



# MACRO MEALS TECHNICAL MANUAL 2023



**SE  
TU**

Ollscoil  
Teicneolaíochta  
an Oirdheiscirt

South East  
Technological  
University

**SUPERVISOR**

Chris Meudec

**STUDENT**

Mohsin Tahir (C00250220)

**DATE**

17-04-2023

## Table of Contents

<b>Introduction</b>	<b>2</b>
<b>Source Code</b>	<b>3</b>
Python Code	3
App.py	3
HTML Code	29
Base.html	29
Changepw.html	31
Edamam.html	36
Gpt.html	73
Index.html	76
Login.html	77
Nav.html	82
Register.html	83
Results.html	88
savedRecipes.html	103
CSS Code	110
Changepw.css	110
Edamam.css	111
Gpt.css	112
Index.css	113
Login.css	114
Register.css	114
Results.css	115
savedRecipes.css	119

## Introduction

The goal of this document is to provide all the source code used during development. It will include the main app.py file which holds all the routing and all the HTML and CSS pages with all the content and styling.

All the code is also available on my GitHub page

<https://github.com/Mohsin272/Y4-Final-Project>

# Source Code

## Python Code

### App.py

```
# Author - Mohsin Tahir
# Date - 17/04/23

from flask import Flask, request, render_template, redirect, session
import requests

import secrets

import DBcm

import bcrypt

from appconfig import config

import re

import openai

import secret

salt = bcrypt.gensalt()

app = Flask(__name__)

app.secret_key = secrets.token_hex(32)

@app.route("/")

def index():
```

```

    return render_template("index.html", title="Welcome")

@app.route("/register")
def register():
    return render_template("register.html", title="Welcome")

@app.route("/deleteRecipe", methods=["GET", "POST"])
def deleteRecipe():
    username = session.get("username", None)
    url = request.form.get("Link")
    if username is not None:
        with DBcm.UseDatabase(config) as db:
            SQL = """delete from saved_recipes where Username = %s and
Link = %s"""
            data = (username, url)
            db.execute(SQL, data)
            return redirect("/savedRecipes")
    else:
        return redirect("/login")

```

```

@app.route("/savedRecipes")
def savedRecipes():
    username = session.get("username", None)

    if username is not None:

        with DBcm.UseDatabase(config) as db:

            SQL = """select * from saved_recipes where Username = %s"""

            db.execute(SQL, (username,))

            res = db.fetchall()

            return render_template("savedRecipes.html", title="Saved
Recipe", res=res,)

        else:

            return redirect("/login")

@app.route("/logout")
def logout():

    session.pop("username", None)

    return redirect("/login")

@app.route("/passreset", methods=["GET", "POST"])
def changepw():

    return render_template("changepw.html", title="Change Password")

```

```

@app.route("/processreset", methods=["GET", "POST"])
def processreset():
    errors = []

    email = request.form.get("email")

    password = request.form.get("password")

    repeatpassword = request.form.get("repeatpassword")

    password = password.encode("utf-8")

    repeatpassword = repeatpassword.encode("utf-8")

    hashed_password = bcrypt.hashpw(password, salt)

    hashed_password_r = bcrypt.hashpw(repeatpassword, salt)

    if hashed_password != hashed_password_r:
        errors.append("Passwords do not match")

    else:

        with DBcm.UseDatabase(config) as db:

            SQL = """

                UPDATE users

                SET Password = %s

                WHERE Email = %s

            """

            db.execute(SQL, (hashed_password, email))

        return redirect("/login")

```

```

        return render_template("changePW.html", title="Change Password",
errors=errors)

@app.route("/login", methods=["GET", "POST"])
def login():

    if "username" in session:

        return redirect("/recipe")

    errors = ""

    if request.method == "POST":

        username = request.form.get("username")

        Userpassword = request.form.get("password")

        with DBcm.UseDatabase(config) as db:

            SQL = """select * from users where Username = %s"""

            db.execute(SQL, (username,))

            res = db.fetchall()

            if res:

                Userpassword = Userpassword.encode("utf-8")

                hashedDB = res[0][3].encode("utf-8")

                passres = bcrypt.checkpw(Userpassword, hashedDB)

                if res and passres:

                    session["username"] = username

                    session["email"] = res[0][2]

```

```

        return redirect("/recipe")

    else:

        errors = "Username/Password are incorrect"

    else:

        errors = "Username/Password are incorrect"

    return render_template("login.html", title="Welcome",
errors=errors)

@app.route("/gpt", methods=["POST", "GET"])
def gpt():

    openai.api_key = secret.open_ai_api_key

    ing = request.form.get("Ingredients").strip("[]")

    message = "print this list of ingredients with carbon footprint
friendly alternatives. Only print a list no explanation please."

    separator = "\n"

    ingredient_list = ing.split(", ")

    joined_ingredients = separator.join(ingredient_list)

    message_with_ingredients = message + "\n\n" + joined_ingredients

    response = openai.ChatCompletion.create(

        model="gpt-3.5-turbo",

        messages=[{"role": "user", "content":
message_with_ingredients}],

    )

```



```

result = ""

for choice in response.choices:

    result = choice.message.content

output = re.split(r"(?<=\D)\s*-\s+", result)

output = [s.strip() for s in output if s.strip()]

return render_template("gpt.html", title="GPT Results",
output=output)

@app.route("/addrecipe", methods=["GET", "POST"])
def addrecipe():

    Ingredients = request.form.get("Ingredients").strip("[]")

    Calories = request.form.get("Calories")

    Servings = request.form.get("Servings")

    Carbs_value = request.form.get("Carbs value")

    Fat_value = request.form.get("Fat value")

    Protein_value = request.form.get("Protein value")

    Sugars_value = request.form.get("Sugars value")

    Fiber_value = request.form.get("Fiber value")

    Link = request.form.get("Link")

    Label = request.form.get("Label")

    Username = session.get("username", None)

    Email = session.get("email", None)

```

```

Carbon = request.form.get("carbon")

Cost = request.form.get("cost")

if Username and Email is not None:

    with DBcm.UseDatabase(config) as db:

        SQL = """SELECT * FROM saved_recipes WHERE Email = %s AND
Link = %s"""

        db.execute(SQL, (Email, Link))

        existing_recipe = db.fetchone()

    if existing_recipe:

        return redirect("/savedRecipes")

    with DBcm.UseDatabase(config) as db:

        SQL = """

            insert into saved_recipes

            (Username, Email, Ingredients, Calories, Servings,

            Carbs_value,

            Fat_value,

            Protein_value,

            Sugars_value,

            Fiber_value,

            Link, Label, Carbon, Cost

            )

```

```
        values
            (%s, %s, %s, %s, %s,
            %s, %s, %s, %s, %s,
            %s, %s, %s, %s
            )
        """
db.execute(
    SQL,
    (
        Username,
        Email,
        Ingredients,
        Calories,
        Servings,
        Carbs_value,
        Fat_value,
        Protein_value,
        Sugars_value,
        Fiber_value,
        Link,
        Label,
        Carbon,
        Cost,
```

```

        ),
    )

    return redirect("/savedRecipes")

else:
    return redirect("/login")

def check_email_exists(email):
    with DBcm.UseDatabase(config) as d:
        SQL = """select * from users where Email = %s"""

        d.execute(SQL, (email,))

        rows = d.fetchall()

    if len(rows) > 0:
        return True

    else:
        return False

@app.route("/processform", methods=["GET", "POST"])
def processform():
    errors = []

    username = request.form.get("username")

    email = request.form.get("email")

```

```

password = request.form.get("password")

repeatpassword = request.form.get("repeatpassword")

password = password.encode("utf-8")

repeatpassword = repeatpassword.encode("utf-8")

hashed_password = bcrypt.hashpw(password, salt)

hashed_password_r = bcrypt.hashpw(repeatpassword, salt)

if hashed_password != hashed_password_r:

    errors.append("Passwords do not match")

if check_email_exists(email) == True:

    errors.append("Email already registered")

else:

    with DBcm.UseDatabase(config) as db:

        SQL = """

            insert into users

            (Username, Email, Password)

            values

            (%s, %s, %s)

        """

        db.execute(SQL, (username, email, hashed_password))

    return redirect("/login")

    return render_template("register.html", title="Welcome",
errors=errors)

