COOKIT

By Guhan Sundaram, Sunil Swarna, Venkata Khande

CookIt - An Introduction

- CookIt is a Web Application focused on providing customized recipes to users based on whatever ingredients they may have or wish to have
- The application features filters and premade ingredient lists, along with the ability to store their preferences and purchased ingredients in the local cache or within their account
- The recipes shown can be detailed and can be tailored to the user’s requirements
- Step-by-step instructions make food prep simple. Each step shows the ingredients needed with a tap, while a shopping list lets users track the grocery items they need.

Targeted E - Efficiency

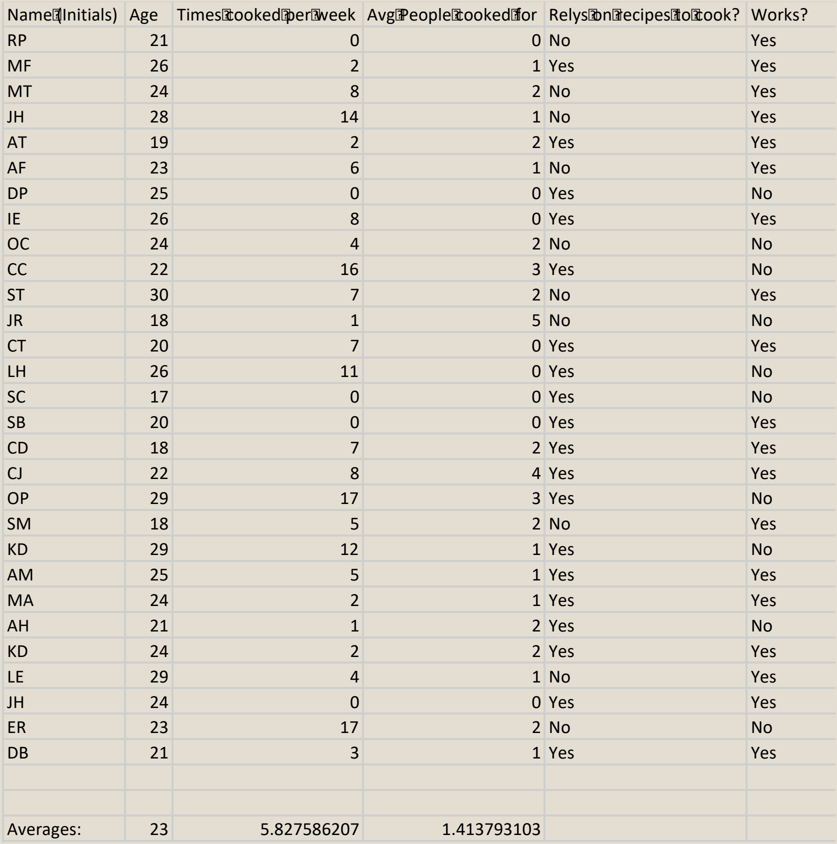
- Our main focus is on **efficiency**.
- This aspect of design is what we'll be focusing on heavily because at the end of the day, we want to provide a quick and easy way for users to look at applicable recipes.
- After analysis of the other design 'E' components, we observed as a group that when you think logically, food recipes are not safety-critical instructions like rocket fuel compositions. This means that focusing on error-tolerance is not required. The simplicity and banality of our app also means that we don't need to focus on effectiveness specifically, as this feature would be inherent to an efficient design of our app.
- A user needs to be able to input their ingredients quickly, filter out unwanted styles of recipes (unhealthy, etc), and view all possible recipes quickly, so that they can focus on cooking, with a simply laid out recipe in front of them.

Measurement of improvement

- Our main metric to measure efficiency will be operational speed
- The way we will do this will be to time how long the user takes to finish several tasks.
- Because we can save contained ingredients in the user's specific account, we need a user account system. This means a login and sign up system, which needs to be similar to existing systems, so as not to confuse users.
- We also time how long a naïve user takes to carry out the core functionality of the app. By this we mean a user who has never used the app before. We would thus time the efficacy of the design at providing information quickly and efficiently to the user.
- We can do this old-school, observing users with a stopwatch. We are thinking not to use timing functions within the app because we want to see the entire process time from the perspective of the user, including button clicks.

Preliminary Data - Survey

- We used a survey firstly to analyze our audience, distributed around the Steven's community. This would include demographic information and details regarding food and cooking choices.
- We also used food related data analytics to look at different recipes and the kind of people that would eat them, such as the data on which supermarket products are most popular and who they are popular with.
- We drew from a questionnaire followed by a short interview of a few participants who were from our target audience (ages 18-25).
- An example of the persona trait derived from the data was the average times a week spent cooking and the statistic on how many people that the person cooks for.



Name (Initials)	Age	Times cooked per week	Avg People cooked for	Relys on recipes to cook?	Works?
RP	21	0	0	No	Yes
MF	26	2	1	Yes	Yes
MT	24	8	2	No	Yes
JH	28	14	1	No	Yes
AT	19	2	2	Yes	Yes
AF	23	6	1	No	Yes
DP	25	0	0	Yes	No
IE	26	8	0	Yes	Yes
OC	24	4	2	No	No
CC	22	16	3	Yes	No
ST	30	7	2	No	Yes
JR	18	1	5	No	No
CT	20	7	0	Yes	Yes
LH	26	11	0	Yes	No
SC	17	0	0	Yes	No
SB	20	0	0	Yes	Yes
CD	18	7	2	Yes	Yes
CJ	22	8	4	Yes	Yes
OP	29	17	3	Yes	No
SM	18	5	2	No	Yes
KD	29	12	1	Yes	No
AM	25	5	1	Yes	Yes
MA	24	2	1	Yes	Yes
AH	21	1	2	Yes	No
KD	24	2	2	Yes	Yes
LE	29	4	1	No	Yes
JH	24	0	0	Yes	Yes
ER	23	17	2	No	No
DB	21	3	1	Yes	Yes
Averages:	23	5.827586207	1.413793103		

Target Population

- Our target population are the people who find it difficult to cook their food on a day to day basis. We provide recipes from the items they have in their kitchen.
- This population is young working people from 18 to 35, who would be inclined to use tech to solve their food related problems.
- This population, known as millennials, are nearly a fourth of the United States population and are overrepresented in using web applications and other forms of modern technology.
- Over 90% of these millennials own smartphones, providing us with a huge market for our application.
- Our core assumption is that this demographic will be most likely to respond warmly to our CookIt app, despite our app not being age-specific

Preliminary Data - Survey

- Qualitative Results Found:
  - Younger adults (18-21) were more likely to be health-conscious, especially young women. Men from ages 18-35 were additionally likely to value protein-containing foods, which was commonly said to be for gym training.
  - Many young adults said that they preferred home-cooked food to takeout or restaurant cuisine, but they said that it was fairly difficult to find time to cook, forcing pricier and unhealthier options
  - Younger people were also more in favor of using food and recipe apps compared to cookbooks and other forms of traditional literature
  - Older people needed cooking recipes and guides less, mostly due to higher levels of cooking experience. This vindicates our focus on younger people, even if they don't cook as much.

