description: ${vibesDescription}.`;
}

// Include variables based on account settings
// If inspired by favorites, include the summaries
if (accountSettings.inspiredByFavorites && favoriteSummaries.length > 0) {
let summaries = favoriteSummaries.join(' ');
prompt += `The recipe should be inspired by the following: ${summaries}.`;
}

// Difficulty
if (accountSettings.variables.difficulty && difficultySlider) {
let difficultyLevels = ['Really Easy', 'Easy', 'Medium', 'Hard', 'Really Hard'];
let difficulty = difficultyLevels[difficultySlider.value() - 1];
prompt += `The recipe should have a ${difficulty} difficulty to make.`;
}

// Number of people
if (accountSettings.variables.people && peopleSlider) {
let people = peopleSlider.value();
prompt += `It should be suitable for ${people} people.`;
}

// Cooking time
if (accountSettings.variables.time && timeSlider) {
let time = timeSlider.value();
prompt += `It should take around ${time} minutes to prepare and cook.`;
}

// Ingredients amount
if (accountSettings.variables.ingredientsAmount && ingredientsSlider) {
let ingredientsLevels = ['Very Few', 'Few', 'Medium', 'Many', 'A Lot'];
let ingredientsAmount = ingredientsLevels[ingredientsSlider.value() - 1];
prompt += `It should use ${ingredientsAmount} different ingredients.`;
}

// Protein content
if (accountSettings.variables.protein && proteinSlider) {
let levels = ['Very Low', 'Low', 'Medium', 'High', 'Very High'];
let proteinContent = levels[proteinSlider.value() - 1];
prompt += `The recipe should be ${proteinContent} in protein.`;
}

// Carbohydrates content
if (accountSettings.variables.carb && carbSlider) {
let levels = ['Very Low', 'Low', 'Medium', 'High', 'Very High'];
let carbContent = levels[carbSlider.value() - 1];
prompt += `The recipe should be ${carbContent} in carbohydrates.`;
}
```

Code for Flavour Fusion

sketch code part 5 [3]

```
// Fat content
if (accountSettings.variables.fat && fatSlider) {
let levels = ['Very Low', 'Low', 'Medium', 'High', 'Very High'];
let fatContent = levels[fatSlider.value() - 1];
prompt += `The recipe should be ${fatContent} in fat.`;
}

// Budget
if (accountSettings.variables.budget && budgetSlider) {
let budgetLevels = ['Very Low', 'Low', 'Medium', 'High', 'Very High'];
let budget = budgetLevels[budgetSlider.value() - 1];
prompt += `The recipe should have a ${budget} budget.`;
}

// General Health
if (accountSettings.variables.generalHealth && healthSlider) {
let healthLevels = ['Very Unhealthy', 'Unhealthy', 'Moderate', 'Healthy', 'Very Healthy'];
let health = healthLevels[healthSlider.value() - 1];
prompt += `The recipe should be ${health} healthy.`;
}

if (accountSettings.variables.dietaryRestrictions) {
let dietaryList = accountSettings.dietaryRestrictions.join(' ') || 'No restrictions';
prompt += `It should adhere to the following dietary restrictions: ${dietaryList}.`;
}

if (accountSettings.variables.ingredientsAtHome && ingredientsInput) {
let ingredients = ingredientsInput.value() || 'Any ingredients';
prompt += `Please incorporate the following ingredients that I have at home. If there is nothing filled in after this sentence, ignore this part of the task: ${ingredients}.`;
}

// Include user tools information
let missingTools = userTools.filter(tool => !accountSettings.availableTools.includes(tool));
if (missingTools.length > 0) {
prompt += `The user does not have the following tools: ${missingTools.join(', ')}. Adjust the recipe accordingly.`;
}

// Add metric measurements
prompt += `Use European metric units (grams, liters) for measurements. Do not use imperial units.`;

// Save the prompt for display
lastPrompt = prompt;

// Show loading indicator
showLoadingIndicator();

// Send the prompt to the AI model
getRecipeFromAI(prompt);
}

function showLoadingIndicator() {
loadingDiv = createDiv('Generating your recipe, please wait...');

loadingDiv.parent('interface-container');
loadingDiv.id('loading-indicator');
}

function hideLoadingIndicator() {
if (loadingDiv) {
loadingDiv.remove();
}
}

function getRecipeFromAI(prompt) {
let api_key = "df-VVBiWkZYMedoQWNaMTRBdzMyRUtVYVNabIdJMIIfscVpKbmlMbjhwtTVuODO="; // Replace with your actual API key
let model_name = "hermes-2-pro-llama-3-8b";
let api_url = "https://data.id.tue.nl/v1/chat/completions";

fetch(api_url, {
method: "POST",
headers: {
"Content-Type": "application/json",
"Authorization": "Bearer " + api_key
},
body: JSON.stringify({
"model": model_name,
"messages": [
{"role": "user",
"content": prompt
},
{"temperature": 0.7,
"max_tokens": 500
}),
}.then((response) => response.json())
.then((json) => {
hideLoadingIndicator();
let recipe = json.choices[0].message.content.trim();
// Ensure the new recipe is different if re-trying
if (recipe === lastRecipe) {
generateRecipe(); // Re-generate if same
}
else {
lastRecipe = recipe;
displayRecipe(recipe);
}
}).catch((error) => {
hideLoadingIndicator();
console.error('Error:', error);
alert('An error occurred while generating the recipe.');
});
}

function displayRecipe(recipe) {
state = 'result';

// Hide interface elements
select('#interface-container').hide();

// Display the recipe in the recipe container
let recipeContainer = select("#recipe-container");
recipeContainer.show();
recipeContainer.html(""); // Clear any existing content

// Recipe content
let recipeContent = createDiv();
recipeContent.parent(recipeContainer);
recipeContent.id('recipe-content');

// Create buttons container
let buttonsContainer = createDiv();
buttonsContainer.parent(recipeContent);
buttonsContainer.addClass('buttons-container');

// Create a reset button
resetButton = createButton('Back to Home');
resetButton.parent(buttonsContainer);
resetButton.addClass('action-button');
resetButton.mousePressed(() => {
recipeContainer.hide();
select('#main-menu-container').show();
state = 'mainMenu';
// Reset variables
selectedCountries = [];
dietaryRestrictions = [];
vibesInput = null;
});

// Create Re-try button
retryButton = createButton('Re-try');
retryButton.parent(buttonsContainer);
retryButton.addClass('action-button');
retryButton.mousePressed(() => {
showLoadingIndicator();
generateRecipe();
});

// Create Save and Print buttons
let saveButton = createButton('Save Recipe');
saveButton.parent(buttonsContainer);
saveButton.addClass('action-button');
saveButton.mousePressed(() => {
saveRecipe(recipe);
});

let printButton = createButton('Print Recipe');
printButton.parent(buttonsContainer);
printButton.addClass('action-button');
printButton.mousePressed(() => {
printRecipe(recipe);
});

// Create Favorite button
let favoriteButton = createButton('Add to Favorites');
favoriteButton.parent(buttonsContainer);
favoriteButton.addClass('action-button');
favoriteButton.mousePressed(() => {
askFavoriteReason(recipe);
});

// Create Modify Recipe button
let modifyButton = createButton('Modify Recipe');
modifyButton.parent(buttonsContainer);
modifyButton.addClass('action-button');
modifyButton.mousePressed(() => {
modifyRecipe(recipe);
});

// Display the recipe
recipeDiv = createDiv(recipe);
recipeDiv.parent(recipeContent);
recipeDiv.style('width', '100%');
recipeDiv.style('margin', '20px auto');
recipeDiv.style('font-size', '18px');
recipeDiv.style('white-space', 'pre-wrap');
}
```

Code for Flavour Fusion

sketch code part 6 [3]

```
// Always display the prompt used
let promptDiv = createDiv('Prompt used:');
promptDiv.parent(recipeContent);
promptDiv.style('margin-top', '20px');
promptDiv.style('font-weight', 'bold');

let promptContent = createDiv(lastPrompt);
promptContent.parent(recipeContent);
promptContent.style('white-space', 'pre-wrap');
promptContent.style('background-color', '#f0f0f0');
promptContent.style('padding', '10px');
promptContent.style('border-radius', '5px');
}

function modifyRecipe(recipe) {
  let modifications = prompt('Enter the ingredients you want to remove or substitute (e.g., "I don\'t have chicken, replace it with tofu").');
  if (modifications) {
    // Create a new prompt to adjust the recipe
    let prompt = `Modify the following recipe to accommodate these changes: ${modifications}. Here is the original recipe: ${recipe}`;
    lastPrompt = prompt; // Save for display
    showLoadingIndicator();
    getRecipeFromAI(prompt);
  }
}

function saveRecipe(recipe) {
  let blob = new Blob([recipe], { type: 'text/plain' });
  let url = URL.createObjectURL(blob);
  let a = createA(url, 'recipe.txt');
  a.setAttribute('download', 'recipe.txt');
  a.style('display', 'none');
  document.body.appendChild(a);
  a.click();
  document.body.removeChild(a);
  URL.revokeObjectURL(url);
}

function printRecipe(recipe) {
  let printWindow = window.open("", "", 'height=600,width=800');
  printWindow.document.write(`<html><head><title>Print Recipe</title></head><body>${recipe}</body></html>`);
  printWindow.document.close();
  printWindow.print();
}

function askFavoriteReason(recipe) {
  let reason = prompt('What did you like about this recipe?');
  if (reason) {
    addToFavorites(recipe, reason);
    alert('Recipe added to favorites!');
  } else {
    alert('Recipe not added to favorites.');
  }
}

function addToFavorites(recipe, reason) {
  if (favorites.length >= 10) {
    alert('You have reached the maximum number of favorites.');
    return;
  }

  favorites.push(recipe);