```

```

@app.route("/recipe")

def edamam():

    return render_template("edamam.html", title="Macro Meals",)

def sort_by_criteria(res, criteria):

    if criteria == "caloriesLH":

        return sorted(res, key=lambda r: int(r["recipe"]["calories"]) /
int(r["recipe"]["yield"]))

    elif criteria == "caloriesHL":

        return sorted(res, key=lambda r: int(r["recipe"]["calories"]) /
int(r["recipe"]["yield"]), reverse=True)

    elif criteria == "servingLH":

        return sorted(res, key=lambda r: r["recipe"]["yield"])

    elif criteria == "servingHL":

        return sorted(res, key=lambda r: r["recipe"]["yield"],
reverse=True)

    elif criteria == "carbsLH":

        return sorted(

            res, key=lambda r:
int(r["recipe"]["totalNutrients"]["CHOCDF.net"]["quantity"]) /
int(r["recipe"]["yield"])

        )

```

```

elif criteria == "carbsHL":

    return sorted(

        res,

        key=lambda r:
int(r["recipe"]["totalNutrients"]["CHOCDF.net"]["quantity"]) /
int(r["recipe"]["yield"]),

        reverse=True,

    )

elif criteria == "fatLH":

    return sorted(

        res, key=lambda r:
int(r["recipe"]["totalNutrients"]["FAT"]["quantity"]) /
int(r["recipe"]["yield"])

    )

elif criteria == "fatHL":

    return sorted(

        res,

        key=lambda r:
int(r["recipe"]["totalNutrients"]["FAT"]["quantity"]) /
int(r["recipe"]["yield"]),

        reverse=True,

    )

elif criteria == "proteinLH":

    return sorted(

```

```

        res, key=lambda
r:int(r["recipe"]["totalNutrients"]["PROCNT"]["quantity"]) /
int(r["recipe"]["yield"])
    )

    elif criteria == "proteinHL":

        return sorted(

            res,

            key=lambda r:
int(r["recipe"]["totalNutrients"]["PROCNT"]["quantity"]) /
int(r["recipe"]["yield"]),

            reverse=True,

        )

    elif criteria == "sugarLH":

        return sorted(

            res, key=lambda r:
int(r["recipe"]["totalNutrients"]["SUGAR"]["quantity"]) /
int(r["recipe"]["yield"])

        )

    elif criteria == "sugarHL":

        return sorted(

            res,

            key=lambda r:
int(r["recipe"]["totalNutrients"]["SUGAR"]["quantity"]) /
int(r["recipe"]["yield"]),

            reverse=True,

        )

```



```

elif criteria == "fiberLH":

    return sorted(

        res, key=lambda r:
int(r["recipe"]["totalNutrients"]["FIBTG"]["quantity"]) /
int(r["recipe"]["yield"])

    )

elif criteria == "fiberHL":

    return sorted(

        res,

        key=lambda r:
int(r["recipe"]["totalNutrients"]["FIBTG"]["quantity"]) /
int(r["recipe"]["yield"]),

        reverse=True,

    )

def process_ingredients(res, items):

    ingredients_list = []

    res_list = []

    for row in res:

        for ing in row["recipe"]["ingredientLines"]:

            ingredients_list.append(ing)

        value = get_carbon_value(ingredients_list, items)

        res_list.append(value)

```

```

    return res_list

@app.route("/sort_results", methods=["POST", "GET"])
def sort_results():
    app_id = secret.edamam_app_id
    app_key = secret.edamam_api_key
    meal_type = request.form.get("meal")
    ingredients = request.form.get("ing")
    health = request.form.get("health")

    if health == "" and meal_type == "":
        result = requests.get(
            f"https://api.edamam.com/search?q={ingredients}&app_id={app_id}&app_key={app_key}"
        )

    elif not meal_type:
        result = requests.get(
            f"https://api.edamam.com/search?q={ingredients}&health={health}&app_id={app_id}&app_key={app_key}"
        )

    elif health == "" or " ":
        result = requests.get(

```

```

f"https://api.edamam.com/search?q={ingredients}&mealType={meal_type}&app
p_id={app_id}&app_key={app_key}"

    )

    else:

        result = requests.get(

f"https://api.edamam.com/search?q={ingredients}&health={health}&mealTyp
e={meal_type}&app_id={app_id}&app_key={app_key}"

        )

    data = result.json()

    res = data["hits"]

    criteria = request.form["criteria"]

    res = sort_by_criteria(res, criteria)

    res_list = []

    res_list = process_ingredients(res, items, items_carbon_values)

    cost_list = process_costs(res, expensive, expensive_cost)

    return render_template(

        "results.html",

        title="Suggested Recipes ",

        data=res,

        cfv=res_list,

```

```

        meal=meal_type,

        ing=ingredients,

        health=health,

        cost=cost_list,

    )

@app.route("/recipe", methods=["GET", "POST"])
def process():

    app_id = secret.edamam_app_id

    app_key = secret.edamam_api_key

    meal_type = request.form.getlist("mealcheck")

    meal_type = ",".join(meal_type)

    ingredients = request.form["ingredients"]

    health = request.form.getlist("dietcheck")

    health = ",".join(health)

    selectedIng = request.form.getlist("check")

    selectedIng = ",".join(selectedIng)

    if ingredients == "":

        ingredients = selectedIng

    else:

        ingredients = selectedIng + "," + ingredients

    if health == "" and meal_type == "":

```

```

        result = requests.get(

f"https://api.edamam.com/search?q={ingredients}&app_id={app_id}&app_key
={app_key}"

        )

        elif not meal_type:

            result = requests.get(

f"https://api.edamam.com/search?q={ingredients}&health={health}&app_id=
{app_id}&app_key={app_key}"

            )

            elif health == "" or " ":

                result = requests.get(

f"https://api.edamam.com/search?q={ingredients}&mealType={meal_type}&ap
p_id={app_id}&app_key={app_key}"

                )

            else:

                result = requests.get(

f"https://api.edamam.com/search?q={ingredients}&health={health}&mealTyp
e={meal_type}&app_id={app_id}&app_key={app_key}"

                )

            data = result.json()

            res = data["hits"]

            res_list = []

```

```

res_list = process_ingredients(res, items, items_carbon_values)
cost_list = process_costs(res, expensive, expensive_cost)

accessible_recipes = []

for row in res:
    url = row["recipe"]["url"]
    status_code = check_status(url)
    if status_code != 404:
        accessible_recipes.append(row)

res = accessible_recipes

return render_template(
    "results.html",
    title="Suggested Recipes ",
    data=res,
    cfv=res_list,
    meal=meal_type,
    ing=ingredients,
    health=health,
    cost=cost_list,
)

```

```

def extract_quantity(ingredient):

    quantity_match = re.search(r"(\d+([\.,]\d+)?)", ingredient)

    if quantity_match:

        return float(quantity_match.group(1))

    else:

        return 1

def get_carbon_value(ingredients_list, items, items_carbon_values,
serving_size):

    total_carbon_value = 0

    for ing in ingredients_list:

        for item, carbon_value in zip(items, items_carbon_values):

            if item.lower() in ing.lower():

                quantity = extract_quantity(ing)

                total_carbon_value += carbon_value * quantity

                break

    if serving_size >= 4:

        total_carbon_value /= serving_size

    if total_carbon_value >= 30:

        return "RED"

    else:

        return "GREEN"

```

```

def process_ingredients(res, items, items_carbon_values):

    res_list = []

    for row in res:

        ingredients_list = []

        for ing in row["recipe"]["ingredientLines"]:

            ingredients_list.append(ing)

        serving_size = row["recipe"]["yield"]

        value = get_carbon_value(

            ingredients_list, items, items_carbon_values, serving_size

        )

        res_list.append(value)

    return res_list

def get_cost_value(ingredients_list, expensive, expensive_cost,
serving_size):

    total_cost_value = 0

    for ing in ingredients_list:

        for item, price in zip(expensive, expensive_cost):

            if item.lower() in ing.lower():

                quantity = extract_quantity(ing)