Preliminary Data - Industry

- Television channels related to cooking have been doing very well in terms of ratings. In 2018, the Food Network saw an increase of 4%, of which both women and millennials were overrepresented. This represents for us proof that millennials are interested in the minutiae of cooking and are willing to peruse media related to recipes and ingredients.
- A report by the US department of Agriculture indicated that millennials eat food outside the home 30% more than other age demographics. This is consistent with our own findings, and this issue is what our app is meant to help solve.
- In another study, only 64% of millennials said they knew how to cook well, struggling to succeed at basic tasks like making salad. Again, this is where our easy to use app that explains cooking instructions help.

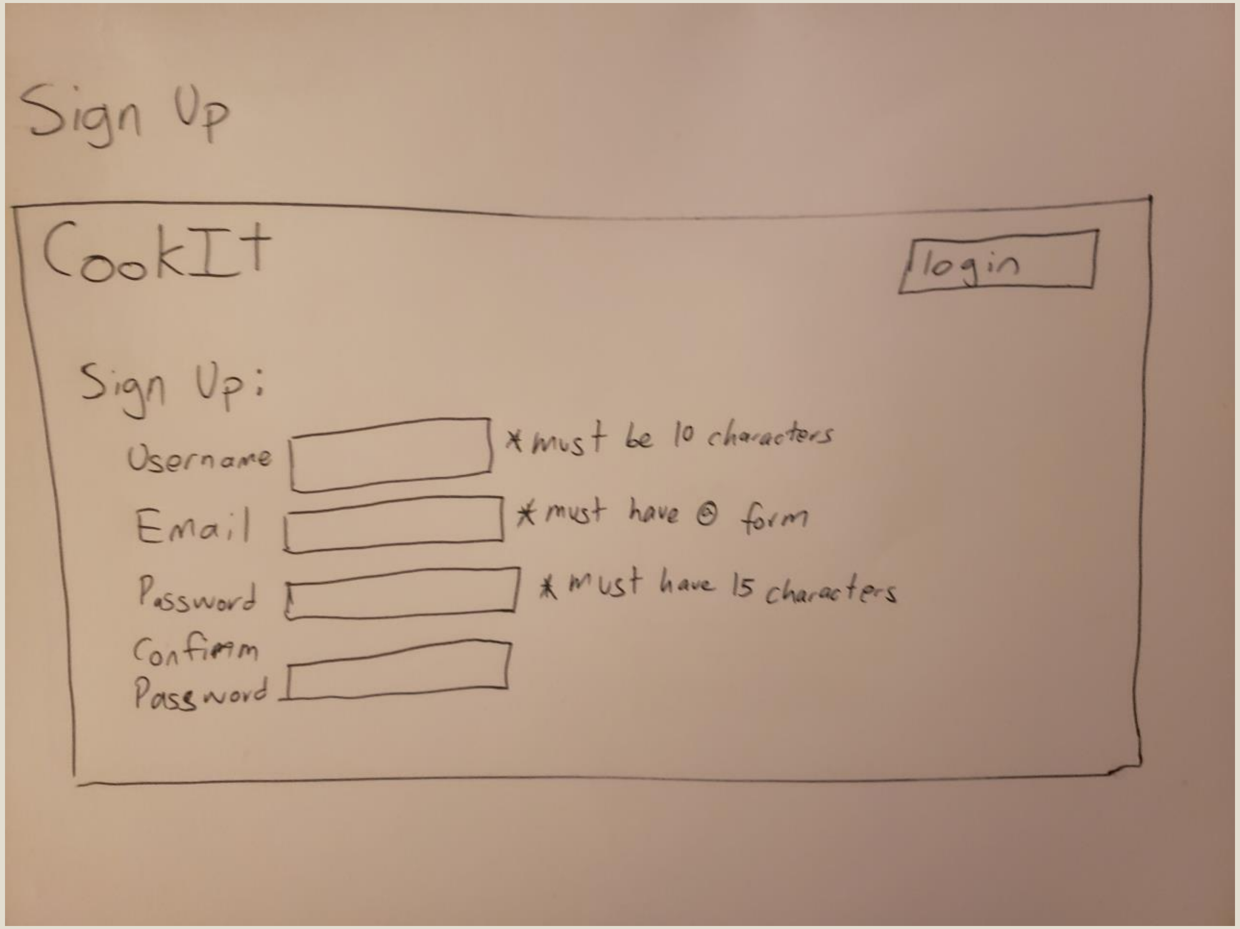
Persona - John the Student

John is a 23-year old graduate student at CRT University studying Electrical Engineering. As such, he’s fairly tech-savvy. He has a part time job at his University’s library and he also works as a teaching assistant for undergrads. He regularly goes to the gym twice a week and tries to live healthily.

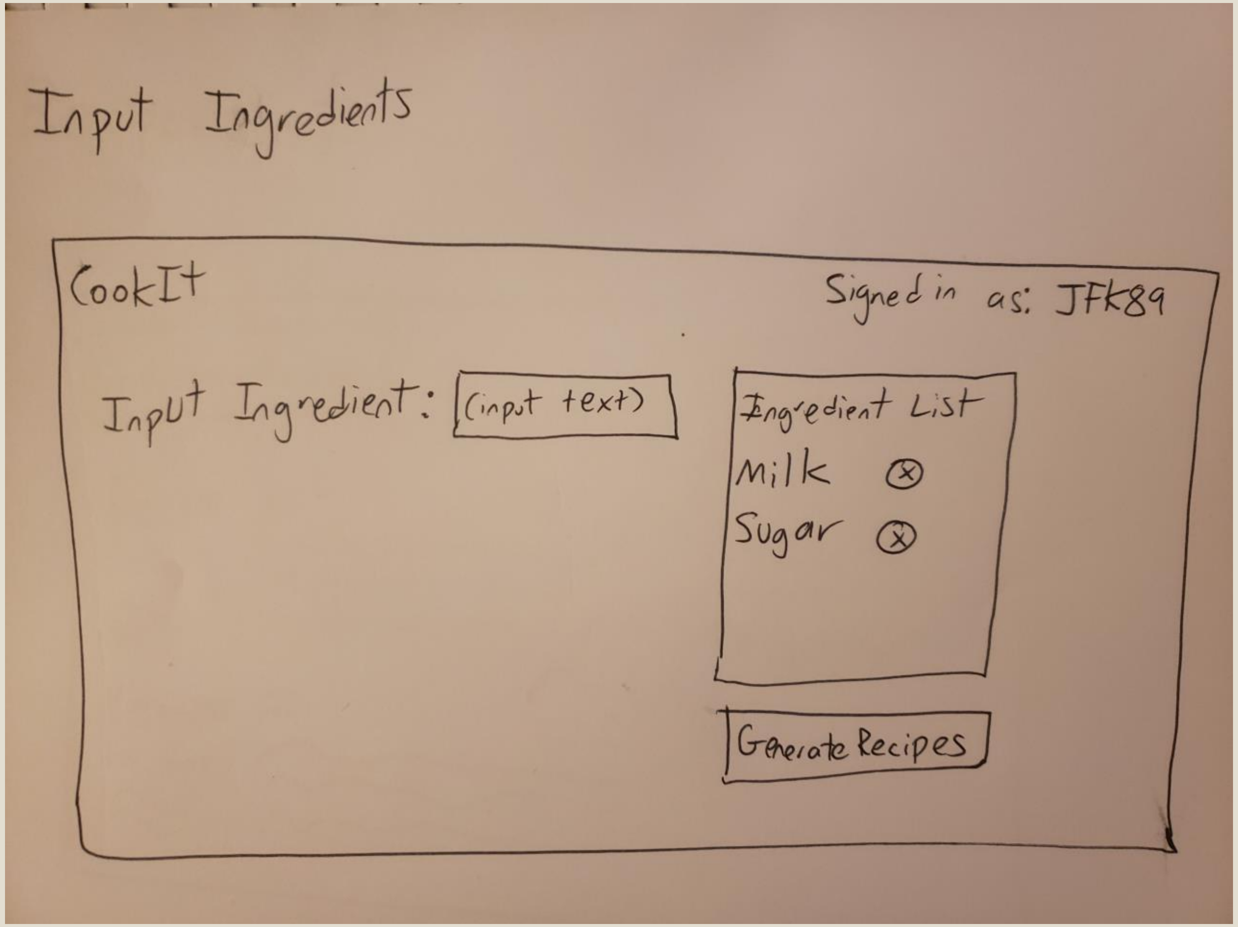
His schedule is thus very busy and he fails to find time to cook often. He estimates that he cooks 6 times a week on average only for himself, eating the rest of his meals outside or delivered. His desire is to learn the ability to cook healthy meals quick given his current ingredients at any time.