  // Generate a summary
  let prompt = `Summarize the most important aspects of this recipe in a few sentences: ${recipe}. The user liked it because: ${reason}`;
  getSummaryFromAI(prompt);
}

function getSummaryFromAI(prompt) {
  let api_key = "df-VVBiWkZYMedoQWNNaMTRBdzMyRUtVYVNabIdJMIscVpKbmLMbjhwTTVuODO="; // Replace with your actual API key
  let model_name = "hermes-2-pro-llama-3-8b";
  let api_url = "https://data.id.tue.nl/v1/chat/completions";

  fetch(api_url, {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "Authorization": "Bearer " + api_key
    },
    body: JSON.stringify({
      "model": model_name,
      "messages": [
        {
          "role": "user",
          "content": prompt
        },
        {
          "temperature": 0.7,
          "max_tokens": 100
        }
      ],
      "temperature": 0.7,
      "max_tokens": 100
    })
  }).then((response) => response.json())
  .then((json) => {
    let summary = json.choices[0].message.content.trim();
    favoriteSummaries.push(summary);
  })
  .catch((error) => {
    console.error('Error:', error);
    alert('An error occurred while generating the summary.');
  })
}

function showFavorites() {
  state = 'favorites';
  hideAllContainers();
  select('header').show();

  let favoritesContainer = select('#favorites-container');
  favoritesContainer.show();
  favoritesContainer.html(""); // Clear previous content

  let title = createDiv('Your Favorite Recipes');
  title.parent(favoritesContainer);
  title.style('font-size', '24px');
  title.style('font-weight', 'bold');
  title.style('text-align', 'center');

  if (favorites.length === 0) {
    let message = createDiv('You have no favorite recipes yet.');
    message.parent(favoritesContainer);
  } else {
    for (let i = 0; i < favorites.length; i++) {
      let recipeDiv = createDiv();
      recipeDiv.parent(favoritesContainer);
      recipeDiv.addClass('favorite-recipe');

      let recipeTitle = createElement('h3', 'Recipe ' + (i + 1));
      recipeTitle.parent(recipeDiv);

      let recipeContent = createDiv(favorites[i]);
      recipeContent.parent(recipeDiv);
      recipeContent.style('white-space', 'pre-wrap');

      let removeButton = createButton('Remove from Favorites');
      removeButton.parent(recipeDiv);
      removeButton.addClass('action-button');
      removeButton.mousePressed(() => {
        favorites.splice(i, 1);
        favoriteSummaries.splice(i, 1);
        saveUserData();
        showFavorites(); // Refresh the favorites list
      });
    }
  }
}

function showUserInfo() {
  state = 'userInfo';
  hideAllContainers();
  select('header').show();

  let userInfoContainer = select('#user-info-container');
  userInfoContainer.show();
  userInfoContainer.html(""); // Clear previous content

  let title = createDiv('User Information');
  title.parent(userInfoContainer);
  title.style('font-size', '24px');
  title.style('font-weight', 'bold');
  title.style('text-align', 'center');

  // User Favorites Information Button
  let favoritesInfoButton = createButton('Favorites Information');
  favoritesInfoButton.parent(userInfoContainer);
  favoritesInfoButton.addClass('action-button');
  favoritesInfoButton.style('margin', '20px auto');
  favoritesInfoButton.mousePressed(showFavoritesInformation);

  // User Tools Button
  let userToolsButton = createButton('User Tools');
  userToolsButton.parent(userInfoContainer);
  userToolsButton.addClass('action-button');
  userToolsButton.style('margin', '20px auto');
  userToolsButton.mousePressed(showUserTools);

  function showFavoritesInformation() {
    let userInfoContainer = select('#user-info-container');
    userInfoContainer.html(""); // Clear previous content

    let title = createDiv('Favorites Information');
    title.parent(userInfoContainer);
    title.style('font-size', '24px');
    title.style('font-weight', 'bold');
    title.style('text-align', 'center');
  }
}
```

Code for Flavour Fusion

sketch code part 7 [3]

```
if (favoriteSummaries.length === 0) {
let message = createElement('p', 'You have no favorite summaries yet.');
message.parent(userInfoContainer);
} else {
for (let i = 0; i < favoriteSummaries.length; i++) {
let summaryDiv = createDiv();
summaryDiv.parent(userInfoContainer);
summaryDiv.addClass('favorite-recipe');

let summaryTitle = createElement('h3', 'Summary ' + (i + 1));
summaryTitle.parent(summaryDiv);
summaryContent.style('white-space', 'pre-wrap');
}
}

// Back button
let backButton = createButton('Back');
backButton.parent(userInfoContainer);
backButton.addClass('action-button');
backButton.mousePressed(showUserInfo);
}

function showUserTools() {
let userInfoContainer = select('#user-info-container');
userInfoContainer.html(""); // Clear previous content

let title = createElement('h2', 'User Tools');
title.parent(userInfoContainer);
title.style('font-size', '24px');
title.style('font-weight', 'bold');
title.style('text-align', 'center');

let toolsContainer = createDiv();
toolsContainer.parent(userInfoContainer);
toolsContainer.style('text-align', 'center');

for (let i = 0; i < userTools.length; i++) {
let tool = userTools[i];
let btn = createButton(tool);
btn.parent(toolsContainer);
btn.addClass('diet-button');
if (accountSettings.availableTools.includes(tool)) {
btn.addClass('active');
}
btn.mousePressed(() => {
if (accountSettings.availableTools.includes(tool)) {
accountSettings.availableTools.splice(accountSettings.availableTools.indexOf(tool), 1);
btn.removeClass('active');
} else {
accountSettings.availableTools.push(tool);
btn.addClass('active');
}
saveUserData();
});
}

// Back button
let backButton = createButton('Back');
backButton.parent(userInfoContainer);
backButton.addClass('action-button');
backButton.mousePressed(showUserInfo);
}

function setupLoginScreen() {
hideAllContainers();
select('#login-container').show();

let loginButton = select('#login-button');
let createAccountButton = select('#create-account-button');
let usernameInput = select('#username-input');
let passwordInput = select('#password-input');

loginButton.mousePressed(() => {
let username = usernameInput.value();
let password = passwordInput.value();
if (users[username] && users[username].password === password) {
currentUser = username;
loadUserData();
goToMainMenu();
} else {
alert('Incorrect username or password.');
}
});

createAccountButton.mousePressed(() => {
let username = usernameInput.value();
let password = passwordInput.value();
if (!username || !password) {
alert('Please enter a username and password.');
return;
}
}

if (users[username]) {
alert('Username already exists.');
} else {
users[username] = {
password: password,
settings: JSON.parse(JSON.stringify(accountSettings)),
favorites: [],
favoriteSummaries: []
};
currentUser = username;
saveUserData();
goToMainMenu();
}
});

function goToMainMenu() {
state = 'mainMenu';
hideAllContainers();
select('header').show();
select('#main-menu-container').show();
}

function saveUserData() {
if (currentUser && users[currentUser]) {
users[currentUser].settings = accountSettings;
users[currentUser].favorites = favorites;
users[currentUser].favoriteSummaries = favoriteSummaries;
}
}

function loadUserData() {
if (currentUser && users[currentUser]) {
accountSettings = users[currentUser].settings;
favorites = users[currentUser].favorites;
favoriteSummaries = users[currentUser].favoriteSummaries;
}
}
```

Code for Flavour Fusion

html code [3]

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Flavor Fusion</title>
  <!-- Include p5.js library -->
  <script src="https://cdnjs.cloudflare.com/ajax/libs/p5.js/1.6.0/p5.js"></script>
  <!-- Include p5.collide2D library for collision detection -->
  <script src="https://cdn.jsdelivr.net/npm/p5.collide2d"></script>
  <!-- Include Google Fonts -->
  <link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@400;600&family=Open+Sans&display=swap" rel="stylesheet">
  <!-- Include your custom CSS -->
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <!-- Login Screen -->
  <div id="login-container">
    <div class="login-form">
      <h1>Welcome to Flavor Fusion</h1>
      <input type="text" id="username-input" placeholder="Username">
      <input type="password" id="password-input" placeholder="Password">
      <button id="login-button">Login</button>
      <button id="create-account-button">Create New Account</button>
    </div>
  </div>

  <!-- Header -->
  <header style="display:none;">
    <div class="logo" id="home-button">Flavor Fusion</div>
    <nav>
      <button id="favorites-button">Favorites</button>
      <button id="account-button">User Settings</button>
      <button id="user-info-button">User Information</button>
    </nav>
  </header>

  <!-- Main Content -->
  <div id="main-menu-container" style="display:none;">
    <div class="hero">
      <!-- Background image will be set via CSS -->
      <div class="hero-content">
        <h1>Embark on a Culinary Journey</h1>
        <p>Explore global flavors and create personalized recipes</p>
        <div class="action-buttons">
          <button id="start-map-button">Cooking Based on Culture</button>
          <button id="start-prompt-button">Cook Based on Prompt</button>
        </div>
      </div>
    </div>
  </div>

  <!-- Containers for canvas and interface -->
  <div id="canvas-container" style="display:none;"></div>
  <div id="interface-container" style="display:none;"></div>
  <div id="recipe-container" style="display:none;"></div>
  <div id="favorites-container" style="display:none;"></div>
  <div id="user-info-container" style="display:none;"></div>

  <!-- The main content will be handled by p5.js -->
  <script src="sketch.js"></script>
</body>
</html>
```

Code for Flavour Fusion

style code part I [3]

```
/* Basic reset */
html, body {
  margin: 0;
  padding: 0;
  height: 100%;
  font-family: 'Open Sans', sans-serif;
  background-color: #FFF8E7; /* Soft Cream */
  color: #333;
}

/* Typography */
h1, h2, h3, h4, h5, h6 {
  font-family: 'Montserrat', sans-serif;
  margin: 0;
}

p {
  margin: 0;
}

/* Login Screen */
#login-container {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: #FFF8E7;
  display: flex;
  justify-content: center;
  align-items: center;
}

.login-form {
  background-color: #FFF;
  padding: 40px;
  border-radius: 10px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
  text-align: center;
}

.login-form h1 {
  margin-bottom: 30px;
  color: #007B7F; /* Deep Turquoise */
}

.login-form input {
  width: 100%;
  padding: 15px;
  margin-bottom: 20px;
  font-size: 16px;
  border: 1px solid #CCC;
  border-radius: 5px;
  box-sizing: border-box;
}

.login-form button {
  width: 100%;
  padding: 15px;
  background-color: #E2725B; /* Warm Terracotta */
  color: #FFF;
  border: none;
  border-radius: 5px;
  font-size: 18px;
  cursor: pointer;
  margin-bottom: 10px;
}

.login-form button:hover {
  background-color: #C65A43;
}

/* Header */
header {
  position: fixed;
  top: 0;
  width: 100%;
  background-color: rgba(255, 255, 255, 0.9);
  display: flex;
  align-items: center;
  padding: 10px 20px;
  box-shadow: 0 2px 4px rgba(0,0,0,0.1);
  z-index: 1000;
}

.logo {
  font-size: 24px;
  font-weight: 600;
  color: #007B7F; /* Deep Turquoise */
  cursor: pointer;
}

nav {
  margin-left: auto;
}

nav button {
  background: none;
  border: none;
  color: #333;
  font-size: 16px;
  margin-left: 20px;
  cursor: pointer;
  transition: color 0.3s;
}

nav button:hover {
  color: #007B7F;
}

/* Main Menu Styles */
#main-menu-container {
```

