

Appendix

Initial Customer Meeting Dated 23 AUG 2020

Me: Hey, I've noticed that your workflow consists of writing down recipes on a piece of paper and updating that book manually. I have a feeling that we could move your workflow onto a functional website so you can use functionality of websites instead of doing everything manually. Does that sound interesting to you?

Them: Yes. One function I do need is to change ingredient portions depending on the number of servings, as that would greatly help with my workflow. Is that possible to do?

Me: For sure. That should be relatively straightforward to handle. Do you have any other features that you need?

Them: I think being able to search through recipes would be very useful. Other than that, I only need something that works similarly to a normal recipe book but done online. Also, is there going to be any sort of mobile device support? I think that it will mainly be used on mobile unless entering in new recipes.

Me: Yes, I will design with mobile interfaces in mind. I think I have a good idea of what to do now from what you've said, so I'll get started writing a set of success criteria for us to agree on. How does that sound?

Them: That sounds good.

Me: Alright, I'll let you know if I have any questions.

Email from client after showing first design sketches

Good morning,

Thank you so much for the mock website. I'm liking the direction that it's heading in, and the design of it is certainly intuitive to use. However, I just have one thing that I would like to see changed.

On the Update Recipe Page, I think having the button turn another colour when you edit that element will make it very clear which ingredient or method I am editing.

Kind Regards,

(Client Name)

Final Customer Meeting Dated 17 JAN 2021

Me: Hey, after you've been using the product for 1 week, what has your experience been like using it?

Them: It's been an amazing experience. I'm glad that I don't have to write down recipes on paper now. I find adding new recipes extremely intuitive with the interface that you designed, and my experience using it on mobile has been perfect. It's very simple and easy to navigate using the navbar as well as the search function on the homepage. I also enjoyed the very convenient log in method — not having to remember another password was very helpful for my workflow.

Me: I'm glad to hear you like it. Do you have any potential issues with the product so far?

Them: I don't have any issues. You've fixed the one issue I had from the email that I sent a few weeks ago. If I were to give any recommendations however, it would be great if there were a place to add images to recipes or to add a recipe to favourites for easier access, but this is just a minor issue.

Me: I'm glad to see you like this product. If you have any questions in the future or any possible additions you want me to make, please feel free to ask me for them.

Them: That sounds good. Thank you so much for your time and effort spent on this project. I'll be using this hopefully for the next few years.

Code

index.html

```
1  <!doctype html>
2  <html lang="en">
3
4  <head>
5      <title>Recipe List</title>
6      <!-- Required meta tags -->
7      <meta charset="utf-8">
8      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit
=no">
9
10     <!-- Bootstrap CSS -->
11     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/
css/bootstrap.min.css"
12         integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQU0hcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
13     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/
css/bootstrap.min.css"
14         integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQU0hcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
15
16     <!-- FontAwesome Icons -->
17     <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.
css"
18         integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+
A56Ebc1zFSJ" crossorigin="anonymous">
```

```

19
20
21     <!-- My CSS -->
22     <link rel="stylesheet" href="styles.css">
23 </head>
24
25 <body onload="index_onload()">
26     <!-- Optional Bootstrap JavaScript -->
27     <!-- jQuery first, then Popper.js, then Bootstrap JS -->
28     <script src="js/script.js"></script>
29     <script src="js/checkboxes.js"></script>
30     <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
31 integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
abTTE1Pi6jizo" crossorigin="anonymous">
32 </script>
33     <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.
min.js"
34 integrity="sha384-
U02eT0CpHqdSJQ6hJty5KVphtPhzWj9W01clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="
anonymous">
35 </script>
36     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.
js"
37 integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60RQ6VrjIEaFf/nJGzIxFDsf4x0xIM+
B07jRM" crossorigin="anonymous">
38 </script>
39     <!-- Fuse.js for fuzzy searching -->
40     <script src="https://cdn.jsdelivr.net/npm/fuse.js@6.4.3"></script>
41
42     <!-- Navbar -->
43     <nav id="navbar" class="navbar navbar-expand-sm bg-dark navbar-dark">
44     <a class="navbar-brand" href="index.html">Recipes</a>
45     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="
#collapsibleNavbar">
46         <span class="navbar-toggler-icon"></span>
47     </button>
48     <div class="collapse navbar-collapse" id="collapsibleNavbar">
49         <ul class="navbar-nav">
50             <li class="nav-item" id="add-recipe"></li>
51         </ul>
52         <ul class="navbar-nav ml-auto" id="log-in"></ul>
53     </div>
54 </nav>
55
56     <div id="search-bar" class="row justify-content-center navbar-padding">
57     <div class="col-11 col-md-10 col-lg-8">
58         <div class="card-body row no-gutters">
59             <div class="col">
60                 <input id="search-field" class="form-control form-control-lg form-control
-borderless" type="search" placeholder="Search...">
61             </div>
62             <div class="col-auto">
63                 <button id="search-btn" class="btn btn-lg btn-success" type="button"
onclick="search()">Search</button>
64             </div>
65         </div>
66     <div class="col-lg-12">
67         <button type="button" class="btn btn-block btn-outline-dark" data-toggle="
collapse">

```

```

68         data-target="#checkboxes">Advanced Options <span class="fa fa-caret-down"
></span>
69     </button>
70     <div id="checkboxes" class="collapse col-lg-12 padding">
71
72     </div>
73 </div>
74 <div class="padding"></div>
75 </div>
76 </div>
77
78 <div class="justify-content-center row padding">
79 <h1 class="col-lg-8 col-md-10 col-11">Recipes</h1>
80 </div>
81
82 <div class="justify-content-center row" id="items">
83 <div class="spinner-grow spinner-grow-lg text-muted mb-2">
84     <span class="sr-only">Loading...</span>
85 </div>
86 </div>
87
88 <!-- Firebase Initialization -->
89 <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-app.js"></script>
90 <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-analytics.js"></
script>
91 <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-auth.js"></script>
92 <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-firestore.js"></
script>
93 <script src="js/firebase.js"></script>
94 <script src="https://www.gstatic.com/firebasejs/ui/4.7.1/firebase-ui-auth.js"></
script>
95 <script src="js/login-check.js"></script>
96 <script src="js/read-database.js"></script>
97 <script src="js/search.js"></script>
98 </body>
99
100 </html>

```

new-recipe.html

```

1 <!doctype html>
2 <html lang="en">
3
4 <head>
5 <title>Add Recipe</title>
6 <!-- Required meta tags -->
7 <meta charset="utf-8">
8 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=
no">
9
10 <!-- Bootstrap CSS -->
11 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css
/bootstrap.min.css"
12     integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQU0hcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

```

```

13 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css
/bootstrap.min.css"
14 integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
15
16 <!-- FontAwesome Icons -->
17 <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.
css"
18 integrity="sha384-lZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+
A56Ebc1zFSJ" crossorigin="anonymous">
19
20
21 <!-- My CSS -->
22 <link rel="stylesheet" href="styles.css">
23 </head>
24
25 <body onload="checkboxes_onload()">
26 <!-- Optional Bootstrap JavaScript -->
27 <!-- jQuery first, then Popper.js, then Bootstrap JS -->
28 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
29 integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
abtTE1Pi6jizo" crossorigin="anonymous">
30 </script>
31 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min
.js"
32 integrity="sha384-
U02eT0CpHqdSJJQ6hJty5KVphtPhzWj9W01clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="
anonymous">
33 </script>
34 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js
"
35 integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+
B07jRM" crossorigin="anonymous">
36 </script>
37 <script src="js/script.js"></script>
38 <script src="js/checkboxes.js"></script>
39 <script src="js/add-recipe.js"></script>
40
41 <!-- Navbar -->
42 <nav id="navbar" class="navbar navbar-expand-sm bg-dark navbar-dark">
43 <a class="navbar-brand" href="index.html">Recipes</a>
44 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="
#collapsibleNavbar">
45 <span class="navbar-toggler-icon"></span>
46 </button>
47 <div class="collapse navbar-collapse" id="collapsibleNavbar">
48 <ul class="navbar-nav">
49 <li class="nav-item" id="add-recipe"></li>
50 </ul>
51 <ul class="navbar-nav ml-auto" id="log-in"></ul>
52 </div>
53 </nav>
54
55 <div class="col-lg-8 col-12 navbar-padding" style="margin: auto;">
56 <form>
57 <div class="form-group">
58 <label for="recipe-name">Recipe Name:</label>
59 <input type="text" class="form-control" id="recipe-name" placeholder="Enter
Recipe Name" name="recipe-name">