PS: This persona was generated from our lived experiences, the survey we did, and analysis from industry of millennials, some of which is seen in the previous slide.

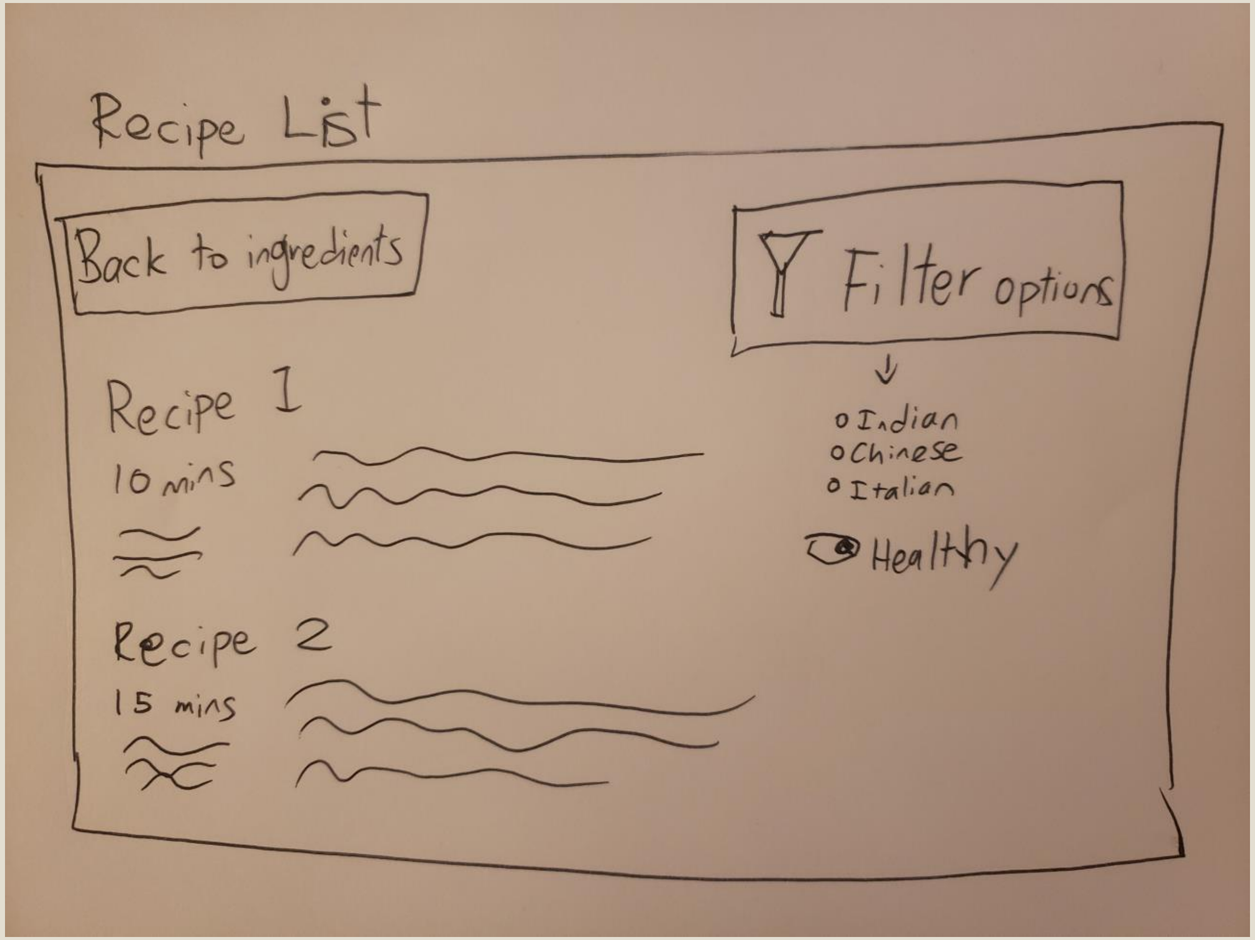
1st Iteration – Paper Prototype



1st Iteration – Paper Prototype



1st Iteration – Paper Prototype

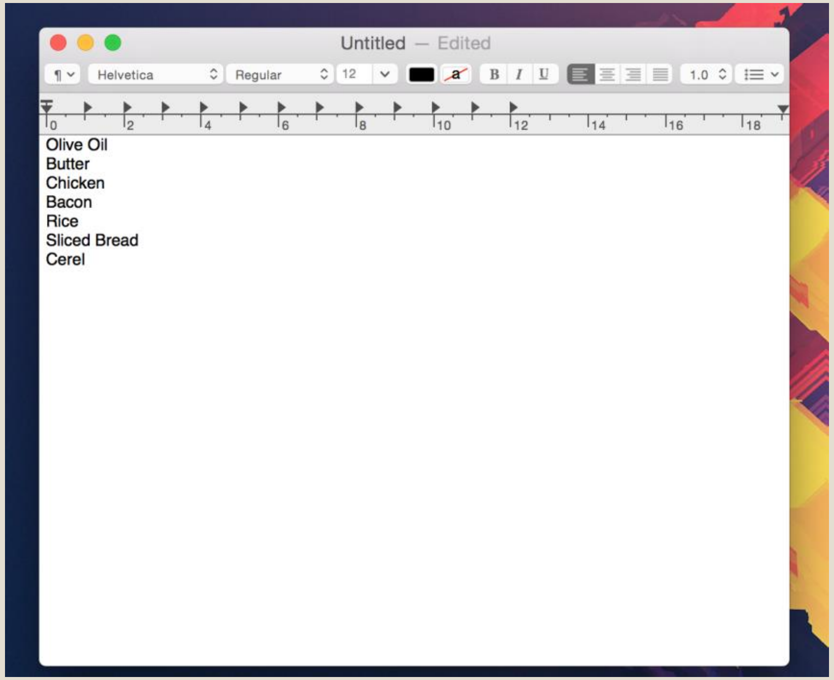


Focus Group : Response 1

- Our predetermined focus group, taken from a random sample of the respondents to our original survey, provided us with feedback on our paper-based prototype of CookIt
  - Users did not like only having to input their ingredients manually using text. They mentioned that some form of drop down menu system would be preferable
  - Users wanted a proper filtration system before viewing the recipes.
  - Users liked that several of the buttons were green
  - They were accepting of the login system, as it was fairly similar to existing websites.
  - Since it was a paper prototype, we couldn't get accurate operational speed readings to measure efficiency . However, people told us that they would likely spend most time writing in their ingredients, hence the request for a drop down menu.

Focus Group: Response 1 cont.

- To implement the paper prototypes, we asked users to type on a keyboard the list of ingredients to mimic the act of typing them into an input bar in the final app. Then, we would show them a mockup of a recipe list that they would likely see after they input the ingredients in.
- Firstly, we found that not only did this take several minutes, the users struggled to even remember their ingredients.
- Furthermore, there were even spelling mistakes made, which would likely mess with the text matching system we had though of. This would require change.



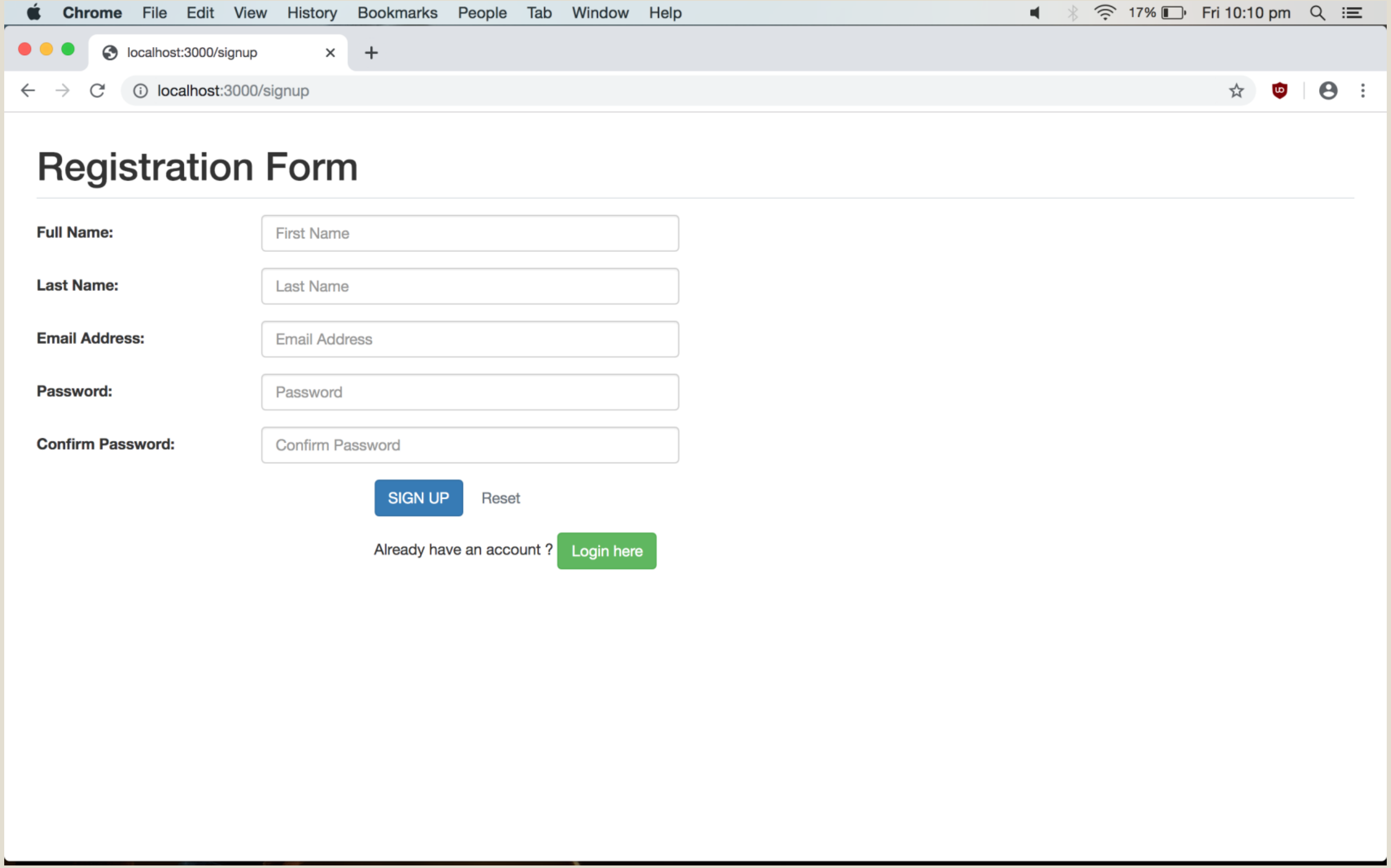
Olive Oil
Butter
Chicken
Bacon
Rice