```



```

        total_cost_value += price * quantity

        break

    if serving_size >= 4:

        total_cost_value /= serving_size

    if total_cost_value >= 50:

        return "High Cost"

    elif total_cost_value >= 25:

        return "Average Cost"

    else:

        return "Low Cost"

def process_costs(res, expensive, expensive_cost):

    res_list = []

    for row in res:

        ingredients_list = []

        for ing in row["recipe"]["ingredientLines"]:

            ingredients_list.append(ing)

        serving_size = row["recipe"]["yield"]

        value = get_cost_value(

            ingredients_list, expensive, expensive_cost, serving_size

```

```
)  
    res_list.append(value)  
  
return res_list  
  
def check_status(url):  
    try:  
        response = requests.head(url)  
        return response.status_code  
    except requests.RequestException:  
        return None  
  
items = [  
    "Avocado",  
    "Beef",  
    "Cheese",  
    "Chicken",  
    "Chocolate",  
    "Coffee",  
    "Corn",  
    "Lamb",  
    "Mutton",
```

```

    "Palm Oil",

    "Pork",

    "Rapeseed Oil",

    "Shrimp",

    "Soy milk",

    "Soybean Oil",

    "Soybeans",

    "Sunflower Oil",

    "Tofu",

    "Steak",

    "Prawn",

]

# Carbon values an items taken from
https://ourworldindata.org/food-choice-vs-eating-local

items_carbon_values = [

    2, # Avocado

    60, # Beef

    21, # Cheese

    6, # Chicken

    19, # Chocolate

    17, # Coffee

    1, # Corn

    24, # Lamb

```

```

24, # Mutton

8, # Palm Oil

7, # Pork

4, # Rapeseed Oil

12, # Shrimp

3, # Soy milk

6.2, # Soybean Oil

1, # Soybeans

3.5, # Sunflower Oil

2, # Tofu

60, # Steak

12, # Prawn

]

# Cost items taken from
https://www.bbc.co.uk/programmes/articles/1LX0tn2vrN5Ls7RXcfb47rz/eight-of-the-world-s-most-expensive-foods

expensive = ["Wagyu", "Lobster", "Truffle", "Manuka Honey", "Saffron",
"Caviar"]

expensive_cost = [

300, # Wagyu (per kg)

50, # Lobster (per kg)

4000, # Truffle (per kg)

62, # Manuka Honey (per kg)

3990, # Saffron (per kg)

```

```
    2100, # Caviar (per kg)
]

if __name__ == "__main__": # pragma no cover
    app.run(debug=True)
```

## HTML Code

### Base.html

```
<!-- Author - Mohsin Tahir
Date - 17/04/23 -->
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.m
in.css" rel="stylesheet"
```

```

integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoq
yl2QvZ6jIW3" crossorigin="anonymous">

    <link rel="shortcut icon" href="{{ url_for('static',
filename='favicon.ico') }}">

    <!--Favicon Logo made using https://www.logoai.com/ -->

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <link
href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">

    <title>{{title}}</title>

</head>

<body>

    <br>

    <div class="text-center">

        <h2>{{heading}}</h2>

    </div>

    <br>

    <div class="container">

        {% block body %}

        {% endblock %}

```

```

</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
integrity="sha384-KJ3o2DKtIkvYIK3UEENzmM7KcKrR/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>

<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

</body>
</html>

```

## Changepw.html

```

<!-- Author - Mohsin Tahir
Date - 17/04/23 -->

{% extends "base.html" %}

```

```

{% block body %}

<link rel="stylesheet" href="{{ url_for('static',
filename='changepw.css') }}">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">

  <a class="navbar-brand">Macro Meals</a>

  <button

    class="navbar-toggler"

    type="button"

    data-toggle="collapse"

    data-target="#navbarSupportedContent"

    aria-controls="navbarSupportedContent"

    aria-expanded="false"

    aria-label="Toggle navigation"

  >

    <span class="navbar-toggler-icon"></span>

  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">

    <ul class="navbar-nav mr-auto">

      <li class="nav-item">

        <a class="nav-link" href="/recipe">Search Recipes</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="/savedRecipes">Saved Recipes</a>

```



```

        </li>
    </ul>
</div>
</nav>
<br><br><br>
<div class="d-flex justify-content-center">
    <div class="card" style="width: 30rem;">
        <div class="card-body">
            <form action="/processreset" method="post" class
="needs-validation" novalidate>
                <p class="h4 text-center">
                    Reset Password
                </p>
                <div class="mb-3">
                    <label for="emailID" class="form-label">Email address</label>
                    <input type="email" class="form-control" id="emailID"
aria-describedby="emailHelp" name="email" required>
                    <div class="invalid-feedback">
                        Please enter a valid email address.
                    </div>
                </div>
            </div>
            <div class="mb-3">
                <label for="passwordID" class="form-label">New Password</label>

```

```

    <input type="password" class="form-control" id="passwordID"
name="password" required minlength="8">

    <div class="invalid-feedback">

        Please enter a password.

    </div>

</div>

<div class="mb-3">

    <label for="passwordrepeatID" class="form-label">Repeat
Password</label>

    <input type="password" class="form-control" id="passwordrepeatID"
name="repeatpassword" required minlength="8">

    <div class="invalid-feedback">

        Please re-enter your password.

    </div>

</div>

<div class="text-center">

    {% if errors %}

    <div class="alert alert-danger" role="alert">

        {% for e in errors %}

            {{e}} <br>

        {% endfor %}

    </div>

    {% endif %}

```

```

        <button type="submit" class="btn btn-success btn-with-icon
btn-block mt-3">

        <span class="material-icons">lock_open</span>

        Change

    </button>

    <br>

    <a href="/login" class="btn btn-secondary btn-with-icon btn-block
mt-3">

        <span class="material-icons">login</span>

        Login

    </a>

</div>

</form>

</div>

</div>

</div>

<script>

(function () {

    'use strict'

    var forms = document.querySelectorAll('.needs-validation')

    Array.prototype.slice.call(forms)

        .forEach(function (form) {

```

```

    form.addEventListener('submit', function (event) {

        if (!form.checkValidity()) {

            event.preventDefault()

            event.stopPropagation()

        }

        form.classList.add('was-validated')

    }, false)

})

}) ()

</script>

{% endblock %}

```

## Edamam.html

```

<!-- Author - Mohsin Tahir

Date - 17/04/23 -->

{% extends "base.html" %} {% block body %}

<link rel="stylesheet" href="{{ url_for('static',
filename='edamam.css') }}">

<script>

    document.addEventListener('DOMContentLoaded', function () {

        let checkboxes = document.querySelectorAll('.form-check-input');

```

```

let selectedVeggiesDiv = document.getElementById('selected-items');

let clearSelectionsButton =
document.getElementById('clear-selections');

clearSelectionsButton.style.display = 'none';

loadSelectedOptions();

checkboxes.forEach(function (checkbox) {

    checkbox.addEventListener('change', function () {

        updateSelectedVeggies();

    });

});

clearSelectionsButton.addEventListener('click', function () {

    checkboxes.forEach(function (checkbox) {

        checkbox.checked = false;

    });

    updateSelectedVeggies();

});

function updateSelectedVeggies() {

    let selectedVeggies = [];

```

```

checkboxes.forEach(function (checkbox) {
    if (checkbox.checked) {
        selectedVeggies.push(checkbox.value);
    }
});

if (selectedVeggies.length > 0) {
    selectedVeggiesDiv.innerHTML = '<h5>Selected Filters:</h5>' +
selectedVeggies.map(function (veggie) {
        return '<span class="badge bg-secondary me-2">' + veggie +
'</span>';
    }).join('');
    clearSelectionsButton.style.display = 'inline-block';
} else {
    selectedVeggiesDiv.innerHTML = '';
    clearSelectionsButton.style.display = 'none';
}

saveSelectedOptions(selectedVeggies);
}

function saveSelectedOptions(selectedVeggies) {
    sessionStorage.setItem('selectedVeggies',
JSON.stringify(selectedVeggies));
}

```

```

}

function loadSelectedOptions() {
    let selectedVeggies =
JSON.parse(sessionStorage.getItem('selectedVeggies'));

    if (selectedVeggies && selectedVeggies.length > 0) {
        checkboxes.forEach(function (checkbox) {
            if (selectedVeggies.includes(checkbox.value)) {
                checkbox.checked = true;
            }
        });
        updateSelectedVeggies();
    }
});
</script>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
    <a class="navbar-brand">Macro Meals</a>
    <button
        class="navbar-toggler"
        type="button"
        data-toggle="collapse"
        data-target="#navbarSupportedContent"