```

```

60     </div>
61     <div class="form-group">
62         <label for="servings">Serves:</label>
63         <input type="number" class="form-control" id="servings" placeholder="How many
64         people?" name="servings"
65         oninput="this.value = this.value.replace(/[^0-9.]/g, '').replace(/(\..*)\./g, '$1');">
66     </div>
67     <div class="form-group">
68         <label for="tags">Tags:</label><br>
69         <div id="option-selector" class="col-lg-12">
70             <div id="form-check" class="col-lg-12">
71                 <div id="checkboxes" class="col-lg-12"></div>
72             </div>
73         </div>
74         <div class="form-group">
75             <div class="row">
76                 <div class="col-lg-2">
77                     <label for="quantity">Quantity:</label>
78                     <input type="text" class="form-control" id="quantity" placeholder="Enter
79                     quantity" name="quantity" oninput="this.value = this.value.replace(/[^0-9.]/g, '').
80                     replace(/(\..*)\./g, '$1');">
81                 </div>
82                 <div class="col-lg-2">
83                     <label for="unit">Units:</label>
84                     <input type="text" class="form-control" id="unit" placeholder="Enter unit
85                     " name="unit">
86                 </div>
87                 <div class="col-lg-8">
88                     <label for="ingredient">Ingredients:</label>
89                     <input type="text" class="form-control" id="ingredient" placeholder="
90                     Enter ingredient" name="ingredient">
91                 </div>
92             </div>
93             <div class="row">
94                 <div class="col-lg-12">
95                     <button type="button" class="btn btn-primary m-lg-3" onclick="
96                     add_ingredients()"
97                     id="ingredient-btn">Add</button>
98                 </div>
99             </div>
100             <div class="row">
101                 <div class="col-lg-12">
102                     <p>Ingredients List:</p>
103                     <div>
104                         <ul id="ingredients-list"></ul>
105                     </div>
106                 </div>
107             </div>
108             <div class="form-group">
109                 <div class="row">
110                     <div class="col-lg-12">
111                         <label for="method">Method:</label>
112                         <input type="text" class="form-control" id="method" placeholder="Enter
113                         method" name="method">
114                     </div>
115                 </div>
116             </div>

```

```

111         <div class="row">
112             <div class="col-lg-12">
113                 <button type="button" class="btn btn-primary m-lg-3" onclick="add_methods
114                 ()" id="method-btn">Add</button>
115             </div>
116         </div>
117         <div class="row">
118             <div class="col-lg-12">
119                 <p>Method List:</p>
120                 <div>
121                     <ol id="methods-list"></ol>
122                 </div>
123             </div>
124         </div>
125         <div id="form-buttons">
126             <button type="button" id="submit-button" class="btn btn-primary btn-block"
127             onclick="get_recipe_data()">Submit</button>
128         </div>
129     </div>
130
131     <!-- Firebase Initialization -->
132     <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-app.js"></script>
133     <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-analytics.js"></
134     script>
135     <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-auth.js"></script>
136     <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-firestore.js"></
137     script>
138     <script src="js/firebase.js"></script>
139     <script src="https://www.gstatic.com/firebasejs/ui/4.7.1/firebase-ui-auth.js"></
140     script>
141     <script src="js/login-check.js"></script>
142     <script src="js/read-database.js"></script>
143
144 </body>
145
146 </html>

```

recipe.html

```

1  <!doctype html>
2  <html lang="en">
3
4  <head>
5      <!-- This will be updated by the code -->
6      <title>Recipe</title>
7
8      <!-- Required meta tags -->
9      <meta charset="utf-8">
10     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=
11     no">
12
13     <!-- Bootstrap CSS -->
14     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css
15     /bootstrap.min.css"

```

```

14     integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQU0hcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
15     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css
/bootstrap.min.css"
16     integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQU0hcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
17
18     <!-- FontAwesome Icons -->
19     <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.
css"
20     integrity="sha384-1ZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+
A56Ebc1zFSJ" crossorigin="anonymous">
21
22     <!-- My CSS -->
23     <link rel="stylesheet" href="styles.css">
24 </head>
25
26 <body onload="render_recipe()">
27     <!-- Optional Bootstrap JavaScript -->
28     <!-- jQuery first, then Popper.js, then Bootstrap JS -->
29     <script src="js/script.js"></script>
30     <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
31     integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
abtTE1Pi6jizo" crossorigin="anonymous">
32     </script>
33     <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min
.js"
34     integrity="sha384-
U02eT0CpHqdsJQ6hJty5KVphtPhzWj9W01clHTMGa3JDZwrnQq4sF86dIHNDz0W1" crossorigin="
anonymous">
35     </script>
36     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js
"
37     integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60Rq6VrjIEaFf/nJGzIxFDsf4x0xIM+
B07jRM" crossorigin="anonymous">
38     </script>
39
40     <!-- Navbar -->
41     <nav id="navbar" class="navbar navbar-expand-sm bg-dark navbar-dark">
42         <a class="navbar-brand" href="index.html">Recipes</a>
43         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="
#collapsibleNavbar">
44             <span class="navbar-toggler-icon"></span>
45         </button>
46         <div class="collapse navbar-collapse" id="collapsibleNavbar">
47             <ul class="navbar-nav">
48                 <li class="nav-item" id="add-recipe"></li>
49             </ul>
50             <ul class="navbar-nav ml-auto" id="log-in"></ul>
51         </div>
52     </nav>
53
54     <div id="title-box" class="row justify-content-center navbar-padding"></div>
55
56     <div id="content" class="row justify-content-center navbar-padding">
57         <div class="col-11 col-md-10 col-lg-3" style="display: flex;">
58             <div id="ingredients" class="card-body" style="flex: 1;">
59                 <h3>Ingredients:</h3>
60                 <div id="servings" class="row col-lg-12">

```



```

61         <h5>Serves:</h5>
62     </div>
63     <ul id="list"></ul>
64 </div>
65 </div>
66 <div class="col-11 col-md-10 col-lg-5" style="flex: 1;">
67     <div id="method" class="card-body no-gutters">
68         <h3>Method:</h3>
69         <ol id="method-list"></ol>
70     </div>
71 </div>
72 </div>
73 <div id="edit-button" class="row justify-content-center">
74
75 </div>
76
77
78 <!-- Firebase Initialization -->
79 <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-app.js"></script>
80 <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-analytics.js"></
script>
81 <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-auth.js"></script>
82 <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-firestore.js"></
script>
83 <script src="js/firebase.js"></script>
84 <script src="https://www.gstatic.com/firebasejs/ui/4.7.1/firebase-ui-auth.js"></
script>
85 <script src="js/login-check.js"></script>
86 <script src="js/read-database.js"></script>
87 <script src="js/change-recipe-portions.js"></script>
88
89 </body>
90
91 </html>

```

sign-in.html

```

1  <!doctype html>
2  <html lang="en">
3
4  <head>
5      <title>Recipe List</title>
6      <!-- Required meta tags -->
7      <meta charset="utf-8">
8      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=
no">
9
10     <!-- Bootstrap CSS -->
11     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css
/bootstrap.min.css"
12         integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
13     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css
/bootstrap.min.css"
14         integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/
iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