Code for Flavour Fusion

style code part 2 [3]

```
margin-top: 60px; /* To account for fixed header */  
}  
  
.hero {  
position: relative;  
width: 100%;  
height: calc(100vh - 60px);  
background-image: url('Background.png');  
background-size: cover;  
background-position: center;  
}  
  
.hero::after {  
content: " ";  
position: absolute;  
top: 0;  
left: 0;  
width: 100%;  
height: 100%;  
background: rgba(255, 248, 231, 0.8); /* Soft Cream overlay */  
}  
  
.hero-content {  
position: relative;  
z-index: 1;  
max-width: 800px;  
margin: 0 auto;  
padding: 100px 20px;  
text-align: center;  
}  
  
.hero-content h1 {  
font-size: 48px;  
color: #007B7F; /* Deep Turquoise */  
margin-bottom: 20px;  
}  
  
.hero-content p {  
font-size: 24px;  
color: #333;  
margin-bottom: 40px;  
}  
  
.action-buttons {  
display: flex;  
justify-content: center;  
}  
  
.action-buttons button {  
background-color: #E2725B; /* Warm Terracotta */  
color: #FFF;  
border: none;  
padding: 20px 40px;  
font-size: 22px;  
margin: 0 20px;  
border-radius: 5px;  
cursor: pointer;  
transition: background-color 0.3s;  
}  
  
.action-buttons button:hover {  
background-color: #C65A43;  
}  
  
.action-buttons button:focus {  
outline: none;  
}  
  
/* Containers */  
#canvas-container, #interface-container, #recipe-container,  
#favorites-container, #user-info-container {  
padding: 80px 20px;  
box-sizing: border-box;  
max-width: 1000px;  
margin: 0 auto;  
}  
  
#canvas-container {  
position: relative;  
height: calc(100vh - 60px);  
}  
  
/* General Buttons */  
button {  
font-family: 'Open Sans', sans-serif;  
border-radius: 5px;  
border: none;  
cursor: pointer;  
transition: background-color 0.3s;  
}  
  
button:hover {  
opacity: 0.9;  
}  
  
input[type=text], input[type=password], input[type=range],  
textarea {  
width: 100%;  
padding: 10px;  
margin-top: 10px;  
font-size: 16px;  
border: 1px solid #CCC;  
border-radius: 5px;  
box-sizing: border-box;  
}  
  
label {  
font-weight: 600;  
}  
  
/* Sliders */  
.input-container {
```

Code for Flavour Fusion

style code part 3 [3]

```
display: flex;
align-items: center;
margin-bottom: 20px;
}

.input-container label {
flex: 1;
}

.input-container input[type=range] {
flex: 2;
margin-left: 20px;
}

/* Dietary Buttons */
.diet-button {
background-color: #E0E0EO;
color: #333;
padding: 15px 25px;
margin: 5px;
border-radius: 5px;
cursor: pointer;
border: none;
font-size: 16px;
transition: background-color 0.3s;
}

.diet-button.active {
background-color: #A3C858; /* Fresh Green */
color: #FFF;
}

.diet-button:hover {
background-color: #D5D5D5;
}

/* Favorite Recipes */
.favorite-recipe {
background-color: #FFF;
border: 1px solid #CCC;
border-radius: 5px;
padding: 20px;
margin: 20px 0;
}

.favorite-recipe h3 {
margin-top: 0;
}

/* Variable Container */
.variable-container {
display: flex;
align-items: center;
margin-bottom: 10px;
}

.variable-container .variable-name {
flex: 1;
font-size: 18px;
}

.variable-container .variable-name.included {
color: #28a745; /* Green */
}

.variable-container .variable-name.excluded {
color: #dc3545; /* Red */
}

.variable-container .toggle-button {
padding: 10px 20px;
background-color: #007B7F;
color: #FFF;
border: none;
border-radius: 5px;
cursor: pointer;
}

.variable-container .toggle-button:hover {
background-color: #005f5f;
}

/* Buttons Alignment */
.action-button {
background-color: #E2725B; /* Warm Terracotta */
color: #FFF;
padding: 15px 30px;
font-size: 18px;
margin: 10px;
border-radius: 5px;
cursor: pointer;
}

.action-button:hover {
background-color: #C65A43;
}

button.action-button {
margin-top: 20px;
}

.buttons-container {
display: flex;
flex-wrap: wrap;
justify-content: center;
margin-top: 20px;
}

.buttons-container button {
margin: 10px;
}
```

Code for Flavour Fusion

style code part 4 [3]

```
/* Center Sliders and Labels */
#interface-container div {
  max-width: 600px;
  margin: 0 auto;
  text-align: left;
}

/* Country Name Tooltip */
.country-name-tooltip {
  position: absolute;
  background-color: rgba(118, 199, 192, 0.85);
  color: white;
  padding: 5px 10px;
  border-radius: 5px;
  font-weight: bold;
  pointer-events: none;
  z-index: 1000;
}

/* Adjust Tooltip Position */
.country-name-tooltip::after {
  content: "";
  position: absolute;
  bottom: 100%; /* At the top of the tooltip */
  left: 50%;
  margin-left: -5px;
  border-width: 5px;
  border-style: solid;
  border-color: transparent transparent rgba(118, 199, 192, 0.85) transparent;
}

/* Recipe Content */
#recipe-content {
  background-color: #FFF;
  border-radius: 5px;
}

#recipe-content pre, #recipe-content div {
  font-size: 18px;
  line-height: 1.6;
  color: #333;
}

/* Scrollbar Styles */
::-webkit-scrollbar {
  width: 10px;
}

::-webkit-scrollbar-track {
  background: #F1F1F1;
}

::-webkit-scrollbar-thumb {
  background: #888;
  border-radius: 5px;
}

/* Webkit scrollbar styles for sliders */
::-webkit-scrollbar-thumb:hover {
  background: #555;
}

/* Next Button on Map */
#next-button {
  position: fixed;
  bottom: 30px;
  right: 30px;
  padding: 20px 40px;
  font-size: 22px;
}

/* Loading Indicator */
#loading-indicator {
  font-size: 24px;
  text-align: center;
  margin-top: 50px;
}

/* Media Queries */
@media screen and (max-width: 768px) {
  .hero-content h1 {
    font-size: 36px;
  }

  .hero-content p {
    font-size: 20px;
  }

  .action-buttons button {
    padding: 15px 30px;
    font-size: 20px;
    margin: 0 10px;
  }

  nav button {
    font-size: 14px;
    margin-left: 10px;
  }

  header {
    flex-wrap: wrap;
  }

  .logo {
    margin-bottom: 10px;
  }

  .input-container {
    flex-direction: column;
    align-items: flex-start;
  }

  .input-container input[type=range] {
    margin-left: 0;
  }

  .buttons-container {
    flex-direction: column;
  }

  .input-container label {

```

Code for KitchenCompass

Python code [2]

```
import cv2
import mediapipe as mp
import time
import threading
import requests
import os
import platform
import subprocess
import tkinter as tk
from PIL import Image, ImageTk
import base64
import json
import queue
import serial
import sys
import pygame
import pyperclip

# Initialize Mediapipe Hand model
mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils

# API details (replace with your actual API key)
API_URL = 'https://data.id.tue.nl/v1/chat/completions'
API_KEY = 'df-
VVBiWkZYMEdoQWNaMTRBdzMyRUtVYVNabIdJMIscVpKbmMbhwTTVuODO=' # Replace with your actual API key

# Serial communication settings
arduino_port = 'COM3' # Replace with your Arduino's serial port
baud_rate = 9600

# Initialize Pygame mixer for sound playback
pygame.mixer.init()

# Global variables
number_of_fingers = 0
finger_count_start_time = 0
finger_count_hold_time = 3 # seconds
capturing = False
capture_images = []
people_count = 1 # Set based on fingers detected
last_capture_time = 0
capture_interval = 4 # seconds
num_captures = 2 # Only 2 photos

status_message = "Status: Idle"
info_messages = []
ingredients = []

slider_value = 0
waiting_for_feedback = False
difficulty_level = 1 # Default to low difficulty
system_busy = False # When True, no new hand-triggered actions

# Thread-safe queue for logging
log_queue = queue.Queue()

# Sound files
sound_files = {
    'connected': 'connected.mp3',
    'recipe_generated': 'recipe_generated.mp3',
    'start_capture': 'start_capture.mp3',
    'start_generate': 'start_generate.mp3',
    'button_press': 'button_press.mp3',
    'difficulty_set': 'difficulty_set.mp3',
    'feedback_received': 'feedback_received.mp3',
    'error_sound': 'error_sound.mp3'
}

def play_sound(sound_name, loop=False):
    if sound_name in sound_files:
        try:
            pygame.mixer.music.load(sound_files[sound_name])
            if loop:
                pygame.mixer.music.play(-1)
            else:
                pygame.mixer.music.play()
        except Exception as e:
            error_message = f"Error playing sound '{sound_name}': {e}"
            print(error_message)
            log_queue.put(error_message)
            update_info_panel()
    else:
        error_message = f"Sound '{sound_name}' not found."
        print(error_message)
        log_queue.put(error_message)
        update_info_panel()

def stop_sound():
    pygame.mixer.music.stop()

def update_info_panel():
    root.after(0, _update_info_panel)

def _update_info_panel():
    while not log_queue.empty():
        message = log_queue.get()
        info_messages.append(message)
        info_text = '\n'.join(info_messages[-10:])
        info_label.config(text=info_text)

def read_serial():
    global slider_value, waiting_for_feedback, difficulty_level, system_busy
    while True:
        try:
            line = arduino_serial.readline().decode('utf-8').strip()
            if line:
                print(f"Received from Arduino: {line}")
                if line.startswith("DIFFICULTY_LEVEL"):
                    _, level = line.split(",")
                    difficulty_level = int(level)
                    log_queue.put(f"Difficulty level received: {difficulty_level}")
                    update_info_panel()
                    difficulty_label.config(text=f"Difficulty: {difficulty_level}")
                elif line.startswith("SLIDER_VALUE"):
                    _, value = line.split(",")
                    slider_value = int(value)
                elif line == "FEEDBACK RECEIVED":
                    waiting_for_feedback = False
                    system_busy = False
                    log_queue.put("Feedback received from user. Returning to idle state.")
                    update_info_panel()
                    send_command_to_arduino("RESET")
                elif line.startswith("PLAY_SOUND"):
                    _, sound_name = line.split(",")
                    play_sound(sound_name)
                elif line.startswith("FINAL REVIEW RATING"):
                    # Final rating received if needed
                    pass
                else:
                    print(f"Unknown message from Arduino: {line}")
        except Exception as e:
            print(f"Serial read error: {e}")

def send_command_to_arduino(command):
    try:
        arduino_serial.write(f"{command}\n".encode())
        print(f"Sent command to Arduino: {command}")
    except Exception as e:
        print(f"Error sending command to Arduino: {e}")

# Initialize serial communication with Arduino
try:
    arduino_serial = serial.Serial(arduino_port, baud_rate, timeout=1)
    print(f"Connected to Arduino on port {arduino_port}")
    play_sound('connected')
except Exception as e:
    print(f"Error connecting to Arduino: {e}")
    sys.exit(1)

# Start the serial reading in a separate thread
```