```

```

aria-controls="navbarSupportedContent"

aria-expanded="false"

aria-label="Toggle navigation"

>

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

  <ul class="navbar-nav mr-auto">

    <li class="nav-item">

      <a class="nav-link" href="/recipe">Search Recipes</a>

    </li>

    <li class="nav-item">

      <a class="nav-link" href="/savedRecipes">Saved Recipes</a>

    </li>

    {% if "username" not in session %}

    <li class="nav-item">

      <a class="nav-link text-success" href="/logout">Login</a>

    </li>

    {% else %}

    <li class="nav-item">

      <a class="nav-link text-danger" href="/logout">Logout</a>

    </li>

    {% endif %}

```



```

    </ul>

  </div>

</nav>

<br /><br /><br />

<div class="text-center"><h1>Search Recipes</h1></div>

<div class="d-flex justify-content-center">

  <div class="text-left">

    <br />

    <form method="post">

      <label>Common Ingredients</label><br />

      <div class="btn-group" role="group">

        <button

          id="btnGroupDrop1"

          type="button"

          class="btn btn-primary dropdown-toggle"

          data-bs-toggle="dropdown"

          aria-expanded="false"

        >

          Vegetables

        </button>

        <ul class="dropdown-menu" aria-labelledby="btnGroupDrop1">

          <div class="form-check">

            <input

```

```
        class="form-check-input"

        type="checkbox"

        value="potato"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Potatoes

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="onion"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Onions

    </label>

</div>

<div class="form-check">
```

```
<input
  class="form-check-input"
  type="checkbox"
  value="carrot"
  id="flexCheckDefault"
  name="check"
/>

<label class="form-check-label" for="flexCheckDefault">
  Carrots
</label>
</div>

<div class="form-check">
  <input
    class="form-check-input"
    type="checkbox"
    value="mushroom"
    id="flexCheckDefault"
    name="check"
  />

  <label class="form-check-label" for="flexCheckDefault">
    Mushrooms
  </label>
</div>
```

```
<div class="form-check">

  <input

    class="form-check-input"

    type="checkbox"

    value="broccoli"

    id="flexCheckDefault"

    name="check"

  />

  <label class="form-check-label" for="flexCheckDefault">

    Broccoli

  </label>

</div>

<div class="form-check">

  <input

    class="form-check-input"

    type="checkbox"

    value="peas"

    id="flexCheckDefault"

    name="check"

  />

  <label class="form-check-label" for="flexCheckDefault">

    Peas

  </label>

</div>
```

```
</div>

<div class="form-check">

  <input

    class="form-check-input"

    type="checkbox"

    value="cabbage"

    id="flexCheckDefault"

    name="check"

  />

  <label class="form-check-label" for="flexCheckDefault">

    Cabbage

  </label>

</div>

<div class="form-check">

  <input

    class="form-check-input"

    type="checkbox"

    value="lettuce"

    id="flexCheckDefault"

    name="check"

  />

  <label class="form-check-label" for="flexCheckDefault">

    Lettuce
```

```
        </label>

    </div>

    <div class="form-check">

        <input

            class="form-check-input"

            type="checkbox"

            value="peppers"

            id="flexCheckDefault"

            name="check"

        />

        <label class="form-check-label" for="flexCheckDefault">

            Peppers

        </label>

    </div>

</ul>

</div>

<div class="btn-group" role="group">

    <button

        id="btnGroupDrop1"

        type="button"

        class="btn btn-primary dropdown-toggle"

        data-bs-toggle="dropdown"

        aria-expanded="false"
```

```
>
  Fruits
</button>
<ul class="dropdown-menu" aria-labelledby="btnGroupDrop1">
  <div class="form-check">
    <input
      class="form-check-input"
      type="checkbox"
      value="tomato"
      id="flexCheckDefault"
      name="check"
    />
    <label class="form-check-label" for="flexCheckDefault">
      Tomato
    </label>
  </div>
  <div class="form-check">
    <input
      class="form-check-input"
      type="checkbox"
      value="apple"
      id="flexCheckDefault"
      name="check"
    >
```

```
    />

    <label class="form-check-label" for="flexCheckDefault">

        Apple

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="banana"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Banana

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="orange"

        id="flexCheckDefault"
```



```
        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Orange

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="pear"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Pear

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="avocado"
```

```
        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Avocado

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="grape"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Grapes

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"
```

```
        value="strawberry"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Strawberry

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="raspberry"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Raspberry

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"
```

```

        type="checkbox"

        value="cherry"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Cherry

    </label>

</div>

</ul>

</div>

<div class="btn-group" role="group">

    <button

        id="btnGroupDrop1"

        type="button"

        class="btn btn-primary dropdown-toggle"

        data-bs-toggle="dropdown"

        aria-expanded="false"

    >

        Carbohydrates

    </button>

    <ul class="dropdown-menu" aria-labelledby="btnGroupDrop1">

        <div class="form-check">

```

```
<input
  class="form-check-input"
  type="checkbox"
  value="bread"
  id="flexCheckDefault"
  name="check"
/>

<label class="form-check-label" for="flexCheckDefault">
  Bread
</label>
</div>

<div class="form-check">
  <input
    class="form-check-input"
    type="checkbox"
    value="oats"
    id="flexCheckDefault"
    name="check"
  />

  <label class="form-check-label" for="flexCheckDefault">
    Oats
  </label>
</div>
```

```
<div class="form-check">

  <input

    class="form-check-input"

    type="checkbox"

    value="rice"

    id="flexCheckDefault"

    name="check"

  />

  <label class="form-check-label" for="flexCheckDefault">

    Rice

  </label>

</div>

<div class="form-check">

  <input

    class="form-check-input"

    type="checkbox"

    value="cereal"

    id="flexCheckDefault"

    name="check"

  />

  <label class="form-check-label" for="flexCheckDefault">

    Cereal

  </label>
```

```
</div>

<div class="form-check">

  <input

    class="form-check-input"

    type="checkbox"

    value="pasta"

    id="flexCheckDefault"

    name="check"

  />

  <label class="form-check-label" for="flexCheckDefault">

    Pasta

  </label>

</div>

</ul>

</div>

<div class="btn-group" role="group">

  <button

    id="btnGroupDrop1"

    type="button"

    class="btn btn-primary dropdown-toggle"

    data-bs-toggle="dropdown"

    aria-expanded="false"

  >
```

Dairy

```
</button>
```

```
<ul class="dropdown-menu" aria-labelledby="btnGroupDrop1">
```

```
<div class="form-check">
```

```
<input
```

```
  class="form-check-input"
```

```
  type="checkbox"
```

```
  value="milk"
```

```
  id="flexCheckDefault"
```

```
  name="check"
```

```
>
```

```
<label class="form-check-label" for="flexCheckDefault">
```

```
  Milk
```

```
</label>
```

```
</div>
```

```
<div class="form-check">
```

```
<input
```

```
  class="form-check-input"
```

```
  type="checkbox"
```

```
  value="egg"
```

```
  id="flexCheckDefault"
```

```
  name="check"
```

```
>
```



```
<label class="form-check-label" for="flexCheckDefault">
    Eggs
</label>
</div>
<div class="form-check">
    <input
        class="form-check-input"
        type="checkbox"
        value="cheese"
        id="flexCheckDefault"
        name="check"
    />
    <label class="form-check-label" for="flexCheckDefault">
        Cheese
    </label>
</div>
<div class="form-check">
    <input
        class="form-check-input"
        type="checkbox"
        value="butter"
        id="flexCheckDefault"
        name="check"
    />
    <label class="form-check-label" for="flexCheckDefault">
        Butter
    </label>
</div>
```

```
    />

    <label class="form-check-label" for="flexCheckDefault">

        Butter

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="yoghurt"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Yoghurt

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="cream"

        id="flexCheckDefault"
```

```

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Cream

    </label>

</div>

</ul>

</div>

<div class="btn-group" role="group">

    <button

        id="btnGroupDrop1"

        type="button"

        class="btn btn-primary dropdown-toggle"

        data-bs-toggle="dropdown"

        aria-expanded="false"

    >

        Meat

    </button>

    <ul class="dropdown-menu" aria-labelledby="btnGroupDrop1">

        <div class="form-check">

            <input

                class="form-check-input"

                type="checkbox"