```

```

15     <!-- FontAwesome Icons -->
16     <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.0/css/all.
17     css"
18         integrity="sha384-1ZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+
19     A56Ebc1zFSJ" crossorigin="anonymous">
20
21     <!-- My CSS -->
22     <link rel="stylesheet" href="styles.css">
23
24     <!-- Firebase CSS -->
25     <link type="text/css" rel="stylesheet" href="https://www.gstatic.com/firebasejs/ui
26     /4.7.1/firebase-ui-auth.css" />
27 </head>
28
29 <body>
30     <!-- Optional Bootstrap JavaScript -->
31     <!-- jQuery first, then Popper.js, then Bootstrap JS -->
32     <script src="js/script.js"></script>
33     <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
34     integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8
35     abtTE1Pi6jizo" crossorigin="anonymous">
36     </script>
37     <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min
38     .js"
39     integrity="sha384-
40     U02eT0CpHqdSJJQ6hJty5KVphtPhzWj9W01c1HTMga3JDZwrnQq4sF86dIHNDz0W1" crossorigin="
41     anonymous">
42     </script>
43     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js
44     "
45     integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy60rQ6VrjIEaFf/nJGzIxFDsf4x0xIM+
46     B07jRM" crossorigin="anonymous">
47     </script>
48
49     <!-- Navbar -->
50     <nav id="navbar" class="navbar navbar-expand-sm bg-dark navbar-dark">
51         <a class="navbar-brand" href="index.html">Recipes</a>
52         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="
53         #collapsibleNavbar">
54             <span class="navbar-toggler-icon"></span>
55         </button>
56         <div class="collapse navbar-collapse" id="collapsibleNavbar">
57             <ul class="navbar-nav">
58                 <li class="nav-item" id="add-recipe"></li>
59             </ul>
60             <ul class="navbar-nav ml-auto" id="log-in"></ul>
61         </div>
62     </nav>
63
64     <div class="col-lg-8 col-12 navbar-padding" style="margin: auto; text-align: center
65     ;">
66         <h1>Login</h1>
67     </div>
68
69     <div class="horizontal-line"></div>
70
71     <div id="firebaseui-auth-container" class="bot-spacing"></div>

```

```

63     <!-- Firebase Initialization -->
64     <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-app.js"></script>
65     <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-analytics.js"></
script>
66     <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-auth.js"></script>
67     <script src="https://www.gstatic.com/firebasejs/8.1.2/firebase-firestore.js"></
script>
68     <script src="js/firebase.js"></script>
69     <script src="https://www.gstatic.com/firebasejs/ui/4.7.1/firebase-ui-auth.js"></
script>
70     <script src="js/login.js"></script>
71     <script src="js/login-check.js"></script>
72
73 </body>
74
75 </html>

```

add-recipe.js

```

1     function check_empty(object) {
2         let name = document.getElementById("recipe-name").value;
3         let servings = document.getElementById("servings").value;
4         let ingredients = get_list("ingredient-reader");
5         let method = get_list("method-reader");
6         if (!name || !servings || ingredients.length === 0 || method.length === 0) {
7             alert_empty();
8             return true;
9         }
10        return false;
11    }
12
13    function get_list(id) {
14        let array = [];
15        let list = document.getElementsByClassName(id);
16        for (let i = 0; i < list.length; i++) {
17            array.push(list[i].textContent.trim());
18        }
19        return array;
20    }
21
22    function read_tags() {
23        let tags = [];
24        let checkboxes = document.getElementsByName("option-selector");
25        for (let i = 0; i < checkboxes.length; i++) {
26            if (checkboxes[i].checked) {
27                tags.push(checkboxes[i].value);
28            }
29        }
30        return tags;
31    }
32
33    function capitalize_words(word) {
34        let array = word.split(" ");
35        for(let i = 0; i<array.length; i++) {
36            array[i] = array[i][0].toUpperCase() + array[i].substr(1);
37        }

```

```

38     return array.join(" ")
39 }
40
41 function split_ingredients(array) {
42     let quantity = [];
43     let unit = [];
44     let name = [];
45
46     for (let i = 0; i < array.length; i++) {
47         quantity.push(array[i].quantity.trim());
48         unit.push(array[i].unit.trim());
49         name.push(capitalize_words(array[i].ingredient.trim()));
50     }
51     return [quantity, unit, name]
52 }
53
54 function get_recipe_data() {
55     if (check_empty()) {
56         alert_empty();
57         return
58     }
59     let name = capitalize_words(document.getElementById("recipe-name").value);
60     let servings = document.getElementById("servings").value;
61     let tags = read_tags();
62     let ingredients_array = split_ingredients(ingredients);
63
64     for(let i = 0; i<methods.length; i++) {
65         methods[i] = methods[i][0].toUpperCase() + methods[i].substr(1);
66     }
67
68     let docData = {
69         "ingredients-quantity": ingredients_array[0],
70         "ingredients-unit": ingredients_array[1],
71         "ingredients-name": ingredients_array[2],
72         method: methods,
73         name: name,
74         servings: servings,
75         tags: tags,
76     }
77
78     db.collection("recipes").add(docData)
79     .then(function(docRef) {
80         console.log("Document written with ID: ", docRef.id);
81         clear_fields();
82         location.assign("index.html");
83     })
84     .catch(function(error) {
85         console.error("Error adding document: ", error);
86         alert("You need to be logged in to an administrator account to do this!");
87     });
88 }
89
90 function update_recipe(id) {
91     if (check_empty()) {
92         alert_empty();
93         return
94     }
95     let name = capitalize_words(document.getElementById("recipe-name").value);
96     let servings = document.getElementById("servings").value;

```

```

97     let tags = read_tags();
98     let ingredients_array = split_ingredients(ingredients);
99     for(let i = 0; i<methods.length; i++) {
100         methods[i] = methods[i][0].toUpperCase() + methods[i].substr(1);
101     }
102
103     let docData = {
104         "ingredients-quantity": ingredients_array[0],
105         "ingredients-unit": ingredients_array[1],
106         "ingredients-name": ingredients_array[2],
107         method: methods,
108         name: name,
109         servings: parseInt(servings),
110         tags: tags,
111     }
112
113     db.collection("recipes").doc(id).set(docData)
114     .then(function() {
115         console.log("Document updated");
116         clear_fields();
117         location.assign("index.html");
118     })
119     .catch(function(error) {
120         console.error("Error adding document: ", error);
121         alert("You need to be logged in to an administrator account to do this!");
122     });
123
124 }
125
126 function clear_fields() {
127     document.getElementById("recipe-name").value = "";
128     document.getElementById("servings").value = "";
129     clear_list("ingredients-list");
130     clear_list("methods-list");
131 }
132
133 function clear_list(id) {
134     let list = document.getElementById(id);
135     list.innerHTML = "";
136 }
137
138 function alert_empty() {
139     let name = document.getElementById("recipe-name").value;
140     let servings = document.getElementById("servings").value;
141     let ingredients = get_list("ingredient-reader");
142     let method = get_list("method-reader");
143     if (!name) {
144         document.getElementById("recipe-name").setAttribute("class", "form-control
145 warning");
146         setTimeout(function(){ document.getElementById("recipe-name").setAttribute("
147 class", "form-control"); }, 3000);
148     }
149     if (!servings) {
150         document.getElementById("servings").setAttribute("class", "form-control
151 warning");
152         setTimeout(function(){ document.getElementById("servings").setAttribute("
153 class", "form-control"); }, 3000);
154     }