Code for KitchenCompass

Python code [2]

```
serial_thread = threading.Thread(target=read_serial)
serial_thread.daemon = True
serial_thread.start()

def count_fingers(hand_landmarks):
    finger_tips_ids = [4, 8, 12, 16, 20]
    fingers = []

    # Thumb
    if hand_landmarks.landmark[finger_tips_ids[0]].x <
        hand_landmarks.landmark[finger_tips_ids[0] - 1].x:
        fingers.append(1)
    else:
        fingers.append(0)

    # Other fingers
    for id in range(1, 5):
        if hand_landmarks.landmark[finger_tips_ids[id]].y <
            hand_landmarks.landmark[finger_tips_ids[id] - 2].y:
            fingers.append(1)
        else:
            fingers.append(0)

    return fingers.count(1)

def send_image_to_api(image, callback):
    buffer = cv2.imencode('.jpg', image)
    base64_image = base64.b64encode(buffer).decode('utf-8')
    data_url = 'data:image/jpeg;base64,' + base64_image

    headers = {
        'Authorization': 'Bearer ' + API_KEY,
        'Content-Type': 'application/json',
    }

    request_body = {
        "model": "llava-llama-3-8b-v1_1",
        "messages": [
            {
                "role": "user",
                "content": [
                    {"type": "text", "text": ""},
                    {"type": "image_url", "image_url": {"url": data_url}}
                ]
            },
            "max_tokens": 500,
            "temperature": 0.9
        ]
    }

    try:
        print("Sending image to API for processing...")
        response = requests.post(API_URL, headers=headers, json=request_body)
        response.raise_for_status()
        data = response.json()
        if 'choices' in data and data['choices']:
            description = data['choices'][0]['message']['content']
            print("Received description from API.")
            callback(description)
        else:
            error_message = "API response does not contain expected data."
            print(error_message)
    except Exception as e:
        error_message = f"Error sending image to API: {e}"
        print(error_message)
        log_queue.put(error_message)
        callback("")

def filter_edible_items(description, callback):
    headers = {
        'Authorization': 'Bearer ' + API_KEY,
        'Content-Type': 'application/json',
    }

    prompt = f'Analyze the following text and extract all edible items: "{description}". Return only the edible items, separated by commas.'

    request_body = {
        "model": "hermes-2-pro-llama-3-8b",
        "messages": [
            {
                "role": "user",
                "content": prompt
            },
            "max_tokens": 150,
            "temperature": 0.7
        ]
    }

    try:
        print("Filtering edible items from description...")
        response = requests.post(API_URL, headers=headers, json=request_body)
        response.raise_for_status()
        data = response.json()
        if 'choices' in data and data['choices']:
            filtered_text = data['choices'][0]['message']['content']
            print("Received filtered items from API.")
            callback(filtered_text.strip())
        else:
            error_message = "API response does not contain expected data."
            print(error_message)
            log_queue.put(error_message)
            callback("")
    except Exception as e:
        error_message = f"Error filtering edible items: {e}"
        print(error_message)
        log_queue.put(error_message)
        callback("")

def generate_recipe(ingredients_list):
    global people_count, difficulty_level

    if not ingredients_list:
        error_message = "No ingredients available to generate a recipe."
        print(error_message)
        log_queue.put(error_message)
        update_info_panel()
        send_command_to_arduino("RESET")
        stop_sound()
        play_sound('error_sound')
        return

    combined_ingredients = ', '.join(ingredients_list)

    difficulty_map = {1: 'easy', 2: 'medium', 3: 'hard'}
    difficulty_text = difficulty_map.get(difficulty_level, 'easy')

    prompt = f"Create a {difficulty_text} recipe using the following ingredients: {combined_ingredients}. The recipe should be suitable for {people_count} people."

    headers = {
        'Authorization': 'Bearer ' + API_KEY,
        'Content-Type': 'application/json',
    }

    request_body = {
        "model": "hermes-2-pro-llama-3-8b",
        "messages": [
            {
                "role": "user",
                "content": prompt
            },
            "max_tokens": 500,
            "temperature": 0.7
        ]
    }
```

Code for KitchenCompass

Python code [2]

```
}

def recipe_thread():
    try:
        log_queue.put("Generating recipe...")
        update_info_panel()
        print("Sending recipe generation request to API...")

        response = requests.post(API_URL, headers=headers, json=request_body)
        response.raise_for_status()
        data = response.json()

        if 'choices' in data and data['choices']:
            recipe = data['choices'][0]['message']['content']
            print("Received recipe from API.")
            log_queue.put("Recipe generated.")
            update_info_panel()

            # Copy recipe to clipboard
            pyperclip.copy(recipe)
            log_queue.put("Recipe copied to clipboard.")
            update_info_panel()

            # Indicate success and wait for feedback
            send_command_to_arduino("WAITING_FOR_FEEDBACK")

            # Keep playing waiting music for 12 more seconds
            time.sleep(12)

            # Now stop waiting music and play the recipe generated sound
            stop_sound()
            play_sound('recipe_generated')

            wait_for_feedback()
        else:
            error_message = "API response does not contain expected data."
            print(error_message)
            log_queue.put(error_message)
            update_info_panel()
            send_command_to_arduino("RESET")
            stop_sound()
            play_sound('error_sound')
    except Exception as e:
        stop_sound()
        error_message = f"Error generating recipe: {e}"
        print(error_message)
        log_queue.put(error_message)
        update_info_panel()
        send_command_to_arduino("RESET")
        play_sound('error_sound')

    threading.Thread(target=recipe_thread).start()

def wait_for_feedback():
    global waiting_for_feedback
    waiting_for_feedback = True
    log_queue.put("Waiting for user feedback... Please adjust slider and then press the button on the Arduino.")

update_info_panel()

process_images()

capture_images, ingredients
ingredients = []
process_count = 0
total_images = len(capture_images)
process_count_lock = threading.Lock()

log_queue.put("Processing images...")
update_info_panel()

# Start looping the 'start_generate' sound (waiting music)
play_sound('start_generate', loop=True)

send_command_to_arduino("CAPTURE_COMPLETE")

process_image(image):
    nonlocal process_count
    def image_callback(description):
        nonlocal process_count
        if description:
            def ingredient_callback(filtered_text):
                nonlocal process_count
                with process_count_lock:
                    process_count += 1
                if filtered_text:
                    ingredients.extend([item.strip() for item in filtered_text.split(',')])

                if process_count == total_images:
                    generate_recipe(ingredients)

            filter_edible_items(description, ingredient_callback)
        else:
            with process_count_lock:
                process_count += 1
            if process_count == total_images:
                generate_recipe(ingredients)

    send_image_to_api(image, image_callback)

for img in capture_images:
    threading.Thread(target=process_image, args=(img,)).start()

capture_process():

global capturing, capture_images, last_capture_time

if capturing and len(capture_images) < num_captures:
    current_time = time.time()
    if current_time - last_capture_time >= capture_interval:
        ret, frame = cap.read()
        if ret:
            capture_images.append(frame)
            last_capture_time = current_time
            message = f"Image {len(capture_images)} captured."
            print(message)
            log_queue.put(message)
            update_info_panel()
            status_message_label.config(text=f"Status: Image {len(capture_images)} captured.")
            play_sound('start_capture')

        if len(capture_images) == num_captures:
            # Done capturing, start processing
            capturing = False
            status_message_label.config(text="Status: Processing images...")
            threading.Thread(target=process_images).start()

        else:
            error_message = "Error capturing image."
            print(error_message)
            log_queue.put(error_message)
            update_info_panel()
            send_command_to_arduino("RESET")
            stop_sound()
            play_sound('error_sound')

video_loop():

global number_of_fingers, finger_count_start_time, capturing, people_count, waiting_for_feedback, system_busy

ret, frame = cap.read()
if not ret:
    print("Cannot read frame from webcam.")
    root.after(10, video_loop)
    return

frame = cv2.flip(frame, 1)
image_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

# Only track hands if not waiting for feedback and not busy
if not waiting_for_feedback and not system_busy:
    results = hands.process(image_rgb)
else:
```

Code for KitchenCompass

Python code [2]

```
results = None

if results and results.multi_hand_landmarks and not system_busy and not waiting_for_feedback:
    hand_landmarks = results.multi_hand_landmarks[0]
    mp_drawing.draw_landmarks(frame, hand_landmarks,
mp_hands.HAND_CONNECTIONS)
    finger_count = count_fingers(hand_landmarks)

    if finger_count > 0 and not capturing:
        if number_of_fingers != finger_count:
            number_of_fingers = finger_count
            finger_count_start_time = time.time()
            message = f"Number of fingers detected: {finger_count}. Timer reset."
            print(message)
            log_queue.put(message)
            update_info_panel()
        else:
            held_time = time.time() - finger_count_start_time
            if held_time >= finger_count_hold_time:
                # Start capturing images
                capturing = True
                system_busy = True
                people_count = finger_count
                capture_images.clear()
                last_capture_time = time.time() - capture_interval
                message = f"Starting image capture for {people_count} people."
                print(message)
                log_queue.put(message)
                update_info_panel()
                status_message_label.config(text="Status: Capturing images.")
                send_command_to_arduino("START_CAPTURE")
                play_sound('start_capture')
            else:
                number_of_fingers = 0
                finger_count_start_time = 0
    else:
        if not system_busy and not waiting_for_feedback:
            message = "No hand detected."
            if not info_messages or info_messages[-1] != message:
                print(message)
                log_queue.put(message)
                update_info_panel()
            number_of_fingers = 0
            finger_count_start_time = 0

capture_process()

img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
img = Image.fromarray(img)
imgtk = ImageTk.PhotoImage(image=img)
camera_label.imgtk = imgtk
camera_label.configure(image=imgtk)

slider_label.config(text=f"Slider Value: {slider_value}")

camera_label.after(10, video_loop)

# Initialize webcam
cap = cv2.VideoCapture(0)

# Initialize Mediapipe Hands
hands = mp_hands.Hands(
    static_image_mode=False,
    max_num_hands=1,
    min_detection_confidence=0.7)

# Setup GUI with Tkinter
root = tk.Tk()
root.title("Ingredient Capture and Recipe Generator")

camera_label = tk.Label(root)
camera_label.pack(side=tk.LEFT)

info_frame = tk.Frame(root)
info_frame.pack(side=tk.RIGHT, fill=tk.BOTH, expand=True)

status_message_label = tk.Label(info_frame, text=status_message, font=('Arial', 14))
status_message_label.pack(pady=10)

info_label = tk.Label(info_frame, text="", justify=tk.LEFT, font=('Arial', 12))
info_label.pack()

slider_label = tk.Label(info_frame, text="Slider Value: 0", font=('Arial', 12))
slider_label.pack(pady=5)

difficulty_label = tk.Label(info_frame, text="Difficulty: 1", font=('Arial', 12))
difficulty_label.pack(pady=5)

video_loop()
root.mainloop()

cap.release()
```