```

```
        value="chicken"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Chicken

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="beef"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Beef

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"
```

```
        type="checkbox"

        value="lamb"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Lamb

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="pork"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Pork

    </label>

</div>

<div class="form-check">

    <input
```

```

        class="form-check-input"

        type="checkbox"

        value="salmon"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Salmon

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="cod"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault"> Cod
</label>

</div>

<div class="form-check">

    <input

```

```

        class="form-check-input"

        type="checkbox"

        value="turkey"

        id="flexCheckDefault"

        name="check"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Turkey

    </label>

</div>

</ul>

</div>

<div class="form-group w-35">

    <br />

    <div class="mb-3">

        <input

            type="text"

            class="form-control"

            id="ingredientsinput"

            placeholder="Enter any additional ingredients, separated by
commas"

            name="ingredients"

            data-bs-toggle="tooltip"

```

```

        title="Enter any additional ingredients, separated by
commas"

    />
</div>

<div class="btn-group" role="group">

    <button

        id="btnGroupDrop1"

        type="button"

        class="btn btn-info dropdown-toggle"

        data-bs-toggle="dropdown"

        aria-expanded="false"

    >

        Meal Type

    </button>

    <ul class="dropdown-menu" aria-labelledby="btnGroupDrop1"
required>

        <div class="form-check">

            <input

                class="form-check-input"

                type="radio"

                value="Breakfast"

                id="flexCheckDefault"

                name="mealcheck"

            />

```



```
<label class="form-check-label" for="flexCheckDefault">
    Breakfast
</label>
</div>
<div class="form-check">
    <input
        class="form-check-input"
        type="radio"
        value="Lunch"
        id="flexCheckDefault"
        name="mealcheck"
    />
    <label class="form-check-label" for="flexCheckDefault">
        Lunch
    </label>
</div>
<div class="form-check">
    <input
        class="form-check-input"
        type="radio"
        value="Dinner"
        id="flexCheckDefault"
        name="mealcheck"
    />
    <label class="form-check-label" for="flexCheckDefault">
        Dinner
    </label>
</div>
```

```
    />

    <label class="form-check-label" for="flexCheckDefault">

        Dinner

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="radio"

        value="Snack"

        id="flexCheckDefault"

        name="mealcheck"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Snack

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="radio"

        value="Teatime"

        id="flexCheckDefault"
```

```

        name="mealcheck"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Teatime

    </label>

</div>

</ul>

</div>

<div class="btn-group" role="group">

    <button

        id="btnGroupDrop1"

        type="button"

        class="btn btn-info dropdown-toggle"

        data-bs-toggle="dropdown"

        aria-expanded="false"

    >

        Dietary Requirements

    </button>

    <ul class="dropdown-menu" aria-labelledby="btnGroupDrop1">

        <div class="form-check">

            <input

                class="form-check-input"

                type="checkbox"

```

```
        value="alcohol-free"

        id="flexCheckDefault"

        name="dietcheck"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Alcohol-Free

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="celery-free"

        id="flexCheckDefault"

        name="dietcheck"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Celery-Free

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"
```

```
        type="checkbox"

        value="dairy-free"

        id="flexCheckDefault"

        name="dietcheck"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Dairy-Free

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="egg-free"

        id="flexCheckDefault"

        name="dietcheck"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Egg-Free

    </label>

</div>

<div class="form-check">

    <input
```

```
        class="form-check-input"

        type="checkbox"

        value="gluten-free"

        id="flexCheckDefault"

        name="dietcheck"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Gluten-Free

    </label>

</div>

<div class="form-check">

    <input

        class="form-check-input"

        type="checkbox"

        value="kosher"

        id="flexCheckDefault"

        name="dietcheck"

    />

    <label class="form-check-label" for="flexCheckDefault">

        Kosher

    </label>

</div>

<div class="form-check">
```

```
<input
  class="form-check-input"
  type="checkbox"
  value="peanut-free"
  id="flexCheckDefault"
  name="dietcheck"
/>
<label class="form-check-label" for="flexCheckDefault">
  Peanut-Free
</label>
</div>
<div class="form-check">
  <input
    class="form-check-input"
    type="checkbox"
    value="pescatarian"
    id="flexCheckDefault"
    name="dietcheck"
  />
  <label class="form-check-label" for="flexCheckDefault">
    Pescatarian
  </label>
</div>
```

```
<div class="form-check">

  <input

    class="form-check-input"

    type="checkbox"

    value="vegan"

    id="flexCheckDefault"

    name="dietcheck"

  />

  <label class="form-check-label" for="flexCheckDefault">

    Vegan

  </label>

</div>

<div class="form-check">

  <input

    class="form-check-input"

    type="checkbox"

    value="vegetarian"

    id="flexCheckDefault"

    name="dietcheck"

  />

  <label class="form-check-label" for="flexCheckDefault">

    Vegetarian

  </label>

</div>
```



```

        </div>

    </ul>

</div>

<br />

<br />

<div class="col-md-12 text-center">

    <button type="submit" class="btn btn-dark">Search</button>

</div>

</div>

</form>

<div id="selected-items"></div>

<button id="clear-selections" class="btn btn-danger btn-sm
mt-3">X</button>

</div>

</div>

{% endblock %}

```

## Gpt.html

```

<!-- Author - Mohsin Tahir
Date - 17/04/23 -->

{% extends "base.html" %}

{% block body %}

```

```

<link rel="stylesheet" href="{{ url_for('static', filename='gpt.css')
}}">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">

  <a class="navbar-brand">Macro Meals</a>

  <button

    class="navbar-toggler"

    type="button"

    data-toggle="collapse"

    data-target="#navbarSupportedContent"

    aria-controls="navbarSupportedContent"

    aria-expanded="false"

    aria-label="Toggle navigation"

  >

    <span class="navbar-toggler-icon"></span>

  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">

    <ul class="navbar-nav mr-auto">

      <li class="nav-item">

        <a class="nav-link" href="/recipe">Search Recipes</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="/savedRecipes">Saved Recipes</a>

      </li>

```

```

    {% if "username" not in session %}

    <li class="nav-item">

        <a class="nav-link text-success" href="/logout">Login</a>

    </li>

    {% else %}

    <li class="nav-item">

        <a class="nav-link text-danger" href="/logout">Logout</a>

    </li>

    {% endif %}

</ul>

</div>

</nav>

<br><br>

<div class="d-flex justify-content-center">

    <div class="text-center">

        <h1>Carbon Footprint Friendly Alternatives Recommended by
ChatGPT</h1>

        <ul class="center-ul">

            {% for substring in output %}

                {% if substring.strip() %}

                    {% if loop.first %}

                        <li>{{ substring.strip() [2:] }}</li>

                    {% else %}

```

```

        <li>{{ substring.strip() }}</li>

        {% endif %}

    {% endif %}

{% endfor %}

</ul>

</div>

</div>

{% endblock %}

```

## Index.html

```

<!-- Author - Mohsin Tahir
Date - 17/04/23 -->

{% extends "base.html" %}

{% block body %}

<link rel="stylesheet" href="{{ url_for('static', filename='index.css')
}}">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">

    <a class="navbar-brand" >Macro Meals</a>

    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent"

```

```

aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">

    <span class="navbar-toggler-icon"></span>

</button>

</nav>

<br><br><br>

<div class="text-center d-flex flex-column align-items-center">

    <h1>Welcome to Macro Meals</h1>

    <br>

    <p class="text-lg">

        Recipe Recommendation System with a Cost and Carbon Footprint
Indicator and AI Suggestions. <br>

        Offering a sustainable and personalised approach to meal planning.

    </p>

    <div class="d-flex justify-content-center">

        <a href="/login" class="btn btn-primary">Register/Login</a>

        <a href="/recipe" class="btn btn-success">Find a Recipe</a>

    </div>

</div>

{% endblock %}