```

```

152
153     if(ingredients.length === 0) {
154         document.getElementById("quantity").setAttribute("class", "form-control
warning");
155         document.getElementById("unit").setAttribute("class", "form-control warning")
;
156         document.getElementById("ingredient").setAttribute("class", "form-control
warning");
157         setTimeout(function(){ document.getElementById("quantity").setAttribute("
class", "form-control"); }, 3000);
158         setTimeout(function(){ document.getElementById("unit").setAttribute("class",
"form-control"); }, 3000);
159         setTimeout(function(){ document.getElementById("ingredient").setAttribute("
class", "form-control"); }, 3000);
160     }
161
162     if(method.length === 0) {
163         document.getElementById("method").setAttribute("class", "form-control warning
");
164         setTimeout(function(){ document.getElementById("method").setAttribute("class"
, "form-control"); }, 3000);
165     }
166
167     focus_boxes("recipe-name");
168 }

```

change-recipe-portions.js

```

1     async function change_servings(input_box) {
2         let URLParams = new URLSearchParams(window.location.search);
3         let id = URLParams.get("id");
4         if (!localStorage.getItem("recipes")) {
5             await load_recipes();
6         }
7         let recipes = JSON.parse(localStorage.getItem("recipes"));
8         let index = recipes.findIndex((elem) => {
9             return elem.id === id;
10        });
11        if (index === -1) {
12            location.replace("index.html");
13        }
14        let default_recipe = recipes[index];
15
16        let servings_value = input_box.value;
17        let default_servings = default_recipe["servings"];
18
19        let ratio = servings_value/default_servings;
20
21        let quantity;
22        let quantity_array = [];
23        for(let i = 0; i<default_recipe["ingredients-quantity"].length; i++) {
24            quantity = default_recipe["ingredients-quantity"][i];
25            let modified_value = quantity * ratio
26            if(is_integer(modified_value)) {
27                quantity_array.push(parseInt(modified_value));
28            } else {

```

```

29         quantity_array.push(Math.round(modified_value*10)/10);
30     }
31 }
32 let new_recipe = {
33     servings: servings_value,
34     "ingredients-name": default_recipe["ingredients-name"],
35     "ingredients-unit": default_recipe["ingredients-unit"],
36     "ingredients-quantity": quantity_array,
37     method: default_recipe.method,
38 }
39 render_ingredients(new_recipe);
40 }
41
42 async function change_ingredient(input_box) {
43     let name = input_box.name;
44     let URLParams = new URLSearchParams(window.location.search);
45     let id = URLParams.get("id");
46     if (!localStorage.getItem("recipes")) {
47         await load_recipes();
48     }
49     let recipes = JSON.parse(localStorage.getItem("recipes"));
50     let id_index = recipes.findIndex((elem) => {
51         return elem.id === id;
52     });
53     if (id_index === -1) {
54         location.replace("index.html");
55     }
56     let default_recipe = recipes[id_index];
57     let default_quantities = default_recipe["ingredients-quantity"];
58     let default_ingredients = default_recipe["ingredients-name"];
59
60     let ingredient_index = default_ingredients.findIndex((elem) => {
61         return elem === name;
62     });
63     if (ingredient_index === -1) {
64         location.replace("index.html");
65     }
66     let quantity
67     let quantity_array = [];
68     let quantity_box_value = input_box.value;
69     let ratio = quantity_box_value/default_quantities[ingredient_index];
70
71     for(let i = 0; i<default_quantities.length; i++) {
72         quantity = default_quantities[i];
73         let modified_value = quantity * ratio
74         if(is_integer(modified_value)) {
75             quantity_array.push(parseInt(modified_value));
76         } else {
77             quantity_array.push(Math.round(modified_value*10)/10);
78         }
79     }
80     let new_recipe = {
81         servings: default_recipe.servings,
82         "ingredients-name": default_recipe["ingredients-name"],
83         "ingredients-unit": default_recipe["ingredients-unit"],
84         "ingredients-quantity": quantity_array,
85         method: default_recipe.method,
86     }
87     render_ingredients(new_recipe);

```

```

88     let servings_input = document.getElementById("servings_input_box");
89     servings_input.value = parseInt(default_recipe.servings * ratio);
90 }
91
92 function is_integer(number) {
93     return (number / parseInt(number) === 1)
94 }

```

checkboxes.js

```

1  const listeners = [
2      ["method", "method-btn"],
3      ["ingredient", "ingredient-btn"],
4      ["quantity", "ingredient-btn"],
5      ["unit", "ingredient-btn"],
6  ]
7
8  var ingredients = [];
9  var methods = [];
10
11 function checkboxes() {
12     let div = document.getElementById("checkboxes");
13     for (let i = 0; i < tags.length; i++) {
14         let checkbox = document.createElement("input");
15         checkbox.type = "checkbox";
16         checkbox.name = "option-selector";
17         checkbox.value = tags[i];
18         checkbox.setAttribute("id", `${checkbox.name}-${tags[i]}`);
19         checkbox.setAttribute("class", 'form-check-input custom-control-input');
20
21         let label = document.createElement("label");
22         label.setAttribute("for", `${checkbox.name}-${tags[i]}`);
23         label.setAttribute("class", 'form-check-label custom-control-label');
24         let t = document.createTextNode(`${tags[i]}`);
25         label.appendChild(t);
26
27         let d = document.createElement("DIV");
28         d.setAttribute("class", "custom-control custom-checkbox");
29
30         let br = document.createElement("BR");
31
32         d.appendChild(checkbox);
33         d.appendChild(label);
34         d.appendChild(br);
35         div.appendChild(d);
36     }
37 }
38
39 async function check_id_exists() {
40     let URLParams = new URLSearchParams(window.location.search);
41     let id = URLParams.get("id");
42
43     if (id === null) {
44         return
45     }
46 }

```



```

47     let temp_ingredients = [];
48
49     if (!localStorage.getItem("recipes")) {
50         await load_recipes();
51     }
52
53     let recipes = JSON.parse(localStorage.getItem("recipes"));
54
55     let index = recipes.findIndex((elem) => {
56         return elem.id === id;
57     });
58
59     if (index === -1) {
60         location.replace("index.html");
61     }
62
63     let recipe = recipes[index];
64     let name = document.getElementById("recipe-name");
65     let servings = document.getElementById("servings");
66     let checkboxes = document.getElementsByName("option-selector");
67     for (let i = 0; i < recipe["ingredients-name"].length; i++) {
68         temp_ingredients.push({
69             quantity: recipe["ingredients-quantity"][i],
70             unit: recipe["ingredients-unit"][i],
71             ingredient: recipe["ingredients-name"][i],
72         });
73     }
74     ingredients = [...temp_ingredients];
75     methods = [...recipe["method"]];
76     name.value = recipe["name"];
77     servings.value = recipe["servings"];
78     let tags = recipe["tags"];
79     render_ingredient_list(ingredients);
80     render_method_list(methods);
81
82     for (let i = 0; i < checkboxes.length; i++) {
83         if (tags.includes(checkboxes[i].value)) {
84             checkboxes[i].checked = true;
85         }
86     }
87
88     let submit_button = document.getElementById("submit-button");
89     submit_button.innerText = "";
90     submit_button.appendChild(document.createTextNode("Update"));
91     submit_button.setAttribute("onclick", `update_recipe("${id}")`);
92     document.title = recipe.name;
93
94     let form_div = document.getElementById("form-buttons");
95     let delete_button = document.createElement("button");
96     delete_button.setAttribute("class", "btn btn-danger btn-block bot-spacing");
97     delete_button.setAttribute("type", "button");
98     delete_button.setAttribute("recipe_id", id);
99     delete_button.setAttribute("onclick", "delete_recipe(this)");
100    delete_button.appendChild(document.createTextNode("Delete"));
101    form_div.appendChild(delete_button);
102}
103
104    async function delete_recipe(b) {
105        let id = b.getAttribute('recipe_id');