Sliced Bread
Cerel

2nd Iteration – Medium Fidelity

- We produced our second iteration in Balsamiq. This system had a dropdown menu which also had a search function embedded within it.
- It also included a basic filtration input that could be accessed after you clicked on the "Find Recipes" button
- The cache that saved user ingredient inputs was not implemented at this stage.
- The most important thing was that the ingredient input method was completely changed. It switched from a raw input bar to a dropdown menu with all the ingredients in a list by alphabetical order, with an additional dropdown menu before that with the overarching categories, like meat and dairy, in order to decrease the time taken to pick a specific ingredient.
- We also included a filter icon and pushed it to the input ingredients page right below the find recipes button, so that users associate the 2.

Focus Group: I2 Response

- Our focus group interacted with the second prototype and provided us with some feedback after utilizing the app
  - One issue was the app was only semi-functional. There was only 1 combination of ingredients that produced a recipe, so the users felt that their feedback would be limited.
  - Another issue is that there weren't enough filtering options. Also, the filtering element was in the page after the recipes were generated whereas people wanted to filter out recipes in the ingredients page itself.
  - 3 members of the focus group felt that one piece of functionality had been lost in the second prototype. This was the ability to type in ingredients and add them, instead of scrolling through what could be an arbitrarily large dropdown menu. This had to be fixed for our final release.
  - Another important problem was that users were not able to remove items from the ingredients list, which they felt was important.