Code for KitchenCompass

Arduino code [2]

```

#include <Adafruit_NeoPixel.h>

#define SLIDER_PIN A0
#define BUTTON_PIN 2
#define LED_PIN 6
#define NUM_LEDS 13

Adafruit_NeoPixel strip(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);

int buttonState = HIGH;
int lastButtonState = HIGH;
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;

int difficultyLevel = 1;
int sliderValue = 0;
int lastSliderValue = -1;

bool waitingForFeedback = false;
bool feedbackReceived = false;
bool sliderMoved = false;

void setup() {
  Serial.begin(9600);
  pinMode(SLIDER_PIN, INPUT);
  pinMode(BUTTON_PIN, INPUT_PULLUP);

  strip.begin();
  strip.show();
  // Idle: dim blue
  setLEDColor(strip.Color(0, 0, 50));
}

void loop() {
  handleButtonPress();

  if (waitingForFeedback) {
    readSlider();
    static int baseline = -1;
    if (lastSliderValue == -1) {
      baseline = analogRead(SLIDER_PIN);
      lastSliderValue = baseline;
      sliderMoved = false;
    } else {
      int current = sliderValue;
      if (abs(current - lastSliderValue) > 5) {
        lastSliderValue = current;
        sliderMoved = true;
      }
    }

    int rating = map(lastSliderValue, 0, 1023, 1, 10);
    int redValue = map(rating, 1, 10, 255, 0);
    int greenValue = map(rating, 1, 10, 0, 255);
    setLEDColor(strip.Color(redValue, greenValue, 0));
  }
}

receiveCommands();
delay(10);
}

void handleButtonPress() {
  unsigned long currentTime = millis();
  int reading = digitalRead(BUTTON_PIN);

  if (reading != lastButtonState) {
    lastDebounceTime = currentTime;
  }

  if ((currentTime - lastDebounceTime) > debounceDelay) {
    if (reading != buttonState) {
      buttonState = reading;

      // Button pressed
      if (buttonState == LOW) {
        if (.waitingForFeedback) {
          // Cycle difficulty levels if not waiting for feedback
          difficultyLevel++;
          if (difficultyLevel > 3) {
            difficultyLevel = 1;
          }
          Serial.print("DIFFICULTY_LEVEL,");
          Serial.println(difficultyLevel);
          setDifficultyLEDs(difficultyLevel);
          Serial.println("PLAY_SOUND,difficulty_set");
        } else {
          // If waiting for feedback, finalize rating on button press
          if (sliderMoved) {
            waitingForFeedback = false;
            feedbackReceived = true;
            int finalRating = map(lastSliderValue, 0, 1023, 1, 10);
            Serial.print("FINAL REVIEW_RATING,");
            Serial.println(finalRating);
            Serial.println("PLAY_SOUND,feedback_received");
          }
        }
      }
    }
  }
}

void receiveCommands() {
  if (Serial.available()) {
    String command = Serial.readStringUntil('\n');
    command.trim();

    if (command == "START_CAPTURE") {
      // Capturing Images: pulse yellow then solid yellow
      pulseLEDs(strip.Color(255, 255, 0), 500);
      setLEDColor(strip.Color(255, 255, 0));
    }
  }
}

```

Code for KitchenCompass

Arduino code [2]

```
}

} else if (command == "CAPTURE_COMPLETE") {
// After capturing: solid blue
setLEDColor(strip.Color(0, 0, 255));
} else if (command == "START_GENERATE") {
// Visual feedback for generation
pulseLEDs(strip.Color(0, 255, 0), 500);
setLEDColor(strip.Color(0, 255, 0));
} else if (command == "WAITING_FOR_FEEDBACK") {
waitingForFeedback = true;
feedbackReceived = false;
lastSliderValue = -1;
sliderMoved = false;
setLEDColor(strip.Color(0, 0, 0));
} else if (command == "RESET") {
feedbackReceived = false;
waitingForFeedback = false;
setLEDColor(strip.Color(0, 0, 50)); // idle
}
}

void blinkLEDs(int times, int delayTime, uint32_t color) {
for (int i = 0; i < times; i++) {
setLEDColor(color);
delay(delayTime);
setLEDColor(strip.Color(0, 0, 0));
delay(delayTime);
}
setLEDColor(strip.Color(0, 0, 50));
}

void pulseLEDs(uint32_t color, int duration) {
for (int brightness = 0; brightness <= 255; brightness += 5) {
setLEDColorDim(color, brightness);
delay(duration / 50);
}
for (int brightness = 255; brightness >= 0; brightness -= 5) {
setLEDColorDim(color, brightness);
delay(duration / 50);
}
}

}

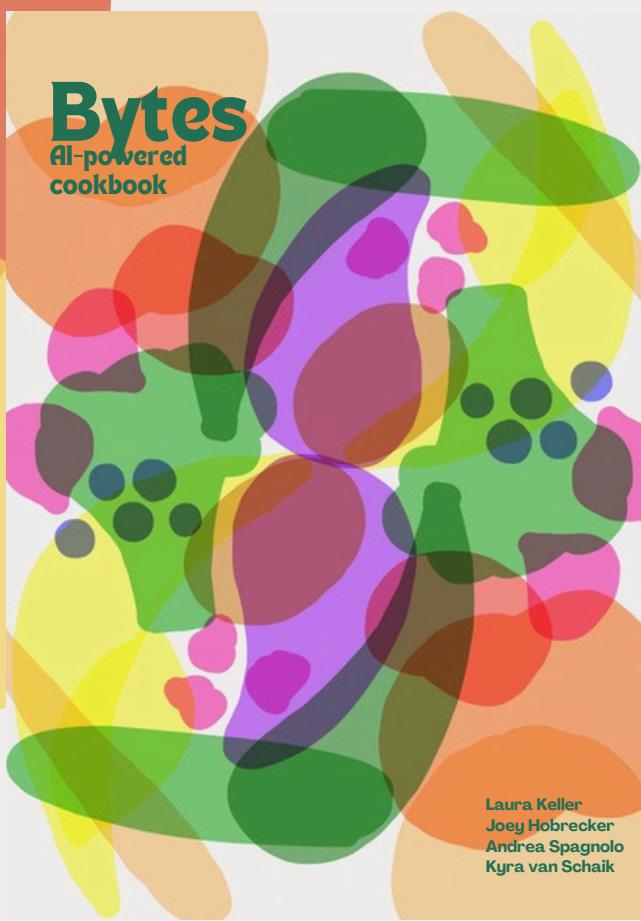
void setLEDColor(uint32_t color) {
for (int i = 0; i < strip.numPixels(); i++) {
strip.setPixelColor(i, color);
}
strip.show();
}

void setLEDColorDim(uint32_t color, int brightness) {
uint8_t r = (uint8_t)((color >> 16) & 0xFF);
uint8_t g = (uint8_t)((color >> 8) & 0xFF);
uint8_t b = (uint8_t)(color & 0xFF);

r = (r * brightness) / 255;
g = (g * brightness) / 255;
b = (b * brightness) / 255;

for (int i = 0; i < strip.numPixels(); i++) {
strip.setPixelColor(i, strip.Color(r, g, b));
}
strip.show();
}
```

Cookbook



Creamy Vegetable Pasta

Prompt to AI (Microsoft Copilot):

Can you give me a vegetarian pasta recipe with vegetables and without tomato sauce for 2 people in European measurements?

Ingredients:

- 200g pasta (penne, fusilli, or your favorite type)
- 1 tablespoon olive oil
- 1 small onion, finely chopped
- 2 cloves garlic, minced
- 1 small zucchini, diced
- 1 small bell pepper, diced
- 100g mushrooms, sliced
- 100g broccoli florets
- 100ml heavy cream
- 50g grated Parmesan cheese
- Salt and pepper, to taste
- Fresh basil or parsley, for garnish

Instructions:

Cook the pasta: Bring a large pot of salted water to a boil. Add the pasta and cook according to the package instructions until al dente. Drain and set aside.

Sauté the vegetables: In a large pan, heat the olive oil over medium heat. Add the onion and garlic, and sauté until the onion is translucent.

Add the vegetables: Add the zucchini, bell pepper, mushrooms, and broccoli to the pan.

Cook for about 5-7 minutes, until vegetables are tender. **Make the sauce:** Reduce the heat to low. Add the heavy cream to the pan and stir to combine. Let it simmer for a couple of minutes until the sauce thickens slightly. **Combine pasta and sauce:** Add the cooked pasta to the pan and toss to coat it with the sauce. Stir in the grated Parmesan cheese and season with salt and pepper to taste. **Serve:** Divide the pasta between two plates. Garnish with fresh basil or parsley and extra Parmesan cheese if desired.

AI-generated image:



Actual image:



Notes:

To create an actual sauce, you need twice the amount of heavy cream. Not sure what heavy cream was in Dutch I stir-fried the broccoli before adding it to the rest to make sure it wasn't raw. For 2 people, big portions could have added a bit more spices.

Tortilla de Verduras

Prompt to AI (ChatGPT):

Can you give me an easy, cheap dinner recipe for 2 people inspired by the Spanish culture without too much spices and with vegetables?