```

```

106
107     db.collection("recipes").doc(id).delete().then(function () {
108         console.log("Document successfully deleted!");
109         location.replace("index.html");
110     }).catch(function (error) {
111         console.error("Error removing document: ", error);
112     });
113 }
114
115 function checkboxes onload() {
116     checkboxes();
117     add_event_listeners(listeners);
118     check_id_exists();
119 }

```

```

1  function add_event_listeners(listeners) {
2      let module;
3      for (let i = 0; i < listeners.length; i++) {
4          module = document.getElementById(listeners[i][0]);
5          module.addEventListener('keypress', function (event) {
6              if (event.code === "Enter") {
7                  event.preventDefault(); // Cancel the default action
8                  document.getElementById(listeners[i][1]).click(); // Click button
9              }
10         });
11     }
12 }
13
14 function clear_ingredient_boxes() {
15     document.getElementById("quantity").value = "";
16     document.getElementById("ingredient").value = "";
17     document.getElementById("unit").value = "";
18 }
19
20 function clear_method_boxes() {
21     document.getElementById("method").value = "";
22 }
23
24 function focus_boxes(id) {
25     document.getElementById(id).focus();
26     document.getElementById(id).select();
27 }
28
29 function move(f, t) {
30     // remove 'from' item and store it
31     var f = data.splice(from, 1)[0];
32     // insert stored item into position 'to'
33     data.splice(t, 0, f);
34 }
35
36
37 function move_item(b, type, movement) {
38     let array;
39     let name;
40     let index;
41     if (type == "i") {
42         array = ingredients;
43         name = b.getAttribute("ingredient");
44         index = array.findIndex((elem) => {

```

```

45         return elem.ingredient === name;
46     });
47     } else if (type == "m") {
48         array = methods;
49         name = b.getAttribute("method");
50         index = array.findIndex((elem) => {
51             return elem === name;
52         });
53     }
54
55     if ((index + 1 == array.length) && movement == 1) {
56         return
57     }
58     if (index == 0 && movement == -1) {
59         return
60     }
61
62     [array[index], array[index + movement]] = [array[index + movement], array[index
63     ]];
64
65     type == "i" ? render_ingredient_list(ingredients) : render_method_list(methods);
66 }
67
68 function add_ingredients() {
69     let flags = false
70     let ingredient = document.getElementById("ingredient").value.trim();
71     let quantity = document.getElementById("quantity").value.replace(/\s+/g, '');
72     let units = document.getElementById("unit").value.trim();
73
74     if (isNaN(quantity) || ingredient === "" || quantity < 0) {
75         clear_ingredient_boxes();
76         focus_boxes("quantity");
77         flags = true;
78     }
79
80     let t;
81
82     if (flags === false) {
83         let object = {
84             quantity: quantity,
85             unit: units,
86             ingredient: ingredient,
87         };
88
89         ingredients.push(object);
90         ingredient_list_add();
91     }
92 }
93
94 function make_ingredient_buttons(index) {
95     let div = document.createElement("div");
96     div.setAttribute("class", "no-gutters");
97     let edit_b = document.createElement("b");
98     edit_b.setAttribute("class", "btn btn-sm btn-primary");
99     edit_b.appendChild(document.createTextNode("Edit"));
100    edit_b.setAttribute("ingredient", `${ingredients[index].ingredient}`);
101    edit_b.setAttribute("onclick", `edit_ingredient_index(this)`);
102    let delete_b = document.createElement("b");

```

```

103     delete_b.setAttribute("class", "btn btn-sm btn-danger right-gap");
104     delete_b.appendChild(document.createTextNode("Delete"));
105     delete_b.setAttribute("onclick", 'delete_ingredient_index(this)');
106     delete_b.setAttribute("ingredient", `${ingredients[index].ingredient}`);
107     let up_arrow = document.createElement("b");
108     up_arrow.setAttribute("class", "fa fa-lg fa-arrow-up right-gap");
109     up_arrow.setAttribute("onclick", "move_item(this, 'i', -1)");
110     up_arrow.setAttribute("ingredient", `${ingredients[index].ingredient}`);
111     up_arrow.setAttribute("id", "ingredient_up_arrow");
112     let down_arrow = document.createElement("b");
113     down_arrow.setAttribute("class", "fa fa-lg fa-arrow-down");
114     down_arrow.setAttribute("onclick", "move_item(this, 'i', 1)");
115     down_arrow.setAttribute("ingredient", `${ingredients[index].ingredient}`);
116     down_arrow.setAttribute("id", "ingredient_down_arrow");
117     div.appendChild(edit_b);
118     div.appendChild(delete_b);
119     div.appendChild(up_arrow);
120     div.appendChild(down_arrow);
121     return div;
122 }
123
124 function ingredient_list_add() {
125     render_ingredient_list(ingredients);
126     focus_boxes("quantity");
127 }

```

```

1
2 function edit_ingredient_index(b) {
3     let success_buttons = document.getElementsByClassName("btn btn-sm btn-success
    ingredient");
4     if (success_buttons.length !== 0) {
5         for (let x = 0; x < success_buttons.length; x++) {
6             success_buttons[x].setAttribute("class", "btn btn-sm btn-primary");
7         }
8         clear_ingredient_boxes();
9     }
10    let name = b.getAttribute("ingredient");
11    let index = ingredients.findIndex((elem) => {
12        return elem.ingredient === name;
13    });
14    if (index == -1) {
15        location.replace("index.html");
16    }
17    b.setAttribute("class", "btn btn-sm btn-success ingredient");
18    focus_boxes("quantity");
19    let btn = document.getElementById("ingredient-btn");
20    btn.innerHTML = "";
21    btn.appendChild(document.createTextNode("Update"));
22    btn.setAttribute("onclick", "submit_ingredient_edits(this)");
23    btn.setAttribute("index", index)
24    let q = document.getElementById("quantity");
25    let u = document.getElementById("unit");
26    let i = document.getElementById("ingredient");
27    q.value = ingredients[index].quantity;
28    u.value = ingredients[index].unit;
29    i.value = ingredients[index].ingredient;
30 }
31
32 function delete_ingredient_index(b) {

```

```

33 let name = b.getAttribute("ingredient");
34 let index = ingredients.findIndex((elem) => {
35     return elem.ingredient === name;
36 });
37 if (index > -1) {
38     ingredients.splice(index, 1);
39 }
40 render_ingredient_list(ingredients);
41 let btn = document.getElementById("ingredient-btn");
42 btn.innerHTML = "";
43 btn.appendChild(document.createTextNode("Add"));
44 btn.setAttribute("onclick", "add_ingredients()");
45 btn.setAttribute("index", "");
46 }
47
48 function submit_ingredient_edits(button) {
49     index = button.getAttribute("index");
50     let q = document.getElementById("quantity").value;
51     let u = document.getElementById("unit").value;
52     let i = document.getElementById("ingredient").value;
53     ingredients[index].quantity = q;
54     ingredients[index].unit = u;
55     ingredients[index].ingredient = i;
56     let btn = document.getElementById("ingredient-btn");
57     btn.innerHTML = "";
58     btn.appendChild(document.createTextNode("Add"));
59     btn.setAttribute("onclick", "add_ingredients()");
60     btn.setAttribute("index", "");
61     clear_ingredient_boxes();
62     render_ingredient_list(ingredients);
63 }
64
65 function add_methods() {
66     let method = document.getElementById("method").value.trim();
67
68     if (method === "") {
69         clear_method_boxes();
70         focus_boxes("method");
71         return
72     }
73
74     methods.push(method);
75     render_method_list(methods);
76 }
77
78 function make_method_buttons(index) {
79     let div = document.createElement("div");
80     div.setAttribute("class", "no-gutters");
81     let edit_b = document.createElement("b");
82     edit_b.setAttribute("class", "btn btn-sm btn-primary");
83     edit_b.appendChild(document.createTextNode("Edit"));
84     edit_b.setAttribute("method", `${methods[index]}`);
85     edit_b.setAttribute("onclick", `edit_method_index(this)`);
86     let delete_b = document.createElement("b");
87     delete_b.setAttribute("class", "btn btn-sm btn-danger right-gap");
88     delete_b.appendChild(document.createTextNode("Delete"));
89     delete_b.setAttribute("onclick", `delete_method_index(this)`);
90     delete_b.setAttribute("method", `${methods[index]}`);
91     let up_arrow = document.createElement("b");