Chrome File Edit View History Bookmarks People Tab Window Help
17% Fri 10:10 pm
localhost:3000/signup
localhost:3000/signup

Registration Form

Full Name:	First Name
Last Name:	Last Name
Email Address:	Email Address
Password:	Password
Confirm Password:	Confirm Password

SIGN UP Reset

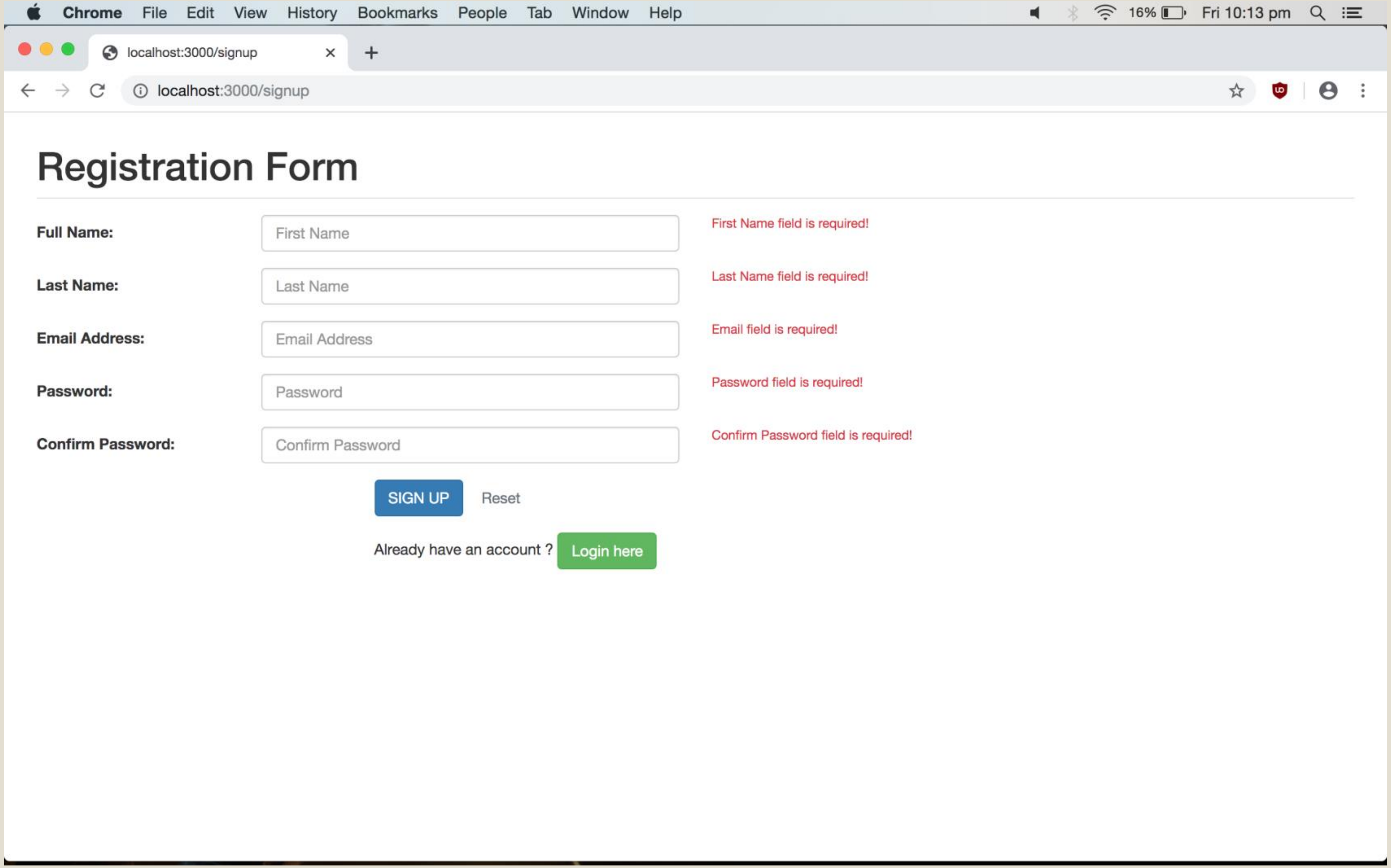
Already have an account ? Login here

Final Iteration – Sign Up Page

Here, we see the basic sign up form. It is simple and leaves no ambiguity.

There is a clear sign up button, with an option to log in instead for existing users below, with clear marking.

There is also finally a reset button just to make filling in the form easier.


Chrome File Edit View History Bookmarks People Tab Window Help
16% Fri 10:13 pm
localhost:3000/signup
localhost:3000/signup

Registration Form

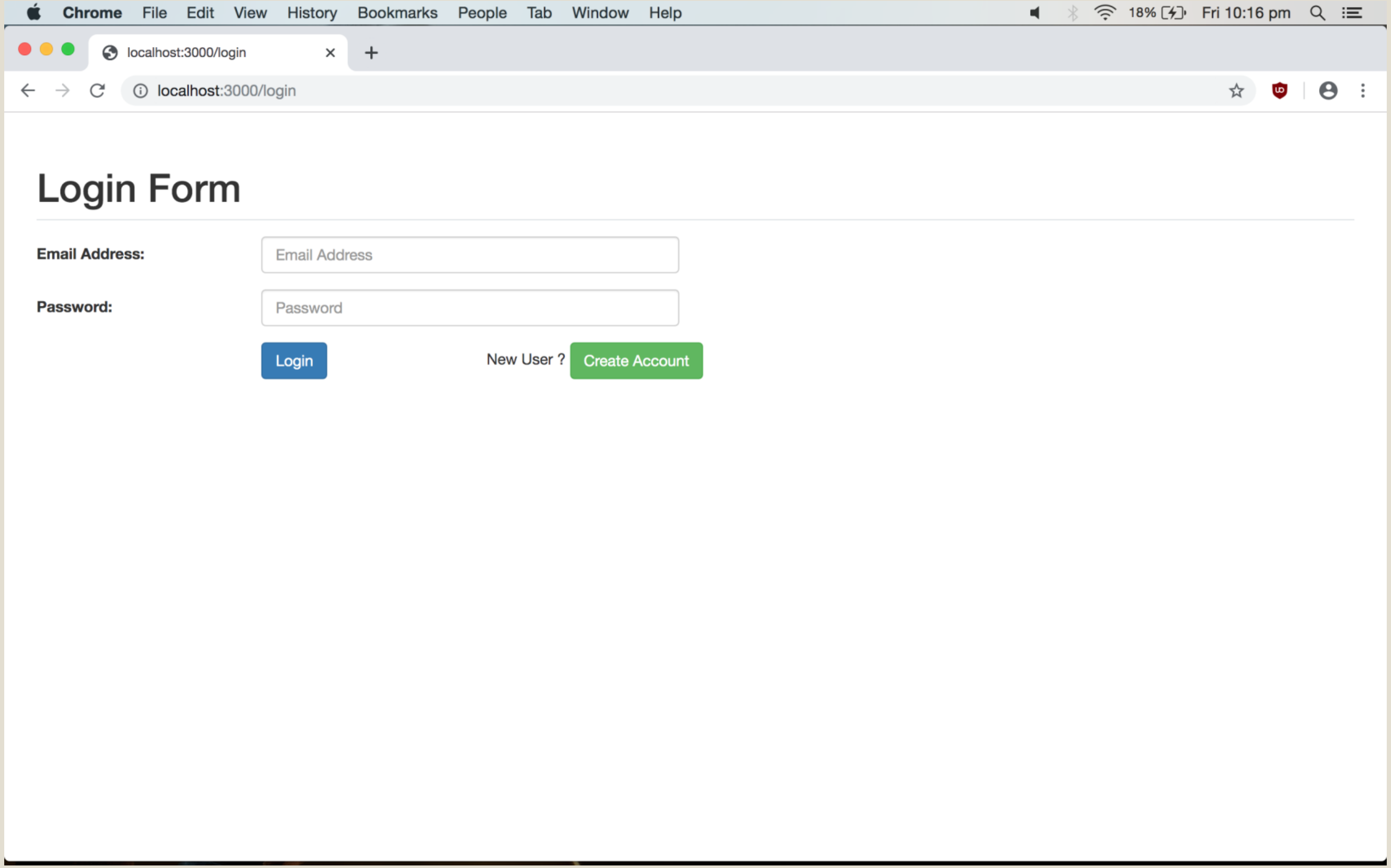
Full Name:	First Name	First Name field is required!
Last Name:	Last Name	Last Name field is required!
Email Address:	Email Address	Email field is required!
Password:	Password	Password field is required!
Confirm Password:	Confirm Password	Confirm Password field is required!