Ingredients:

- 4 medium-sized potatoes
- 1 medium onion
- 1 small zucchini
- 1 small red bell pepper
- 4 large eggs
- 2-3 tablespoons olive oil
- Salt and pepper to taste
- Fresh parsley

Instructions:

Prep the Vegetables: Peel the potatoes and cut them into thin slices. Dice the onion, zucchini, and bell pepper. Heat 2 tablespoons of olive oil in a large pan over medium heat. Add the potato slices and cook gently, stirring occasionally, until they're soft and slightly golden (about 10-12 minutes). Add the diced onion and bell pepper, and cook for an additional 5 minutes until the veggies are softened. Stir in the zucchini in the last 3 minutes of cooking. **Beat the Eggs:** While the veggies are cooking, crack the eggs into a bowl, season with a pinch of salt and pepper, and whisk them until well combined. Once the vegetables are done, drain any excess oil (if needed) and let them cool for a minute. Then mix them into the bowl with the eggs. Heat a bit more oil in the same pan (if necessary), and pour the egg and vegetable mixture back in. Cook on low heat for 5-7 minutes, shaking the pan gently to avoid sticking. **Flip the Tortilla:** When the edges are set but the center is still slightly runny, use a plate to help you flip the tortilla. Slide it back into the pan and cook for another 4-5 minutes until fully set. **Serve:** Let the tortilla cool slightly before slicing. Garnish with fresh parsley if you like.

AI-generated image:



Actual image:



Notes:

It took way longer than the recipe said
The measurements weren't right, not enough egg for the amount of vegetables + potatoes
The flipping is really hard > not an easy recipe
The taste was very plain > not tasty
It was way too much for 2 people
It was very cheap > 2.19 pp (excl. salad)

Pasta with Tomato Sauce

Prompt to AI (Our AI model Prototype):

Can you give me a recipe with pasta, meat and tomato sauce?

Ingredients:

- 300g pasta (e.g., penne, fusilli, or spaghetti)
- 200g ground turkey or chicken
- 2 medium-sized zucchini, diced
- 1 medium-sized red bell pepper, diced
- 1 medium-sized onion, diced
- 2 cloves of garlic, minced
- 1 cup (200g) cherry tomatoes, halved
- 2 cups (500g) tomato sauce
- 2 tbsp (30g) olive oil
- Salt and pepper to taste
- Optional: fresh basil, grated Parmesan cheese

Instructions:

Begin by cooking the pasta according to the package instructions in a large pot of salted water. While the pasta is cooking, you can start preparing the rest of the dish. In a large skillet or frying pan, heat 1 tbsp of olive oil over medium heat. Add the diced onion and cook for 3-4 minutes until softened. Add the minced garlic and cook for another minute until fragrant. Add the ground turkey or chicken to the skillet, breaking it up with a spoon. Cook for 5-7 minutes until the meat is browned and cooked through. While the meat is cooking, heat the remaining 1 tbsp of olive oil in a separate pan over medium heat. Add the diced zucchini and red bell pepper and cook for 5-7 minutes until the vegetables are tender but still have a bit of crunch. Add the cherry tomatoes to the zucchini and bell pepper pan and cook for another 2-3 minutes until the tomatoes are slightly softened. Once the meat is cooked, add the cooked pasta to the skillet with the meat and stir to combine. Add the cooked vegetables and cherry tomatoes to the skillet with the pasta and meat, along with the tomato sauce. Stir well to combine all the ingredients. Season the pasta dish with salt and pepper to taste. If desired, add some fresh basil and grated Parmesan cheese for extra flavor. Serve the pasta dish immediately in 3 portions, garnishing with fresh basil and Parmesan cheese.

AI-generated image:



Actual image:



Notes:

I added an orange bell pepper, I used hamster pasta and I switched the chicken with red meat. There was a lot of zucchini compared to the bell pepper and tomatoes. It was not saucy (not enough tomato sauce), but you still had the taste of tomato sauce. It was really tasty! I feel like using a different pan for the meat and vegetables was not necessarily needed. You could eat it with 4 people.

Cookbook

Pizza

Question to AI (Microsoft Copilot):

Can you give me a pizza recipe for 2 people. I want vegetables and some meat on it. I already have the pizza bases. They contain zucchini.

Ingredients:

- 2 pre-made pizza bases with zucchini
- 1 cup pizza sauce
- 2 cups shredded mozzarella cheese
- 1 small bell pepper, sliced
- 1 small red onion, sliced
- 5-6 mushrooms, sliced
- 8-10 cherry tomatoes, halved
- 1-2 Italian sausages, cooked and sliced
- 1/4 cup sliced black olives
- Fresh basil leaves (about 10 leaves)
- 1 tsp dried oregano
- 1 tsp dried basil
- Salt and pepper to taste
- 1 tbsp olive oil

Instructions:

- Preheat your oven to 220°C (425°F).
- Prepare the pizza bases: Place the zucchini pizza bases on a baking sheet or pizza stone. **Spread the sauce:** Evenly spread the pizza sauce over each base.
- **Add the cheese:** Sprinkle a generous amount of shredded mozzarella cheese over the sauce.
- **Layer the toppings:**
 - Arrange the Italian sausages, bell peppers, onions, mushrooms, cherry tomatoes, and black olives over the cheese. Add the cooked Italian sausage slices and black olives.
 - Season:** Sprinkle dried oregano, dried basil, salt, and pepper over the toppings.
 - Bake:** Drizzle a little olive oil over the pizza and bake in the preheated oven for about 12-15 minutes, or until the cheese is melted and bubbly, and the edges of the pizza are golden brown.
 - Garnish:** Once out of the oven, top with fresh basil leaves.
 - Serve:** Slice and enjoy your delicious homemade pizza!

AI-generated image:



Actual image:



Notes:

Too much ingredients for small pizza base. Onion was still raw, maybe baking before putting on pizza base. Tasty combination of toppings.

Pasta with Tomato Sauce

Question to AI (Our AI model prototype):

Can you give me an Italian pasta recipe with vegetables? I already have tomato sauce and filled pasta.

Ingredients:

- 200 g filled pasta (e.g., ravioli or tortellini)
- 15 ml olive oil
- 1 medium onion, finely chopped
- 2 cloves garlic, minced
- 1 medium zucchini, sliced into half-moons
- 1 medium carrot, sliced into rounds
- 150 g cherry tomatoes, halved
- 450 ml tomato sauce
- 15 g fresh basil, chopped (0.6 oz)
- Salt and pepper, to taste
- Freshly grated Parmesan cheese, for serving (optional)

Instructions:

Cook the filled pasta according to the package instructions in a large pot of salted boiling water until al dente. Reserve 1/2 cup of pasta water, then drain and set aside. In a large skillet or frying pan, heat the olive oil over medium heat. Add the chopped onion and cook for 3-4 minutes until softened. Add the minced garlic and cook for another minute until fragrant. Add the sliced zucchini and carrot to the pan, and cook for 5-7 minutes or until the vegetables are tender but still have a bit of crunch. Stir in the cherry tomatoes and cook for another 2-3 minutes until they begin to soften. Pour in the tomato sauce, and season with salt and pepper to taste. Simmer the sauce for 5-7 minutes until heated through. Add the cooked pasta and half of the chopped fresh basil to the pan, and toss everything together to combine. If the sauce is too thick, add some of the reserved pasta water to loosen it up. Serve the pasta and vegetables with freshly grated Parmesan cheese, if desired, and garnish with the remaining basil. Enjoy your Italian vegetable pasta!

AI-generated image:



Actual image:



Notes:

The first time it gave me measurements in cups and the basil was in milliliters. The carrots should have cooked a bit longer, but the zucchini was fine. I used ravioli filled with tomato and cheese, and the tomato sauce was with basil. It was a bit too much, but I changed the tomato sauce measurements.

Oyster Sauce Chicken & Rice

Question to AI (ChatGPT):

I have 30 minutes or so to cook, create a recipe which is tasty and not too difficult and has a high source of protein.

Ingredients:

- Oyster Sauce Chicken & Chickpea Stir-Fry
- Ingredients:
- 2 chicken thighs (or any cut of chicken you have) or tofu for a plant-based option
- 1 can chickpeas (drained and rinsed)
- 1 red bell pepper (sliced)
- 1 small onion (sliced)
- 2 cloves garlic (minced)
- 2-3 tablespoons oyster sauce
- 1 tablespoon soy sauce
- 1 tablespoon olive oil or sesame oil
- Salt and pepper to taste
- Fresh cilantro or parsley for garnish (optional)

Instructions:

- Prep the Chicken:** Cut chicken thighs (or tofu) into bite-sized pieces. Season with salt and pepper. **Cook Chicken:** Heat a tablespoon of oil in a large pan over medium-high heat. Add the chicken (or tofu) and cook for 5-6 minutes until browned and cooked through. Remove from the pan and set aside. **Stir-Fry Veggies:** In the same pan, add a bit more oil if needed. Sauté the onion, garlic, and red bell pepper for 3-4 minutes until softened. Add Chickpeas: Toss in the chickpeas and stir for another 2-3 minutes. **Combine & Sauce:** Return the cooked chicken (or tofu) to the pan. Add the oyster sauce and soy sauce. Stir everything together and cook for 2-3 more minutes to coat everything in the sauce and let the flavors combine. **Serve:** Garnish with fresh herbs if desired. Enjoy on its own or with a side of rice or noodles.

AI-generated image:



Actual image:



Notes:

The cooking time overall was longer than what AI recipe stated. Added more tablespoons of both soy sauce and oyster sauce than expected.

Pasta with Cream & Mushrooms

Question to AI (Our AI model prototype):

I have some pasta and mushrooms, make a recipe based on this.

Ingredients:

- 200g pasta (e.g., spaghetti or penne)
- 1 tablespoon olive oil
- 1 medium onion, chopped
- 2 cloves garlic, minced
- 150g mushrooms, sliced
- 1/2 cup cherry tomatoes, halved
- 1 tablespoon chopped fresh basil
- 2 tablespoons cream
- Salt and black pepper, to taste
- 50g grated Parmesan cheese

Instructions:

Cook the pasta according to package instructions until al dente. Drain and set aside. In a large pan, heat the olive oil over medium heat. Add the chopped onion and sauté for 3-4 minutes until softened. Add the minced garlic and sliced mushrooms to the pan and continue cooking for 5-7 minutes until the mushrooms have released their water and are lightly browned. Stir in the cherry tomatoes and cook for an additional 2-3 minutes until they begin to soften. Add the cooked pasta to the pan with the vegetables and mix well to combine. Season with salt and black pepper to taste. Stir in the chopped basil and 2 tablespoons of cream. Serve the pasta immediately, topped with grated Parmesan cheese. This dish is medium in protein, carbohydrates, and fat, and has a moderate budget. It's suitable for people with no dietary restrictions and can be easily adapted to be vegetarian or vegan by

AI-generated image:



Actual image:



Notes:

I asked to change the parsley for cream and the recipe put the cream in the order where the parsley would go. Not appropriate.