```

## Login.html

```
<!-- Author - Mohsin Tahir
```

```

Date - 17/04/23 -->

{% extends "base.html" %}

{% block body %}

<link rel="stylesheet" href="{{ url_for('static', filename='login.css')
}}">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">

  <a class="navbar-brand">Macro Meals</a>

  <button

    class="navbar-toggler"

    type="button"

    data-toggle="collapse"

    data-target="#navbarSupportedContent"

    aria-controls="navbarSupportedContent"

    aria-expanded="false"

    aria-label="Toggle navigation"

  >

    <span class="navbar-toggler-icon"></span>

  </button>

  <div class="collapse navbar-collapse" id="navbarSupportedContent">

    <ul class="navbar-nav mr-auto">

      <li class="nav-item">

        <a class="nav-link" href="/recipe">Search Recipes</a>

      </li>

```

```

    <li class="nav-item">
      <a class="nav-link" href="/savedRecipes">Saved Recipes</a>
    </li>
  </ul>
</div>
</nav>
<br><br><br>
<div class="d-flex justify-content-center">
  <div class="card" style="width: 30rem;">
    <div class="card-body">
      <form action="/login" method="post" class="needs-validation"
novalidate>
        <p class="h4 text-center">
          Login
        </p>
        <div class="mb-3">
          <label for="usernameID" class="form-label">Username</label>
          <input type="text" class="form-control" id="usernameID"
name="username" required>
          <div class="invalid-feedback">
            Please enter an username.
          </div>
        </div>
      </div>
    <div class="mb-3">

```

```

    <label for="passwordID" class="form-label">Password</label>

    <input type="password" class="form-control" id="passwordID"
name="password" required>

    <div class="invalid-feedback">

        Please enter a password.

    </div>

</div>

<div class="text-center">

    {% if errors %}

    <div class="alert alert-danger" role="alert">

        {{errors}}

    </div>

    {% endif %}

    <div class="text-center">

        <button type="submit" class="btn btn-success btn-with-icon
mt-3">

            <span class="material-icons">login</span>

            Login

        </button>

        <br>

        <a href="/register" class="btn btn-secondary btn-with-icon
mt-3">

            <span class="material-icons">person_add</span>

            Register

```



```

    </a>

    <br>

    <a href="/passreset" class="btn btn-primary btn-with-icon
mt-3">

        <span class="material-icons">lock_open</span>

        Reset Password

    </a>

</div>

</div>

</form>

</div>

</div>

</div>

<script>

(function () {

    'use strict'

    var forms = document.querySelectorAll('.needs-validation')

    Array.prototype.slice.call(forms)

        .forEach(function (form) {

            form.addEventListener('submit', function (event) {

                if (!form.checkValidity()) {

```

```

        event.preventDefault()

        event.stopPropagation()

    }

    form.classList.add('was-validated')

    }, false)

    })

    }) ()

</script>

{% endblock %}

```

## Nav.html

```

<!-- Author - Mohsin Tahir
Date - 17/04/23 -->

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

    <a class="navbar-brand" href="/">Macro Meals</a>

    <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">

```

```

    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="/">Home</a>
      </li>
    </ul>
  </div>
</nav>

```

## Register.html

```

<!-- Author - Mohsin Tahir
Date - 17/04/23 -->
{% extends "base.html" %}
{% block body %}
<link rel="stylesheet" href="{{ url_for('static',
filename='register.css') }}">
<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
  <a class="navbar-brand">Macro Meals</a>
  <button
    class="navbar-toggler"
    type="button"
    data-toggle="collapse"
    data-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent"

```

```

    aria-expanded="false"

    aria-label="Toggle navigation"

>

    <span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">

    <ul class="navbar-nav mr-auto">

        <li class="nav-item">

            <a class="nav-link" href="/recipe">Search Recipes</a>

        </li>

        <li class="nav-item">

            <a class="nav-link" href="/savedRecipes">Saved Recipes</a>

        </li>

    </ul>

</div>

</nav>

<br><br><br>

<div class="d-flex justify-content-center">

    <div class="card" style="width: 30rem;">

        <div class="card-body">

            <form action="/processform" method="post" class
="needs-validation" novalidate>

```

```

    <p class="h4 text-center">
      Register
    </p>

    <div class="mb-3">
      <label for="usernameID" class="form-label">Username</label>
      <input type="text" class="form-control" id="usernameID"
name="username" required>
      <div class="invalid-feedback">
        Please enter an username.
      </div>
    </div>

    <div class="mb-3">
      <label for="emailID" class="form-label">Email address</label>
      <input type="email" class="form-control" id="emailID"
aria-describedby="emailHelp" name="email" required>
      <div class="invalid-feedback">
        Please enter a valid email address.
      </div>
    </div>

    <div class="mb-3">
      <label for="passwordID" class="form-label">Password (min length
is 8 characters)</label>
      <input type="password" class="form-control" id="passwordID"
name="password" required minlength="8">

```

```

    <div class="invalid-feedback">

        Please enter a password.

    </div>

</div>

<div class="mb-3">

    <label for="passwordrepeatID" class="form-label">Repeat
Password</label>

    <input type="password" class="form-control" id="passwordrepeatID"
name="repeatpassword" required minlength="8">

    <div class="invalid-feedback">

        Please re-enter your password.

    </div>

</div>

<div class="text-center">

    {% if errors %}

    <div class="alert alert-danger" role="alert">

        {% for e in errors %}

            {{e}} <br>

        {% endfor %}

    </div>

    {% endif %}

    <button type="submit" class="btn btn-success btn-with-icon
btn-block mt-3">

        <span class="material-icons">person_add</span>

```

```

        Submit
    </button>

    <br>

    <a href="/login" class="btn btn-secondary btn-with-icon btn-block
mt-3">

        <span class="material-icons">login</span>

        Login
    </a>
</div>
</form>
</div>
</div>
</div>
</div>
<script>
(function () {
    'use strict'

    var forms = document.querySelectorAll('.needs-validation')

    Array.prototype.slice.call(forms)
        .forEach(function (form) {
            form.addEventListener('submit', function (event) {
                if (!form.checkValidity()) {

```

```

        event.preventDefault()

        event.stopPropagation()

    }

    form.classList.add('was-validated')

    }, false)

    })

}) ()

</script>

{% endblock %}

```

## Results.html

```

<!-- Author - Mohsin Tahir
Date - 17/04/23 -->

{% extends "base.html" %} {% block body %}

<link rel="stylesheet" href="{{ url_for('static',
filename='results.css') }}">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">

    <a class="navbar-brand">Macro Meals</a>

    <button

        class="navbar-toggler"

        type="button"

        data-toggle="collapse"

```



```

data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent"
aria-expanded="false"
aria-label="Toggle navigation"
>
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item">
      <a class="nav-link" href="/recipe">Search Recipes</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="/savedRecipes">Saved Recipes</a>
    </li>
    {% if "username" not in session %}
    <li class="nav-item">
      <a class="nav-link text-success" href="/logout">Login</a>
    </li>
    {% else %}
    <li class="nav-item">
      <a class="nav-link text-danger" href="/logout">Logout</a>
    </li>

```

```

        {% endif %}

    </ul>

</div>

</nav>

<br />

{% if not data%}

<div class="text-center alert-centered">

    <div class="alert alert-warning alert-large" role="alert">

        Sorry no recipes found with your inputted ingredients

    </div>

</div>

{% else %}

<div class="text-center"><h1>Your Personalised Recipes</h1>

    <div class="dropdown">

        <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton" data-toggle="dropdown" aria-haspopup="true"
aria-expanded="false">

            Sort Recipes

        </button>

        <div class="dropdown-menu" aria-labelledby="dropdownMenuButton">

            <div class="text-left">

                <form method="post" action="{{ url_for('sort_results') }}">

                    <input type="hidden" name="meal" value="{{meal}}">

                    <input type="hidden" name="ing" value="{{ing}}">