Already have an account ?

Final Iteration – Sign Up Page 2

Now, we have a way of rectifying the sign up process, for example, if a user has missed out specific items.

This is very typical to sign up forms; we didn't feel the need to reinvent the wheel.



Chrome File Edit View History Bookmarks People Tab Window Help
18% Fri 10:16 pm
localhost:3000/login
localhost:3000/login

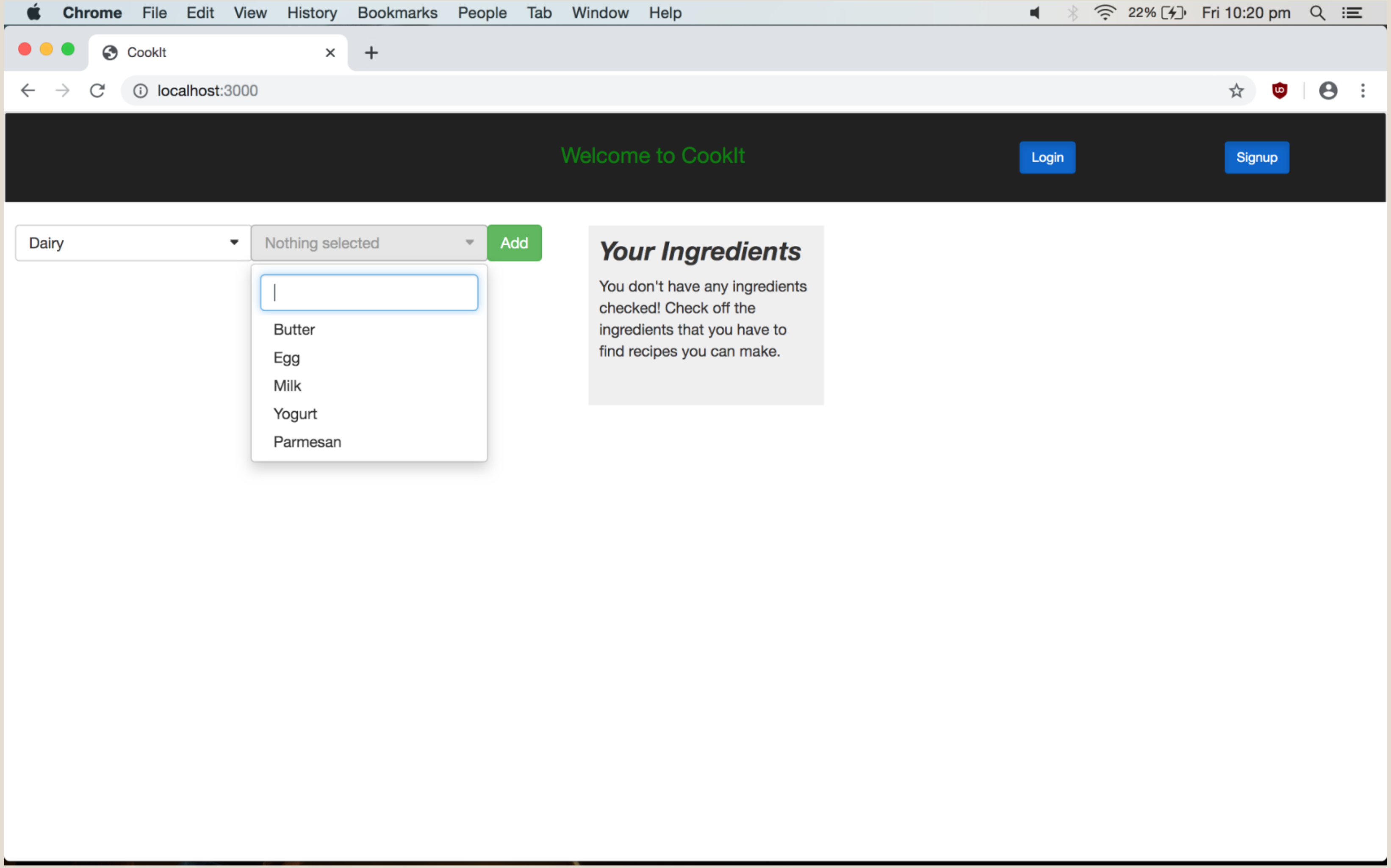
Login Form

Email Address: Email Address

Password: Password

Final Iteration – Login Form

A fairly self-explanatory login form



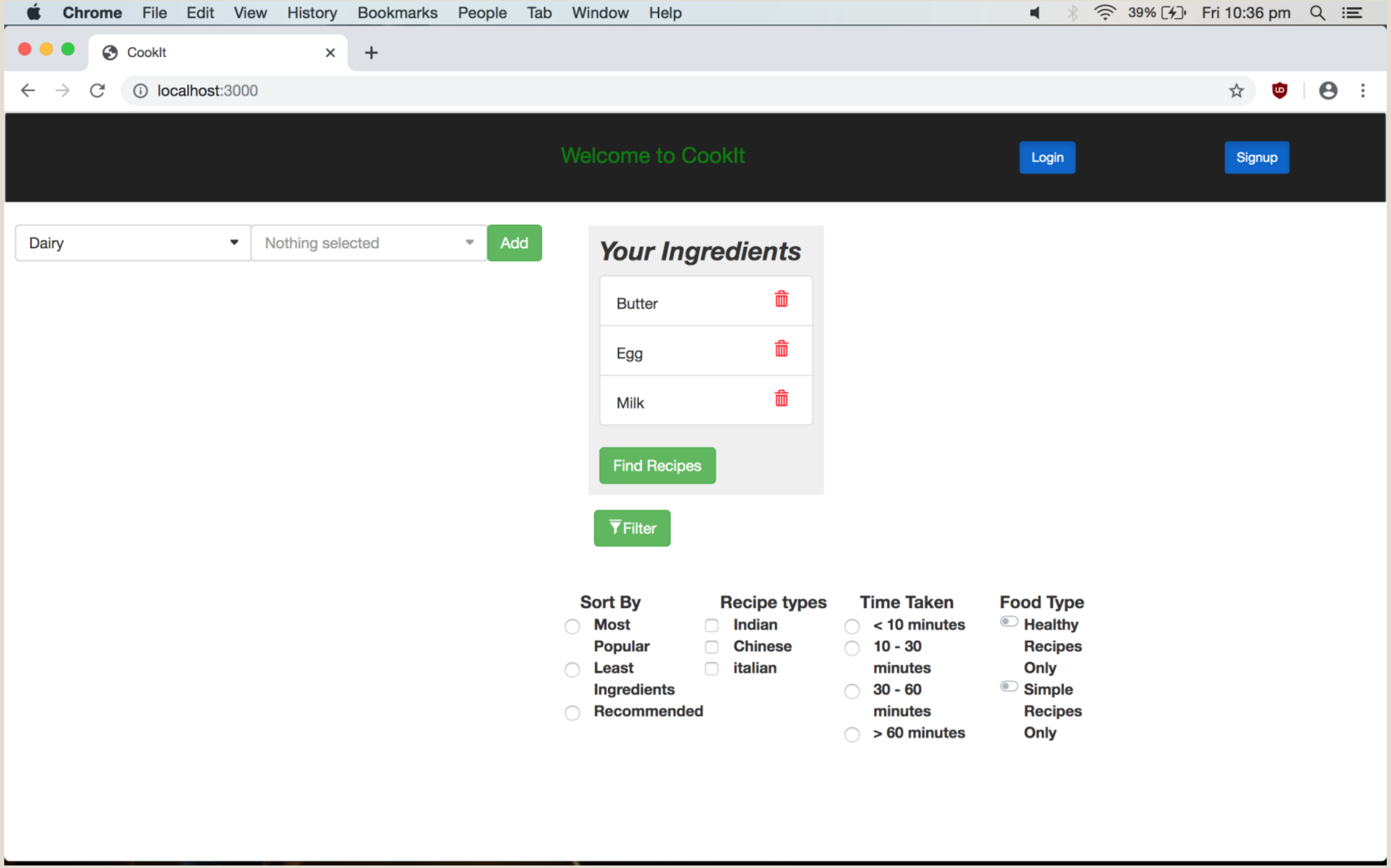
Your Ingredients

You don't have any ingredients checked! Check off the ingredients that you have to find recipes you can make.