```

```

92 up_arrow.setAttribute("class", "fa fa-lg fa-arrow-up right-gap");
93 up_arrow.setAttribute("onclick", "move_item(this, 'm', -1)");
94 up_arrow.setAttribute("method", `${methods[index]}`);
95 let down_arrow = document.createElement("b");
96 down_arrow.setAttribute("class", "fa fa-lg fa-arrow-down");
97 down_arrow.setAttribute("onclick", "move_item(this, 'm', 1)");
98 down_arrow.setAttribute("method", `${methods[index]}`);
99 div.appendChild(edit_b);
100 div.appendChild(delete_b);
101 div.appendChild(up_arrow);
102 div.appendChild(down_arrow);
103 return div;
104 }
105
106 function method_list_add() {
107     render_method_list(methods);
108     focus_boxes("method");
109 }

```

```

1 function render_method_list(array) {
2     let UL = document.getElementById("methods-list");
3     UL.innerHTML = "";
4     let LI;
5     let t;
6
7     for (let i = 0; i < array.length; i++) {
8         div = make_method_buttons(i);
9         LI = document.createElement("LI");
10        t = document.createTextNode(`${array[i]}`);
11        LI.appendChild(t);
12        LI.appendChild(div);
13        LI.setAttribute("class", "method-reader");
14        UL.appendChild(LI);
15    }
16    clear_method_boxes();
17 }
18
19 function edit_method_index(b) {
20     let success_buttons = document.getElementsByClassName("btn btn-sm btn-success method");
21     if (success_buttons.length !== 0) {
22         for (let x = 0; x < success_buttons.length; x++) {
23             success_buttons[x].setAttribute("class", "btn btn-sm btn-primary");
24         }
25         clear_method_boxes();
26     }
27     let name = b.getAttribute("method");
28     let index = methods.findIndex((elem) => {
29         return elem === name;
30     });
31     if (index == -1) {
32         location.replace("index.html");
33     }
34     b.setAttribute("class", "btn btn-sm btn-success method");
35     focus_boxes("method");
36     let btn = document.getElementById("method-btn");
37     btn.innerHTML = "";
38     btn.appendChild(document.createTextNode("Update"));
39     btn.setAttribute("onclick", "submit_method_edits(this)");

```

```

40     btn.setAttribute("index", index);
41     let m = document.getElementById("method");
42     m.value = methods[index];
43 }
44
45 function delete_method_index(b) {
46     let name = b.getAttribute("method");
47     let index = methods.findIndex((elem) => {
48         return elem === name;
49     });
50     if (index > -1) {
51         methods.splice(index, 1);
52     }
53     render_method_list(methods);
54     let btn = document.getElementById("method-btn");
55     btn.innerHTML = "";
56     btn.appendChild(document.createTextNode("Add"));
57     btn.setAttribute("onclick", "add_methods()");
58     btn.setAttribute("index", "");
59 }
60
61 function submit_method_edits(button) {
62     index = button.getAttribute("index");
63     let m = document.getElementById("method").value;
64     methods[index] = m;
65     let btn = document.getElementById("method-btn");
66     btn.innerHTML = "";
67     btn.appendChild(document.createTextNode("Add"));
68     btn.setAttribute("onclick", "add_methods()");
69     btn.setAttribute("index", "");
70     clear_method_boxes();
71     method_list_add();
72 }

```

firebase.js

```

1  // Your web app's Firebase configuration
2  // For Firebase JS SDK v7.20.0 and later, measurementId is optional
3  var firebaseConfig = {
4      apiKey: "AIzaSyD3Hlg2IxQqfjXxxskqNx-oH1HtPZYM11k",
5      authDomain: "recipe-database-list.firebaseio.com",
6      databaseURL: "https://recipe-database-list.firebaseio.com",
7      projectId: "recipe-database-list",
8      storageBucket: "recipe-database-list.appspot.com",
9      messagingSenderId: "18138758400",
10     appId: "1:18138758400:web:d947b75a5c4d1d675887e5",
11     measurementId: "G-PBQ309NTQL"
12 };
13 // Initialize Firebase
14 firebase.initializeApp(firebaseConfig);
15 firebase.analytics();
16
17 var db = firebase.firestore();

```

login-check.js

```
1  function sign_out() {
2      firebase.auth().signOut().then(function () {
3          // Sign-out successful.
4      }).catch(function (error) {
5          // An error happened.
6      });
7  }
8
9  function create_recipe_navbar(uid) {
10     console.log(uid);
11     let usersRef = db.collection("users").doc(uid);
12     usersRef.get().then(function (doc) {
13         if (doc.exists) {
14             let li = document.getElementById("add-recipe");
15             let a = document.createElement("a");
16             a.setAttribute("href", "new-recipe.html");
17             a.setAttribute("class", "nav-link");
18             let t = document.createTextNode("Add New Recipe");
19             a.appendChild(t);
20             li.appendChild(a);
21         } else {
22             console.log("bentao");
23         }
24     })
25 }
26
27 initApp = function () {
28     firebase.auth().onAuthStateChanged(function (user) {
29         if (user) {
30             clearDiv("log-in");
31             clearDiv("add-recipe");
32             let ul = document.getElementById("log-in");
33             let li = document.createElement("LI");
34             li.setAttribute("class", "nav-item");
35             let a = document.createElement("A");
36             a.setAttribute("class", "nav-link");
37             a.setAttribute("onclick", "sign_out()");
38             a.setAttribute("href", "index.html");
39             let span = document.createElement("SPAN");
40             span.setAttribute("class", "fas fa-sign-out-alt");
41             let t = document.createTextNode(" Sign Out");
42             a.appendChild(span);
43             a.appendChild(t);
44             li.appendChild(a);
45             ul.appendChild(li);
46             logged_in = true;
47             let uid = user.uid
48             create_recipe_navbar(uid);
49             if(window.location.pathname == "/recipe.html") {
50                 create_edit_button();
51             }
52         } else {
53             clearDiv("log-in");
54             clearDiv("add-recipe");
55             let ul = document.getElementById("log-in");
56             let li = document.createElement("LI");
```



```

57     li.setAttribute("class", "nav-item");
58     let a = document.createElement("A");
59     a.setAttribute("class", "nav-link");
60     a.setAttribute("href", "sign-in.html");
61     let span = document.createElement("SPAN");
62     span.setAttribute("class", "fas fa-sign-in-alt");
63     let t = document.createTextNode(" Sign In");
64     a.appendChild(span);
65     a.appendChild(t);
66     li.appendChild(a);
67     ul.appendChild(li);
68     logged_in = false;
69     // create_edit_button();
70   }
71   }, function (error) {
72     console.log(error);
73   });
74 };
75
76 window.addEventListener('load', function () {
77   initApp()
78 });

```

login.js

```

1  var uiConfig = {
2    signInSuccessUrl: 'index.html',
3    signInOptions: [
4      firebase.auth.GoogleAuthProvider.PROVIDER_ID,
5    ],
6  };
7
8  // Initialize the FirebaseUI Widget using Firebase.
9  var ui = new firebaseui.auth.AuthUI(firebase.auth());
10 // The start method will wait until the DOM is loaded.
11 ui.start('#firebaseui-auth-container', uiConfig);