```

```
<input type="hidden" name="health" value="{{health}}">

<button class="dropdown-item" type="submit" name="criteria"
value="caloriesLH">

    Calories (Low-High)

</button>

<button class="dropdown-item" type="submit" name="criteria"
value="caloriesHL">

    Calories (High-Low)

</button>

<button class="dropdown-item" type="submit" name="criteria"
value="servingLH">

    Serving Size (Low-High)

</button>

<button class="dropdown-item" type="submit" name="criteria"
value="servingHL">

    Serving Size (High-Low)

</button>

<button class="dropdown-item" type="submit" name="criteria"
value="carbsLH">

    Carbohydrates (Low-High)

</button>

<button class="dropdown-item" type="submit" name="criteria"
value="carbsHL">

    Carbohydrates (High-Low)

</button>
```

```
        <button class="dropdown-item" type="submit" name="criteria"
value="fatLH">

        Fat (Low-High)

    </button>

    <button class="dropdown-item" type="submit" name="criteria"
value="fatHL">

        Fat (High-Low)

    </button>

    <button class="dropdown-item" type="submit" name="criteria"
value="proteinLH">

        Protein (Low-High)

    </button>

    <button class="dropdown-item" type="submit" name="criteria"
value="proteinHL">

        Protein (High-Low)

    </button>

    <button class="dropdown-item" type="submit" name="criteria"
value="sugarLH">

        Sugar (Low-High)

    </button>

    <button class="dropdown-item" type="submit" name="criteria"
value="sugarHL">

        Sugar (High-Low)

    </button>
```

```

        <button class="dropdown-item" type="submit" name="criteria"
value="fiberLH">
        Fibre (Low-High)
    </button>

    <button class="dropdown-item" type="submit" name="criteria"
value="fiberHL">
        Fibre (High-Low)
    </button>

</form>

</div>
</div>

</div>

</div>

<br />

<div class="d-flex flex-wrap justify-content-evenly">

    {% for row in data %} {% set val = cfv[loop.index0] if cfv else None
%} {% set cost_val = cost[loop.index0] if cost else None %}

    <div class="d-flex flex-column position-relative border-0 card mb-5"
style="width: 17rem">

        

<div class="col-md-8 d-flex flex-grow-1">

  <div class="card-body d-flex flex-column">

    <h5 class="card-title">{{row['recipe']['label']}}</h5>

    {% if val %}

    <br>

    <h5>Carbon Footprint Indicator</h5>

    <div class="traffic-light">

      {% if val == 'GREEN' %}

      <div class="light red"></div>

      <div class="light green activelightgreen"></div>

      {% endif %} {% if val == 'RED' %}

      <div class="light red activelightred"></div>

      <div class="light green"></div>

      {% endif %}

    </div>

    {% endif %}

    <br>

    {% if cost_val %}

    <h5>Cost Estimation</h5>

    <h6>{{cost_val}}</h6>

    {% endif %}

  </div>

```

```

</div>

<button
  type="button"
  class="btn btn-dark mt-auto"
  data-toggle="modal"
  data-target="#itemModal-{{ loop.index }}"
>
  Click for More Info
</button>

<!-- Modal -->
<div
  class="modal fade"
  id="itemModal-{{ loop.index }}"
  tabindex="-1"
  role="dialog"
  aria-labelledby="itemModalLabel-{{ loop.index }}"
  aria-hidden="true"
>
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">

```

```

<h5 class="modal-title" id="itemModalLabel-{{ loop.index
}}">
    {{ row['recipe']['label'] }}
</h5>
<button
    type="button"
    class="close"
    data-dismiss="modal"
    aria-label="Close"
>
    <span aria-hidden="true">&times;</span>
</button>
</div>
<div class="modal-body">
    <b>Ingredients:</b>
    {% for row in row['recipe']['ingredientLines'] %}
    <br />
    {{row}} {% endfor %}
    <br />
    <hr />
    <b>Serving Size:</b> {{(row['recipe']['yield'])|int }}
    <br />
    <b>Calories per serving:</b>
    {{{(row['recipe']['calories'])|int} /

```



```

        (row['recipe']['yield']|int) |int }}

<br />

<hr />

<b>Nutritional Breakdown per serving:</b>

<br />

        {{row['recipe']['totalNutrients']['CHOCDF.net']['label']}}

{{{(row['recipe']['totalNutrients']['CHOCDF.net']['quantity']|int /
        (row['recipe']['yield']|int) |int)}}

        {{row['recipe']['totalNutrients']['CHOCDF.net']['unit']}}

<br />

        {{row['recipe']['totalNutrients']['FAT']['label']}}

        {{{(row['recipe']['totalNutrients']['FAT']['quantity']|int
/
        (row['recipe']['yield']|int) |int)}}

        {{row['recipe']['totalNutrients']['FAT']['unit']}}

<br />

        {{row['recipe']['totalNutrients']['PROCNT']['label']}}

        {{{(row['recipe']['totalNutrients']['PROCNT']['quantity']|int /
        (row['recipe']['yield']|int) |int)}}

        {{row['recipe']['totalNutrients']['PROCNT']['unit']}}

<br />

        {{row['recipe']['totalNutrients']['SUGAR']['label']}}

```

```

{{{(row['recipe']['totalNutrients']['SUGAR']['quantity']|int /
      (row['recipe']['yield']|int)|int)}}
      {{row['recipe']['totalNutrients']['SUGAR']['unit']}}
    <br />
    Fibre

{{{(row['recipe']['totalNutrients']['FIBTG']['quantity']|int /
      (row['recipe']['yield']|int)|int)}}
      {{row['recipe']['totalNutrients']['FIBTG']['unit']}}
    <hr />
    <a
      href="{{ row['recipe']['url'] }}"
      target="_blank"
      class="btn btn-outline-success"
      data-ingredients="{{ row['recipe']['ingredientLines'] }}"
      data-target="#itemModal-{{ loop.index }}"
      >Get Instructions</a
    >
    <br>
    <form id="gpt-form-{{ loop.index }}" action="/gpt"
method="post">
      <input type="hidden" name="Ingredients"
value="{{row['recipe']['ingredientLines']}}" />

```

```

        <button type="submit" class="btn btn-outline-warning">Ask
GPT</button>

</form>

<form action="/addrecipe" method="post">

    <!--

        Hidden fields to hold all recipe information so it can be
called by the save recipe function

    -->

    <input type="hidden" name="carbon" value="{{val}}" />

    <input type="hidden" name="cost" value="{{cost_val}}" />

    <input

        type="hidden"

        name="Ingredients"

        value="{{row['recipe']['ingredientLines']}}"

    />

    <input

        type="hidden"

        name="Calories"

        value="{{((row['recipe']['calories'])|int /
(row['recipe']['yield'])|int) |int }}"

    />

    <input

        type="hidden"

        name="Servings"

```

```

        value="{{(row['recipe']['yield'])|int }}"

    />

    <input

        type="hidden"

        name="Carbs value"

        value="
{{((row['recipe']['totalNutrients']['CHOCDF.net']['quantity'])|int /
(row['recipe']['yield'])|int)|int}}"

    />

    <input

        type="hidden"

        name="Fat value"

        value="
{{((row['recipe']['totalNutrients']['FAT']['quantity'])|int /
(row['recipe']['yield'])|int)|int}}"

    />

    <input

        type="hidden"

        name="Protein value"

        value="
{{((row['recipe']['totalNutrients']['PROCNT']['quantity'])|int /
(row['recipe']['yield'])|int)|int}}"

    />

    <input

        type="hidden"

```

```

        name="Sugars value"

        value="
{{{(row['recipe']['totalNutrients']['SUGAR']['quantity'])|int /
(row['recipe']['yield'])|int)|int}}}"

    />

    <input

        type="hidden"

        name="Fiber value"

        value="
{{{(row['recipe']['totalNutrients']['FIBTG']['quantity'])|int /
(row['recipe']['yield'])|int)|int}}}"

    />

    <input

        type="hidden"

        name="Link"

        value="{{ row['recipe']['url'] }}"

    />

    <input

        type="hidden"

        name="Label"

        value="{{ row['recipe']['label'] }}"

    />

    {% if "username" not in session %}

    <button disabled type="submit" class="btn btn-outline">

```

```

        Login to Save Recipes

    </button>

    {% else %}

    <button type="submit" class="btn btn-outline-info">

        Save Recipe

    </button>

    {% endif %}

</form>

</div>

<div class="modal-footer">

    <button

        type="button"

        class="btn btn-secondary"

        data-dismiss="modal"

    >

        Close

    </button>

</div>

</div>

</div>

</div>

</div>

{% endfor %} {% endif %}