Final Iteration – Ingredients Input

We can observe the dual dropdown menu. The second one is dynamic and changes with the first menu choice.

We've also implemented a search function to add ingredients easily and quickly



Dairy

Nothing selected

Your Ingredients

- Butter
- Egg
- Milk

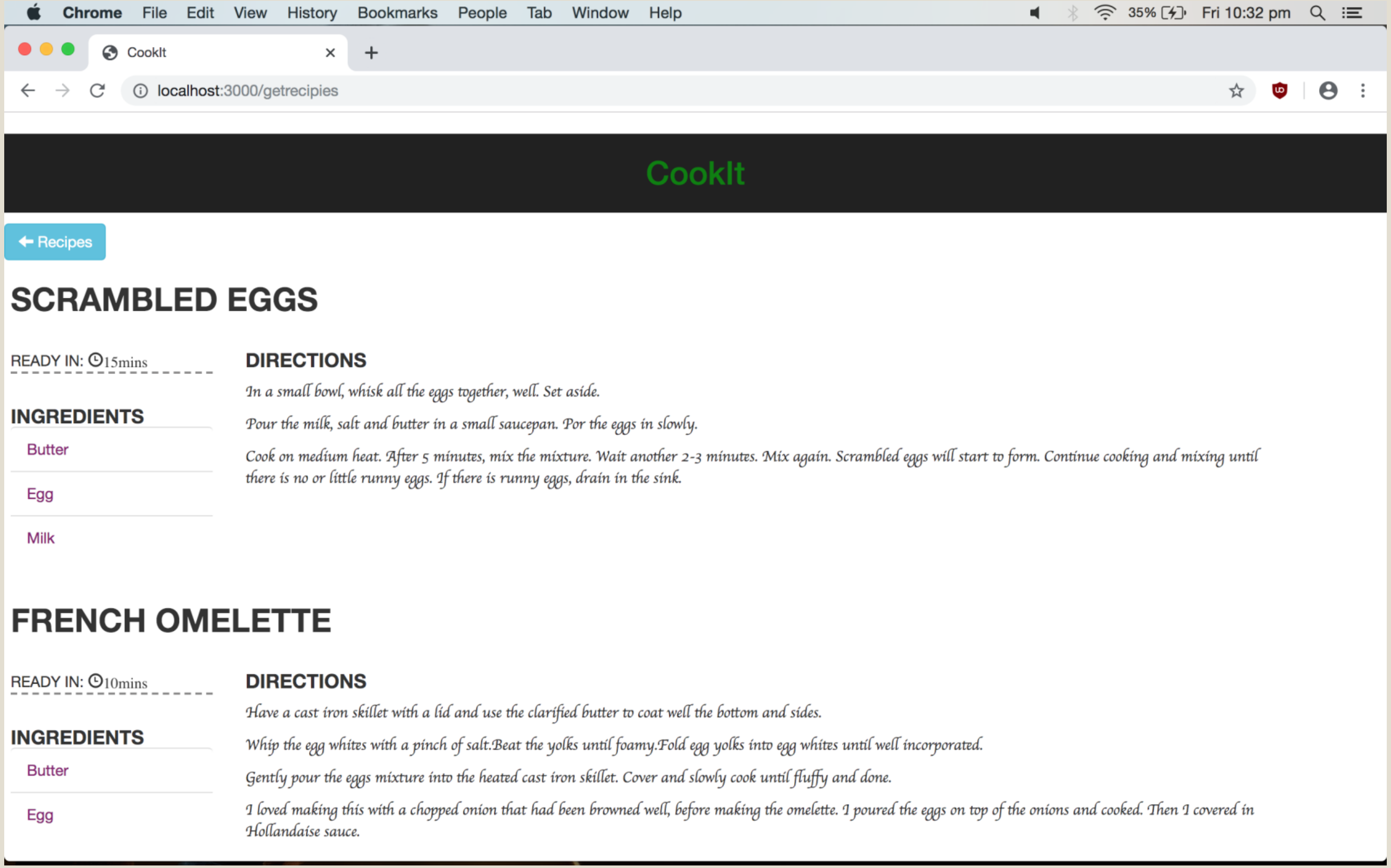
Sort By	Recipe types	Time Taken	Food Type
○ Most Popular	□ Indian	○ < 10 minutes	Healthy Recipes Only
○ Least Ingredients	□ Chinese	○ 10 - 30 minutes	Simple Recipes Only
○ Recommended	□ italian	○ 30 - 60 minutes	
		○ > 60 minutes	

Final Iteration – Ingredients Input 2

Here we can see that there is an easy to see option to delete each item.