```

```

1  let search_tags = []
2  var recipes = [];
3
4  function update_tags() {}
5
6  async function load_recipes() {
7    let object = {};
8    await db.collection("recipes").orderBy("name").get().then((q) => {
9      localStorage.setItem("recipes", "");
10     q.forEach((document) => {
11       object = document.data();
12       object.id = document.id;
13       recipes.push(object);
14     })
15   });
16   if (recipes.length) {
17     localStorage.setItem("recipes", JSON.stringify(recipes));
18   }
19 }

```

```

20
21 function render_recipes(items) {
22   let div = document.getElementById("items")
23   div.innerHTML = "";
24   for(let i = 0; i<items.length; i++) {
25     let outer_div = document.createElement("DIV");
26     outer_div.setAttribute("class", "card card-sm col-lg-8 col-md-10 col-11");
27     let inner_div = document.createElement("DIV");
28     inner_div.setAttribute("class", "card-body");
29     inner_div.setAttribute("id", items[i].id)
30     let h2 = document.createElement("H2");
31     h2.appendChild(document.createTextNode(items[i].name));
32     let p = document.createElement("P");
33     inner_div.appendChild(h2);
34     if(items[i].tags.length !== 0) {
35       let tags = items[i].tags.join(", ")
36       p.appendChild(document.createTextNode('Tags: ${tags}'));
37       inner_div.appendChild(p)
38     }
39     let a = document.createElement("A");
40     a.setAttribute("href", `recipe.html?id=${items[i].id}`);
41     a.setAttribute("class", "btn btn-primary stretched-link")
42     a.setAttribute("onclick", `see_recipe(${i})`)
43     a.appendChild(document.createTextNode("See Recipe"))
44     inner_div.appendChild(a);
45     outer_div.appendChild(inner_div);
46     div.appendChild(outer_div);
47   }
48 }

```

read-database.js

```

1   let search_tags = []
2   var recipes = [];
3
4   function update_tags() {}
5
6   async function load_recipes() {
7     let object = {};
8     await db.collection("recipes").orderBy("name").get().then((q) => {
9       localStorage.setItem("recipes", "");
10      q.forEach((document) => {
11        object = document.data();
12        object.id = document.id;
13        recipes.push(object);
14      })
15    });
16    if (recipes.length) {
17      localStorage.setItem("recipes", JSON.stringify(recipes));
18    }
19  }
20
21 function render_recipes(items) {
22   let div = document.getElementById("items")
23   div.innerHTML = "";
24   for(let i = 0; i<items.length; i++) {

```

```

25     let outer_div = document.createElement("DIV");
26     outer_div.setAttribute("class", "card card-sm col-lg-8 col-md-10 col-11");
27     let inner_div = document.createElement("DIV");
28     inner_div.setAttribute("class", "card-body");
29     inner_div.setAttribute("id", items[i].id);
30     let h2 = document.createElement("H2");
31     h2.appendChild(document.createTextNode(items[i].name));
32     let p = document.createElement("P");
33     inner_div.appendChild(h2);
34     if(items[i].tags.length !== 0) {
35         let tags = items[i].tags.join(", ");
36         p.appendChild(document.createTextNode('Tags: ${tags}'));
37         inner_div.appendChild(p);
38     }
39     let a = document.createElement("A");
40     a.setAttribute("href", `recipe.html?id=${items[i].id}`);
41     a.setAttribute("class", "btn btn-primary stretched-link");
42     a.setAttribute("onclick", `see_recipe(${i})`);
43     a.appendChild(document.createTextNode("See Recipe"));
44     inner_div.appendChild(a);
45     outer_div.appendChild(inner_div);
46     div.appendChild(outer_div);
47 }
48 }

```

script.js

```

1  const tags = [
2      "Seafood", "Vegetables", "Pork", "Chicken", "Beef", "Soup", "Eggs", "Dessert"
3  ]
4
5  var logged_in = true;
6
7  function clearDiv(div) {
8      document.getElementById(div).innerHTML = "";
9  }
10
11  async function index_onload() {
12      let div = document.getElementById("checkboxboxes");
13      let search_field = document.getElementById("search-field");
14      search_field.value = "";
15      for (let i = 0; i < tags.length; i++) {
16          let checkbox = document.createElement("input");
17          checkbox.type = "checkbox";
18          checkbox.name = "option-selector";
19          checkbox.value = tags[i];
20          checkbox.setAttribute("id", `${checkbox.name}-${tags[i]}`);
21          checkbox.setAttribute("class", 'form-check-input custom-control-input');
22
23          let label = document.createElement("label");
24          label.setAttribute("for", `${checkbox.name}-${tags[i]}`);
25          label.setAttribute("class", 'form-check-label custom-control-label');
26          let t = document.createTextNode(`${tags[i]}`);
27          label.appendChild(t);
28
29          let d = document.createElement("DIV");

```

```

30         d.setAttribute("class", "custom-control custom-checkbox");
31
32         let br = document.createElement("BR");
33
34         d.appendChild(checkbox);
35         d.appendChild(label);
36         d.appendChild(br);
37         div.appendChild(d);
38     }
39     await load_recipes();
40     await add_search_listener();
41     render_recipes(recipes);
42 }
43
44 async function render_recipe() {
45     let URLParams = new URLSearchParams(window.location.search);
46     let recipes;
47     id = URLParams.get("id")
48     if(id === null) {
49         location.replace("index.html");
50     }
51     if (localStorage.getItem("recipes") === null) {
52         await load_recipes();
53     }
54
55     recipes = JSON.parse(localStorage.getItem("recipes"));
56
57     let index = recipes.findIndex((elem) => {
58         return elem.id === id;
59     });
60
61     if (index === -1) {
62         location.replace("index.html");
63     }
64
65     let recipe = recipes[index];
66
67     document.title = recipe.name;
68     render_title(recipe);
69     render_ingredients(recipe);
70     render_method(recipe);
71 }
72
73 function create_edit_button() {
74     let URLParams = new URLSearchParams(window.location.search);
75     let id = URLParams.get("id")
76     let div = document.getElementById("edit-button");
77     let a = document.createElement("a");
78     a.setAttribute("class", "btn btn-primary btn-block col-lg-8 col-11 btn-sm");
79     a.appendChild(document.createTextNode("Edit Recipe"));
80     a.setAttribute("onclick", `edit_recipe("${id}")`);
81     div.appendChild(a);
82 }
83
84 function edit_recipe(id) {
85     location.assign(`new-recipe.html?id=${id}`);
86 }
87
88 function render_title(recipe) {

```

```

89     let title = document.getElementById("title-box");
90     let title_container = document.createElement("div");
91     title_container.setAttribute("class", "col-11 col-md-10 col-lg-8 card card-sm");
92     title.appendChild(title_container);
93     let header_container = document.createElement("div");
94     header_container.setAttribute("class", "card-body row no-gutters");
95     let header_h2 = document.createElement("H2");
96     header_h2.appendChild(document.createTextNode(recipe.name));
97     header_container.appendChild(header_h2);
98     title_container.appendChild(header_container);
99     console.log(recipe.tags.length)
100    if (recipe.tags.length !== 0) {
101        let tag_container = document.createElement("div");
102        tag_container.setAttribute("class", "col-lg-12 card-body");
103        let tags = recipe.tags.join(", ");
104        tag_container.appendChild(document.createTextNode('Tags: ${tags}'));
105        title_container.appendChild(tag_container);
106        title.appendChild(title_container);
107    }
108 }
109
110 function render_ingredients(recipe) {
111     document.getElementById("list").innerHTML = "";
112     document.getElementById("servings").innerHTML = "";
113     let servings = document.getElementById("servings");
114     let h5 = document.createElement("h5");
115     h5.appendChild(document.createTextNode("Serves:"))
116     servings.appendChild(h5);
117     let servings_input = document.createElement("input");
118     servings_input.setAttribute("type", "text");
119     servings_input.setAttribute("class", "form-control");
120     servings_input.setAttribute("style", 'width: 20%; margin-left: 1em;');
121     servings_input.setAttribute("value", recipe.servings);
122     servings_input.setAttribute("oninput", "this.value = this.value.replace(/[~0-9.]/,
123     '');"");
124     servings_input.setAttribute("onfocusout", "change_servings(this)");
125     servings_input.setAttribute("id", "servings_input_box")
126     servings.appendChild(servings_input);
127
128     let i_list = document.getElementById("list");
129     let div;
130     let li;
131     let span;
132     let input;
133     let t;
134     let small;
135     for (let i = 0; i < recipe["ingredients-name"].length; i++) {
136         div = document.createElement("div");
137         div.setAttribute("class", "row col-lg-12");
138         li = document.createElement("li");
139         span = document.createElement("span");
140         if (recipe["ingredients-quantity"][i]) {
141             if(recipe["ingredients-unit"][i]) {
142                 t = document.createTextNode(`${recipe["ingredients-name"][i]} `)
143                 small = document.createElement("small");
144                 small.appendChild(document.createTextNode(`${recipe["ingredients-unit"]
145                 "[i]} `))
146
147                 span = document.createElement("span");
148                 input = document.createElement("input")