Pasta measurements were excessive. Took longer to sauté the onion with the garlic and mushrooms.

Cookbook

Lemon-Pepper Salmon with Sautéed Vegetables

Question to AI (Our AI model prototype):

I have salmon and some vegetables, what can I do with this?

Ingredients:

- 1 salmon fillet
- 1 tbsp olive oil
- 1 tsp lemon juice
- ½ tsp black pepper
- ¼ tsp salt
- Optional: A pinch of garlic powder
- ½ zucchini, sliced
- ½ cup green beans, trimmed
- ½ cup sliced carrots
- ½ red or green bell pepper, diced
- 1 small onion, chopped
- 1 tbsp olive oil
- ½ tsp Italian seasoning (or a mix of thyme, oregano, and basil)
- Salt and pepper to taste
- ½ avocado, sliced
- A pinch of salt and a drizzle of olive oil (optional)

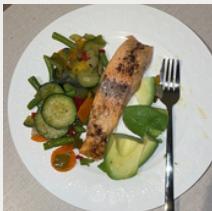
Instructions:

- Preheat your oven to 200°C (400°F). Place the salmon fillet on a baking tray lined with parchment paper. Drizzle olive oil and lemon juice over the salmon. Season with salt, pepper, and garlic powder if using. Bake for 12-15 minutes, or until the salmon is cooked through and flakes easily with a fork. While the salmon bakes, heat olive oil in a large skillet over medium heat. Add the onions and sauté until translucent, about 2 minutes. Add the carrots, green beans, zucchini, and bell pepper. Sprinkle with Italian seasoning, salt, and pepper. Cook for 5-7 minutes, stirring occasionally, until the vegetables are tender but still slightly crisp. Place the sautéed veggies and avocado on one side of the plate as well as the salmon. Serve the dish warm and enjoy a balanced, nutrient-packed meal!

AI-generated image:



Actual image:



Notes:

I personally like salmon little cooked, so I put it in the oven for around 8 minutes. I also put the oven at 180 degrees, 200 seemed too high.

I cooked the vegetables for longer than 7 minutes.

Eggplant and Bell Pepper Coconut Curry

Question to AI (ChatGPT):

I'm feeling curry today, can you create a non-spicy and easy-to-make curry recipe?

Ingredients:

- 1 medium eggplant, diced
- 1 red bell pepper, chopped
- 1 medium onion, finely chopped
- 2 cloves garlic, minced
- 1 tbsp ginger, grated
- 1 can (400ml) coconut milk
- 2 tbsp tomato paste
- 1 tbsp oyster sauce
- 1 tsp curry powder
- 1 tsp turmeric
- 1 tsp chili flakes (optional)
- 1 tbsp vegetable oil
- Salt and pepper to taste
- 1 cup basmati rice
- Fresh cilantro (optional, for garnish)
- Warm flatbread (optional, for scooping)

Instructions:

- Rinse the basmati rice thoroughly. Cook according to package instructions and set aside. Heat the vegetable oil in a large pan over medium heat. Add the diced eggplant and sauté for about 5 minutes until it softens slightly. Remove and set aside. In the same pan, sauté the onion, garlic, and ginger until fragrant (2-3 minutes). Add the chopped bell pepper and cook for another 3 minutes. Stir in the tomato paste, curry powder, turmeric, and chili flakes (if using). Cook for 1-2 minutes to toast the spices. Return the eggplant to the pan. Add the coconut milk and oyster sauce, stirring to combine. Simmer the curry over low heat for 10-12 minutes, allowing the flavors to meld. Season with salt and pepper to taste. Plate the cooked rice on one side of a shallow bowl or plate, adding the eggplant and bell pepper curry. Serve with flatbread for a heartier meal.

AI-generated image:



Actual image:



Notes:

Measurements between rice and chicken were not accurate

One-Pan Lemon Garlic Chicken with Veggies

Question to AI (ChatGPT):

Can you create a recipe high in protein with vegetables.

Ingredients:

- 1 boneless, skinless chicken breast
- 1 small zucchini, sliced
- 1 bell pepper (any color), sliced
- 1/2 cup cherry tomatoes
- 1 tablespoon olive oil
- 2 cloves garlic, minced
- 1/2 lemon (zest and juice)
- 1/2 teaspoon dried oregano
- Salt and pepper to taste
- Fresh parsley (optional, for garnish)

AI-generated image:



Actual image:



Instructions:

- Prep the Chicken:** Cut the chicken breast into bite-sized pieces. Season with a pinch of salt, pepper, and half the lemon zest. **Cook the Chicken:** Heat the olive oil in a large skillet over medium heat. Add the chicken and cook for about 5-7 minutes, stirring occasionally, until it's cooked through and lightly browned. **Add Veggies:** Toss in the zucchini, bell pepper, cherry tomatoes, and minced garlic. Cook for another 5 minutes until the veggies are tender but still crisp. **Add Flavor:** Squeeze the lemon juice over the chicken and veggies, sprinkle with oregano and the rest of the lemon zest. Stir well and let it cook for another **Serve:** Garnish with fresh parsley if you like, and enjoy your healthy, tasty meal straight from the pan!

Pasta with Creamy Zucchini Sauce and Salmon

Question to AI (ChatGPT):

Give me a gluten free pasta recipe. I currently have zucchini and salmon.

Ingredients:

- 100g lentil pasta
- one salmon fillet
- 1 medium zucchini
- 75g of peas
- 125g of cottage cheese
- 1 garlic clove
- 1/2 onion
- 1/2 tablespoon of olive oil
- 60ml of chicken broth
- salt and pepper to taste
- Fresh basil or parsley

AI-generated image:



Actual image:



Instructions:

- Cook the pasta according to package instructions. • Preheat air fryer to 180 degrees. • Season the salmon with salt and pepper. • Cook the salmon in the air fryer for 10-12 minutes until it is cooked through and slightly crispy on the edges. • Sauté the onion and garlic in olive oil until translucent. • Add the zucchini and peas. • Simmer the zucchini and peas in chicken broth until tender. • Blend the zucchini, peas, and broth until smooth. • Add the salmon back into the pan and pour the zucchini sauce over it. • Serve with a side of pasta.

Notes:

Too much onion reduced it to 1/4
the cottage cheese did not mix very easily
didn't find lentil pasta and therefore used spinach pasta

Cookbook

Mexican-Style Steak with Avocado and Potatoes

Question to AI (ChatGPT):

Can you give me a mexican recipe with steak?

Ingredients:

- 4 medium potatoes
- 1 tbsp paprika
- 2 tbsp olive oil
- 1 ripe avocado, diced
- 1 medium zucchini, sliced
- Beef tenderloin: seasoned with salt and pepper.
- 2 tomatoes (diced)
- 1 can of black beans (drained and rinsed)
- 1 small onion (chopped)
- 2 cloves garlic (minced)
- 1 tsp cumin
- 1 tsp chili powder
- salt and pepper to taste.

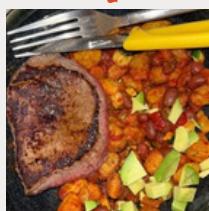
Instructions:

Prepare the Paprika Potatoes: Preheat the oven to 200°C (400°F). Toss the diced potatoes with olive oil, paprika, salt, and pepper. Spread them on a baking sheet and roast for 25-30 minutes until crispy. Cook the Kogelbeifstuk: Heat a skillet over medium-high heat. Add a bit of oil and sear the kogelbeifstuk for about 3-4 minutes on each side for medium-rare. Adjust the cooking time to your preferred doneness. Sauté the Zucchini: In the same skillet, add the sliced zucchini and sauté for about 5 minutes until tender. Make the Sauce: In a saucepan, heat a bit of oil over medium heat. Add the chopped onion and garlic, and sauté until fragrant. Add the diced tomatoes, black beans, cumin, chili powder, salt, and pepper. Simmer for about 10 minutes until the sauce thickens. Assemble the Dish: On a plate, arrange the roasted paprika potatoes, sautéed zucchini, and diced avocado. Place the kogelbeifstuk on top and spoon the tomato and black bean sauce over everything.

AI-generated image:



Actual image:



Notes:

- This was really good 3-4mins per side seemed too much so I only did a total of 5 mins to cook the steak I changed the order and cooked the vegetables before cooking the steak

Creamy Mushroom Chicken Zucchini Pasta

Question to AI (ChatGPT):

Meal for 2 people with mushrooms, chicken and pasta.

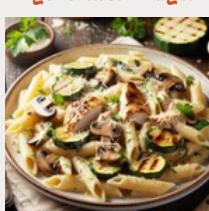
Ingredients:

- 200g of pasta
- 2 diced chicken breasts
- 1 zucchini
- 200g of mushrooms
- 1 cup heavy cream
- 1/2 cup of parmesan cheese
- 2 cloves of garlic
- 1 tablespoon of olive oil
- salt and pepper
- fresh coriander

Instructions:

Cook the pasta according to package, drain and set aside. Prepare chicken and vegetables: heat the olive oil over medium-high heat. Add the diced chicken and cook until browned and cooked through. Remove the chicken from the skillet and set aside. Sauté the vegetables: in the same skillet, add the garlic, for about one minute until fragrant. Add the sliced zucchini and mushrooms and cook until they are tender. Make the sauce: Reduce the heat to medium, add the heavy cream until the sauce is smooth and creamy. Season with salt and pepper. Combine everything: add the cooked pasta to the skillet, tossing to coat everything evenly with the sauce. Cook for an additional 2-3 minutes until everything is heated through.

AI-generated image:



Actual image:



Notes:

- I personally like it when it is only a bit creamy so I would reduce the amount of heavy cream.
- Not enough mushrooms

Mexican Rice Chicken Bowl

Question to AI (ChatGPT):

Make a rice bowl with chicken, vegetables and cottage cheese.

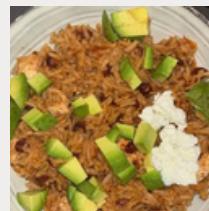
Ingredients:

- 2 chicken breasts
- 1 tsp cumin
- 1 tsp chili powder
- Salt and pepper to taste
- 2 tbsp olive oil (divided)
- 1 cup cooked rice
- 1 can black beans (drained and rinsed)
- 1 ripe avocado (diced)
- 1/2 cup cottage cheese
- Salsa
- Chopped cilantro
- Lime wedges

AI-generated image:



Actual image:



Instructions:

- **Cook the Chicken:** Season the chicken breasts with cumin, chili powder, salt, and pepper. Heat 1 tbsp of olive oil in a skillet over medium-high heat. Cook the chicken breasts for about 6-7 minutes on each side until fully cooked. Let them rest before slicing.
- **Prepare the Rice:** Cook the rice according to package instructions.
- **Heat the Beans:** In a small pot, heat the black beans over medium heat until warmed through.
- **Assemble the Bowl:** In each bowl, layer the cooked rice, black beans, diced avocado, and sliced chicken. Add a dollop of cottage cheese on top. Garnish with salsa, chopped cilantro, and a squeeze of lime juice.