We can also see that there is a filter symbol to help understanding with clear selection options with proper types. This part is only seen when clicking on "Filter", to simplify the app.

Final Iteration – Recipe List



SCRAMBLED EGGS

READY IN: 15mins

INGREDIENTS

- Butter
- Egg
- Milk

DIRECTIONS

In a small bowl, whisk all the eggs together, well. Set aside.

Pour the milk, salt and butter in a small saucepan. Por the eggs in slowly.

Cook on medium heat. After 5 minutes, mix the mixture. Wait another 2-3 minutes. Mix again. Scrambled eggs will start to form. Continue cooking and mixing until there is no or little runny eggs. If there is runny eggs, drain in the sink.

FRENCH OMELETTE

READY IN: 10mins

INGREDIENTS

- Butter
- Egg

DIRECTIONS

Have a cast iron skillet with a lid and use the clarified butter to coat well the bottom and sides.

Whip the egg whites with a pinch of salt.Beat the yolks until foamy.Fold egg yolks into egg whites until well incorporated.

Gently pour the eggs mixture into the heated cast iron skillet. Cover and slowly cook until fluffy and done.

I loved making this with a chopped onion that had been browned well, before making the omelette. I poured the eggs on top of the onions and cooked. Then I covered in Hollandaise sauce.

Here we see the recipe list generated from our chosen ingredients. It is clear to see each recipe separately.

We chose to make the internal text a stylish font in order to break up monotony.

PAR Review

- Perception: We've used Maeda's design laws, especially those of organization and time, to simplify the app. We've also implemented agreeable fonts and colors according to user feedback so that a new user understands the system implicitly as soon as he/she sees it. We built it to be idiot-proof.
- Attention: We used bright green coloring buttons to drag attention. Especially with the Find recipes button, it is in the middle of the app and the proverbial bell of the ball, seeing as it is the core raison d'etre of our project. The ingredients list is self-contained in its own box, clearly demarcated.
- Retention: We keep the user occupied with the system. Their ingredients list is both stored in the cache and in their account, so they can see it when they login anywhere. The capacity to edit the list is high, along with the ability to filter the possible recipes that emerge from these lists. They don't need to remember anything, as our app is built to be self-evident.

Laws of Simplicity Review

- The first law we chose to implement here was **organization**. We can clearly see this with the dropdown menu design to add food ingredients. Even though there are dozens of possible ingredients in our list, the fact that it is divided into 2 menus, first to specialize ingredient and then choose the specific one, makes it easier. Additionally, the option to search for the item makes it even easier. There is also organization in terms of the recipe list being sortable and the filter options being clearly laid out logically.
- The second law we chose was the law of **time**. Savings in time feel like simplicity. At every stage, we sought to make the design blatantly obvious and sought to minimize the time needed to parse through our user stories, adding a sense of simplicity to our app. For example, the design of the recipe list, where we can add or remove stuff, is optimized to add/remove items quickly, without much overhead.

Accessibility Review

- https://khan.github.io/tota11y/
- We used the above website to implement our accessibility implementation. To see the accessibility of our website we used tools to automate the process and can tell us what exactly we need to fix.
- One fix was that each button had text and symbols, meaning that color did not matter too much, even if we did use green and red, which could cause issues.
- We also made the buttons large and easy to click to satisfy Fitt's law, along with a larger font, to help people who couldn't see well. Finally, there are no components that involve hearing, which is very helpful for deaf people.

Heuristic Evaluation

- We used the Nielson and Kessler evaluation method.
- Our heuristic evaluation went very well. All our tools performed well, and all actions could be reversed, making it very user fault-tolerant.
- There are easily seen exits, and there is a reset button in the sign up page. Everything, including the ingredient list can be reversed.
- We hide the filter options until the user clicks on the filter button, meaning that we maintain simplicity and don’t bombard the user with information, providing minimalism.
- There is also instantaneous feedback, with the results of the user’s decisions and actions clearly showing up on the UI. There are no side effects to their actions which may har, their browsing experience.

Targeted E – Efficiency Evaluation

- We, as before used a stopwatch and took the average of specific tasks done by our focus group, without them having seen the specific iteration before, in order to view how efficient and speedy our application is. However, our paper prototype readings may be imperfect due to the roughness of the model.
- All metrics rounded to 0 decimal places, we have the same ingredients to add each time and tell everyone to do the same thing each time.

Iteration	Paper Prototype	Balsamiq	Final Node App
Sign Up Task	40 seconds	68 seconds	126 seconds
Add ingredients Task	452 seconds	399 seconds	139 seconds
Filter and view recipes Task	68 seconds	38 seconds	19 seconds
Overall Speed	560 seconds	505 seconds	284 seconds

Focus Group: Final Feedback

- As seen from our targeted E metric, the efficiency and speed of using the app increased throughout the iterations. There were a few points of final feedback from our focus group:
  - The first was that the font used to show the steps in the recipe list wasn’t user-friendly. Had we stuck to a boring but easy to read font like Helvetica or Arial, it would have been easier for users to read them while actually cooking, which was something we hadn’t considered, but overall an easy possible fix to the code.
  - The second was that the simplicity we stressed on bored the users a bit. They felt that there needed to be more color and pictures in the app itself, to hold attention and look beautiful.
  - However, most of the users thought that overall, the app performed its core competency admirably. It clearly and unmistakably showed its purpose and was very easy and quick to use.

Future Work

- Continuing on with the project, with final feedback from our focus groups, here are some of the things we would include:
  - The ability to add your own recipes to the system and edit the recipes in the database and save those to your account
  - More filtering options, including the exact type of food, i.e appetizers, breakfast, dessert
  - Calorie counting tools to augment the health filters on the recipe app

References

- https://www.wellandgood.com/good-food/home-cooking-decline/
- https://corporate.discovery.com/discovery-newsroom/food-network-finishes-2018-up/

- Feel free to browse the code that this project is based on here:
- https://github.com/SunilSwarna/CS545-CookIt
- We will also have added videos to help understand the workflow.