```

```

146         input.setAttribute("type", "text");
147         input.setAttribute("class", "form-control");
148         input.setAttribute("style", "width: 5em; float:left; margin-right: 1
em");
149         input.setAttribute("oninput","this.value = this.value.replace
(/>
150         (/^[^0-9.]/g, ''"));
151         input.setAttribute("value", `${recipe["ingredients-quantity"][i]}');
152         input.setAttribute("name", recipe["ingredients-name"][i]);
153         input.setAttribute("onfocusout", "change_ingredient(this)");
154         // input.setAttribute("onfocusout")
155         span.appendChild(input);
156         li.appendChild(span).appendChild(small)
157         li.appendChild(t);
158     } else {
159         t = document.createTextNode(`${recipe["ingredients-name"][i]}')
160         span = document.createElement("span");
161         input = document.createElement("input")
162         input.setAttribute("type", "text");
163         input.setAttribute("class", "form-control");
164         input.setAttribute("style", "width: 5em; float:left; margin-right: 1
em");
165         input.setAttribute("value", `${recipe["ingredients-quantity"][i]}');
166         input.setAttribute("oninput","this.value = this.value.replace
(/>
167         (/^[^0-9.]/g, ''"));
168         input.setAttribute("name", recipe["ingredients-name"][i]);
169         input.setAttribute("onfocusout", "change_ingredient(this)");
170         span.appendChild(input);
171         li.appendChild(span).appendChild(t);
172     }
173     } else {
174         t = document.createTextNode(`${recipe["ingredients-name"][i]}')
175         span = document.createElement("span");
176         input = document.createElement("input");
177         input.setAttribute("type", "text");
178         input.setAttribute("class", "form-control");
179         input.setAttribute("style", "width: 5em; float:left; margin-right: 1em");
180         input.setAttribute("disabled","true");
181         input.setAttribute("onfocusout","this.value = this.value.replace
(/>
182         (/^[^0-9.]/g, '').replace(/(\.\.*)"\/g, '$1')");
183         input.setAttribute("name", recipe["ingredients-name"][i])
184         span.appendChild(input);
185         li.appendChild(span).appendChild(t);
186     }
187     div.appendChild(li);
188     i_list.appendChild(div);
189 }
190
191 function render_method(recipe) {
192     let ol = document.getElementById("method-list");
193     let li;
194     for(let i = 0; i<recipe["method"].length; i++) {
195         li = document.createElement("LI");
196         li.appendChild(document.createTextNode(`${recipe["method"][i]}'));
197         ol.appendChild(li);
198     }
199 }

```

```

1  const fuse_options = {
2      includeScore: true,
3      keys: ['name']
4  }
5
6  function add_search_listener() {
7      let listener = ["search-field", "search-btn"]
8      let module = document.getElementById(listener[0]);
9      module.addEventListener('keypress', function (event) {
10         if (event.code === "Enter") {
11             event.preventDefault(); // Cancel the default action
12             document.getElementById(listener[1]).click(); // Click button
13         }
14     });
15 }
16
17 async function search() {
18     let search_bar = document.getElementById("search-field");
19     let search_value = search_bar.value;
20     if (!localStorage.getItem("recipes")) {
21         await load_recipes();
22     }
23     let recipes = JSON.parse(localStorage.getItem("recipes"));
24
25     let fuse = new Fuse(recipes, fuse_options);
26     let result = fuse.search(search_value);
27     console.log(result);
28     let options = read_search_options();
29
30     let weighted_result = result.filter((e) => {
31         return e.score < 0.6
32     }).map((e) => {
33         return e.item
34     });
35
36     let final_result = [];
37
38     if (options.length) {
39         for (let i = 0; i < weighted_result.length; i++) {
40             let result_tags = weighted_result.map((e) => {
41                 return e.tags
42             })
43             console.log(result_tags);
44             let flag = true;
45             for (let j = 0; j < options.length; j++) {
46                 if (!weighted_result[i].tags.includes(options[j])) {
47                     flag = false;
48                 }
49             }
50             if (flag) {
51                 final_result.push(weighted_result[i])
52             }
53         }
54     } else {
55         final_result = weighted_result
56     }
57
58     let items = document.getElementById("items");
59     items.innerHTML = "";

```

```

60     if (!search_value) {
61         render_recipes(recipes);
62         return;
63     }
64
65     if (final_result.length) {
66         render_recipes(final_result);
67         return;
68     } else {
69         render_no_results();
70     }
71
72 }
73
74 function read_search_options() {
75     let options = [];
76     let checkboxes = document.getElementsByName("option-selector");
77     for (let i = 0; i < checkboxes.length; i++) {
78         if (checkboxes[i].checked) {
79             options.push(checkboxes[i].value);
80         }
81     }
82     return options;
83 }
84
85 function render_no_results() {
86     let items = document.getElementById("items");
87     let h5 = document.createElement("h5");
88     h5.appendChild(document.createTextNode('No results were found. Reset?'))
89     let a = document.createElement("a");
90     a.appendChild(h5);
91     a.setAttribute("href", "index.html");
92     items.appendChild(a);
93 }

```

styles.css

```

1  @import url('https://fonts.googleapis.com/css2?family=Open+Sans&family=Proza+Libre&
2  display=swap');
3
4  /* Padding for the navbar only */
5  .navbar-padding {
6      padding-top: 2em;
7  }
8
9  /* Normal Padding for any element */
10 .padding {
11     padding-top: 1em;
12 }
13
14 /* Space between every card under recipes */
15 div#items > div {
16     padding-top: 1em;
17     padding-bottom: 1em;
18     margin: 1em;
19 }

```



```

19
20  /* Disable scrolling */
21  html, body {
22      max-width: 100%;
23      overflow-x: hidden;
24  }
25
26  /* Disable scrolling (mobile) */
27  body {
28      position: relative;
29  }
30
31  .checkbox-spacing {
32      margin-right: 0.5rem;
33  }
34
35  .label-spacing > p {
36      margin-right: 2rem;
37  }
38
39  .bot-spacing {
40      margin-bottom: 5em;
41  }
42
43  .warning {
44      border: 1px solid red;
45  }
46
47  .right-gap {
48      margin-right: 1em;
49  }
50
51  .col {
52      flex: 1;
53  }
54
55  /* Remove arrows from number fields */
56
57  /* Chrome, Safari, Edge, Opera */
58  input::-webkit-outer-spin-button,
59  input::-webkit-inner-spin-button {
60      -webkit-appearance: none;
61      margin: 0;
62  }
63
64  /* Firefox */
65  input[type=number] {
66      -moz-appearance: textfield;
67  }

```