Notes:

- Very easy to make
- A bit too much rice so it was a little bit dry either add more wet ingredients or reduce the amount of rice

Minced Beef Fried Rice Bowl

Question to AI (ChatGPT):

Give me an Indonesian style protein rich meal.

Ingredients:

- 300g minced beef or pork
- 1 small onion, finely chopped
- 2 cloves garlic, minced
- Salt and pepper to taste
- 1 tsp paprika
- 1 tbsp olive oil
- 1 cup cooked rice
- 1 medium zucchini, sliced
- 2 tomatoes, diced
- 2 eggs
- 1/2 cup tomato sauce
- 1 tbsp soy sauce
- 1 tsp chili sauce (optional)
- 1 tsp honey

AI-generated image:



Actual image:



Instructions:

- **Cook the Minced Meat:** Heat 1 tbsp of olive oil in a skillet over medium heat. Add the chopped onion and garlic, and sauté until fragrant. Add the minced meat, paprika, salt, and pepper. Cook until the meat is browned and cooked through.
- **Prepare the Rice:** Cook the rice according to package instructions.
- **Sauté the Vegetables:** In the same skillet, add the sliced zucchini and diced tomatoes. Sauté for about 5 minutes until tender.
- **Cook the Eggs:** In a separate pan, cook the eggs to your liking (fried, scrambled, or poached).
- **Make the Sauce:**

Notes:

- Really liked it, however you can't really taste the honey so maybe add another teaspoon.

Cookbook

Schnitzel and Pearl Couscous

Question to AI (ChatGPT):

Create a recipe using pearl couscous, paprika, zucchini, cut-up schnitzel, and parsley.

Ingredients:

- Pearl Couscous: 1 cup pearl couscous
- Paprika: 1 medium red bell pepper, diced
- Zucchini: 1 medium zucchini, sliced
- Schnitzel: 2 pieces of schnitzel, cooked and cut into strips
- Parsley: 1/4 cup fresh parsley, chopped
- Olive Oil: 2 tbsp
- Garlic: 2 cloves, minced
- Salt and Pepper: to taste
- Lemon Juice: 1 tbsp

Instructions:

Cook the Pearl Couscous: Prepare the pearl couscous according to the package instructions. Once cooked, set aside. **Sauté the Vegetables:** In a large skillet, heat 1 tbsp of olive oil over medium heat. Add the minced garlic and sauté until fragrant. Add the diced paprika and sliced zucchini, and cook until tender, about 5-7 minutes. **Combine Ingredients:** Add the cooked pearl couscous to the skillet with the vegetables. Stir to combine. **Add the Schnitzel:** Add the cut-up schnitzel strips to the skillet and mix well. **Season and Garnish:** Season with salt and pepper to taste. Drizzle with the remaining 1 tbsp of olive oil and lemon juice. Garnish with chopped parsley.

AI-generated image:



Actual image:



Notes:

I overcooked the schnitzel. I wasn't given instructions on it so I just guessed. Other than that it was very good.

Chicken and Lentil Curry Rice

Question to AI (ChatGPT):

Create a recipe for chicken and lentil curry rice.

Ingredients:

- 2 chicken breasts, diced
- 1 tbsp olive oil
- Salt and pepper to taste
- 1 cup red lentils
- 1 onion, finely chopped
- 2 cloves garlic, minced
- 1 tbsp ginger, minced
- 1 can coconut milk
- 1 can diced tomatoes
- 2 tbsp curry powder
- 1 tsp turmeric
- 1 tsp cumin
- 1 tsp coriander
- 2 cups vegetable or chicken broth
- 1 cup basmati rice
- 2 cups water
- 1 tbsp butter or oil
- Fresh cilantro, chopped
- Lime wedges

AI-generated image:



Actual image:



Instructions:

Cook the Rice: In a pot, bring 2 cups of water to a boil. Add the basmati rice, butter or oil, and a pinch of salt. Reduce heat to low, cover, and simmer for 15-20 minutes until the rice is cooked. **Prepare the Chicken:** In a skillet, heat 1 tbsp of olive oil over medium-high heat. Season the diced chicken with salt and pepper. Cook the chicken until browned and cooked through. Set aside. **Make the Lentil Curry:** In a large pot, heat a bit of oil over medium heat. Add the chopped onion, garlic, and ginger, and sauté until fragrant. Add the curry powder, turmeric, cumin, coriander, and cook for another minute. Stir in the lentils, coconut milk, diced tomatoes, and broth. Bring to a boil, then reduce heat and simmer for 20-25 minutes until the lentils are tender. **Combine and Serve:** Add the cooked chicken to the lentil curry and stir to combine. Serve the curry over the cooked rice. Garnish with fresh cilantro and lime wedges.

Notes:

- Lots of ingredients were needed
- I don't really like ginger in food so I didn't add it. Forgot to buy tomatoes so also didn't use those

Tom Kha Gai Soup

Question to AI (ChatGPT):

Give me a recipe of a traditional Thai soup.

Ingredients:

- 2 cups coconut milk 2 cups chicken broth 1 lb (450g) chicken breast, thinly sliced 3-4 kaffir lime leaves, torn into pieces 2-3 stalks lemongrass, cut into 2-inch pieces and smashed 3-4 slices galangal (or ginger if unavailable) 2-3 Thai birds eye chilies, smashed (adjust to taste) 200g mushrooms, sliced 1 medium onion, sliced 1-2 tomatoes, cut into wedges 2-3 tbsp fish sauce (adjust to taste) 1-2 tbsp lime juice (adjust to taste) 1 tsp sugar Fresh cilantro, chopped (for garnish)
- ...
- ...
- ...
- ...

Instructions:

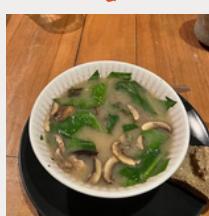
Prepare the Broth: In a large pot, combine the coconut milk and chicken broth. Bring to a gentle boil over medium heat. Add the kaffir lime leaves, lemongrass, and galangal. Let it simmer for about 5 minutes to infuse the flavors. **Cook the Chicken:** Add the sliced chicken breast to the pot. Cook until the chicken is just done, about 5-7 minutes. **Add Vegetables:** Add the mushrooms, onion, and tomatoes. Continue to simmer until the vegetables are tender, about 5 minutes. **Season the Soup:** Add the fish sauce, lime juice, and sugar. Adjust the seasoning to taste. If you prefer it spicier, add more chilies. **Serve:** Ladle the soup into bowls. Garnish with fresh cilantro.

- ...
- ...
- ...
- ...

AI-generated image:



Actual image:



Notes:

- It was very flavorful.
- Mushrooms should have a cooked a bit longer. Didn't add the chicken, because i was eating with a vegetarian friend.

Soy-Garlic Noodles with Crispy Seaweed

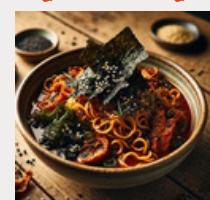
Question to AI (ChatGPT):

I have noodles and seaweed can you create a recipe?

Ingredients:

- 200g wok noodles
- 200g chicken thighs (or tofu for a vegetarian version)
- 2 tbsp oyster sauce
- 1 tbsp soy sauce
- 1 tbsp honey or brown sugar
- 1 tsp sesame oil
- 2 garlic cloves, minced
- 1 tbsp vegetable oil
- 1 tsp cornstarch (optional, for thickening)
- Toppings:
 - Sheets of roasted seaweed (cut or torn into small pieces)
 - Sesame seeds (optional)

AI-generated image:



Instructions:

If using chicken, cut it into bite-sized pieces. Marinate with 1 tsp soy sauce, 1 tsp oyster sauce, and a pinch of cornstarch for 10 minutes. Cook the noodles: Boil the noodles according to package instructions. Drain and set aside. Cook the chicken (or tofu): Heat 1 tbsp vegetable oil in a skillet over medium heat. Add the marinated chicken (or tofu) and cook until golden and fully cooked, about 5-7 minutes. Remove from the pan and set aside. Make the sauce: In the same pan, add garlic and sauté until fragrant. Mix in 1 tbsp oyster sauce, 1 tbsp soy sauce, honey, and sesame oil. Stir and let simmer for 1 minute. Add a splash of water if the sauce is too thick.

Combine noodles and sauce: Toss the cooked noodles in the sauce until evenly coated. Add the chicken back to the pan and mix well. Serve and garnish:

- Transfer the noodles to a bowl. Top with torn seaweed and sesame seeds for added crunch and flavor.
- Enjoy your Soy-Garlic Noodles with Crispy Seaweed!

Actual image:



Notes:

Gave a vegetarian version even though I didn't ask for it.

Cookbook

Apple Cinnamon Rice Pudding with Mango

Question to AI (Microsoft Copilot):

Give me a sweet healthy breakfast recipe.

Ingredients:

- 1 cup rice (preferably Arborio or short-grain rice)
- 2 cups water 2 cups milk 1 cup coconut milk 1/2 cup sugar (adjust to taste) 1 tsp vanilla extract 1 tsp ground cinnamon 1/2 tsp ground nutmeg 2 apples, peeled, cored, and diced 1 ripe mango, peeled and diced 1/4 cup raisins (optional) A pinch of salt
- .
- .
- .
- .
- .
- .

Instructions:

- Cook the Rice:** In a medium saucepan, bring the water to a boil. Add the rice, reduce the heat to low, cover, and simmer for about 15 minutes or until the water is absorbed. **Prepare the Pudding:** Add the milk, coconut milk, sugar, vanilla extract, cinnamon, nutmeg, and a pinch of salt to the cooked rice. Stir well to combine. Cook over medium heat, stirring frequently, until the mixture thickens and the rice is tender, about 20-25 minutes. **Add the Apples:** Stir in the diced apples and raisins (if using). Continue to cook for another 5-10 minutes until the apples are tender. **Serve:** Remove the rice pudding from the heat and let it cool slightly. Serve warm or chilled, topped with the diced mango.
- .
- .

AI-generated image:



Actual image:



Notes:

- It was super good
- I don't like raisins, so I didn't put them in.

The perfect cookbook
for the perfect future
chef!