```

```
</div>

{% endblock %}
```

## savedRecipes.html

```
<!-- Author - Mohsin Tahir
Date - 17/04/23 -->

{% extends "base.html" %} {% block body %}

<link rel="stylesheet" href="{% url_for('static',
filename='savedRecipes.css') %}">

<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">

  <a class="navbar-brand">Macro Meals</a>

  <button

    class="navbar-toggler"

    type="button"

    data-toggle="collapse"

    data-target="#navbarSupportedContent"

    aria-controls="navbarSupportedContent"

    aria-expanded="false"

    aria-label="Toggle navigation"

  >

    <span class="navbar-toggler-icon"></span>

  </button>
```

```

<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item">
      <a class="nav-link" href="/recipe">Search Recipes</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="/savedRecipes">Saved Recipes</a>
    </li>
    <li class="nav-item">
      <a class="nav-link text-danger" href="/logout">Logout</a>
    </li>
  </ul>
</div>
</nav>
{% if not res%}
<div class="text-center alert-centered">
  <div class="alert alert-danger alert-large" role="alert">
    Sorry, you have no Recipes Saved
  </div>
</div>
<!-- <p class="text-center">Sorry you have no Recipes Saved</p -->
{% else %}
<div class="text-center"><h1>Your Saved Recipes</h1></div>

```



```

<br>

<div class="d-flex flex-wrap justify-content-evenly">

  {% for row in res %}

    <div class="d-flex flex-column position-relative border-0 card mb-5"
    style="width: 17rem">

      <div class="col-md-8 d-flex flex-grow-1">

        <div class="card-body d-flex flex-column">

          <h5 class="card-title">{{row[11]}}</h5>

          <br>

          <h5>Carbon Footprint Indicator</h5>

          <div class="traffic-light">

            {% if row[12] == 'GREEN' %}

              <div class="light red"></div>

              <div class="light green activelightgreen"></div>

            {% endif %} {% if row[12] == 'RED' %}

              <div class="light red activelightred"></div>

              <div class="light green"></div>

            {% endif %}

          </div>

          <br>

          <h5>Cost Estimation</h5>

          <h6>{{row[13]}}</h6>

        </div>

```

```

</div>

<button
  type="button"
  class="btn btn-dark"
  data-toggle="modal"
  data-target="#itemModal-{{ loop.index }}"
>
  Click for More Info
</button>

<!-- Modal -->
<div
  class="modal fade"
  id="itemModal-{{ loop.index }}"
  tabindex="-1"
  role="dialog"
  aria-labelledby="itemModalLabel-{{ loop.index }}"
  aria-hidden="true"
>
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">

```

```

    <h5 class="modal-title" id="itemModalLabel-{{ loop.index
}}">

        {{ row[11] }}

    </h5>

    <button

        type="button"

        class="close"

        data-dismiss="modal"

        aria-label="Close"

    >

        <span aria-hidden="true">&times;</span>

    </button>

</div>

<div class="modal-body">

    <b>Ingredients:</b>

    {% for ing in row[2].split(", ") %}

    <br />

    {{ing.strip(" ")}} {% endfor %}

    <br />

    <hr />

    <b>Calories per serving:</b> {{ (row[3]) }}

    <br />

    <b>Serving Size:</b> {{(row[4]) }}

```

```

<br />

<hr />

<b>Nutritional Breakdown per serving:</b>

<br />

Carbohydrates (net) {{(row[5])}} g

<br />

Fat {{(row[6])}} g

<br />

Protein {{row[7]}} g

<br />

Sugars {{row[8]}} g

<br />

Fibre {{row[9]}} g

<hr />

<a

    href="{{ row[10] }}"

    target="_blank"

    class="btn btn-outline-success"

    >Get Instructions</a

><br />

<form id="gpt-form-{{ loop.index }}" action="/gpt"
method="post">

    <input type="hidden" name="Ingredients"
value="{{row[2]}}" />

```

```

        <button type="submit" class="btn btn-outline-warning">Ask
GPT</button>

    </form>

    <br />

    <!--Session to store the link so it can be used for
delete-->

    <form action="/deleteRecipe" method="post">

        <input type="hidden" name="Link" value="{{ row[10] }}" />

        <button type="submit" class="btn btn-danger">

            Delete Recipe

        </button>

    </form>

</div>

<div class="modal-footer">

    <button

        type="button"

        class="btn btn-secondary"

        data-dismiss="modal"

    >

        Close

    </button>

</div>

</div>

```

```
    </div>

  </div>

</div>

{% endfor %} {% endif %}

</div>

{% endblock %}
```

## CSS Code

### Changepw.css

```
body {

background: linear-gradient(to bottom, #89ffde 0%, #8f4bff 100%)
no-repeat

  fixed;

background: url("doodle.jpg") no-repeat

  center center fixed;

-webkit-background-size: cover;

-moz-background-size: cover;

-o-background-size: cover;

background-size: cover;

position: relative;

}

.btn-with-icon {
```

```
display: inline-flex;
align-items: center;
}

.btn-with-icon span.material-icons {
margin-right: 4px;
}
```

## Edamam.css

```
body {
    background: linear-gradient(to bottom, #89ffde 0%, #8f4bff 100%)
no-repeat
    fixed;
    background: url("green.jpg") no-repeat
    center center fixed;
    -webkit-background-size: cover;
    -moz-background-size: cover;
    -o-background-size: cover;
    background-size: cover;
    position: relative;
}

.dropdown {
    width: 100%;
```

```
text-align: center;
}
```

## Gpt.css

```
body {
background: linear-gradient(to bottom, #89ffde 0%, #8f4bff 100%)
no-repeat
    fixed;
background: url("green.jpg") no-repeat
    center center fixed;
-webkit-background-size: cover;
-moz-background-size: cover;
-o-background-size: cover;
background-size: cover;
position: relative;
}

.center-ul {
display: inline-block;
text-align: left;
}
```



## Index.css

```
body {  
  
background: linear-gradient(to bottom, #89ffde 0%, #8f4bff 100%)  
no-repeat  
  
    fixed;  
  
background: url("bg.jpg") no-repeat  
  
/* Reference  
  
https://www.pexels.com/photo/assorted-vegetables-on-brown-wooden-table-1414651/  
  
*/  
  
    center center fixed;  
  
-webkit-background-size: cover;  
-moz-background-size: cover;  
-o-background-size: cover;  
background-size: cover;  
  
position: relative;  
  
}  
  
.btn:hover {  
  
    transform: scale(1.1);  
  
}
```

## Login.css

```
body {  
  
background: linear-gradient(to bottom, #89ffde 0%, #8f4bff 100%)  
no-repeat  
  
    fixed;  
  
background: url("doodle.jpg") no-repeat  
  
    center center fixed;  
  
-webkit-background-size: cover;  
-moz-background-size: cover;  
-o-background-size: cover;  
background-size: cover;  
  
position: relative;  
}  
  
.btn-with-icon {  
  
display: inline-flex;  
  
align-items: center;  
}  
  
.btn-with-icon span.material-icons {  
  
margin-right: 4px;  
}
```

## Register.css

```
body {
```

```

background: linear-gradient(to bottom, #89ffde 0%, #8f4bff 100%)
no-repeat

    fixed;

background: url("doodle.jpg") no-repeat

    center center fixed;

-webkit-background-size: cover;

-moz-background-size: cover;

-o-background-size: cover;

background-size: cover;

position: relative;
}

.btn-with-icon {

display: inline-flex;

align-items: center;

}

.btn-with-icon span.material-icons {

margin-right: 4px;

}

```

## Results.css

```

body {

background: linear-gradient(to bottom, #89ffde 0%, #8f4bff 100%)
no-repeat

```

```
    fixed;

background: url("green.jpg") no-repeat

    center center fixed;

-webkit-background-size: cover;

-moz-background-size: cover;

-o-background-size: cover;

background-size: cover;

position: relative;

}

.card-title {

margin-bottom: 0;

}

.traffic-light {

width: 75px;

height: 30px;

background-color: black;

border-radius: 10px;

display: flex;

flex-direction: row;

justify-content: space-around;

align-items: center;

}
```

```
.light {
width: 20px;
height: 20px;
border-radius: 50%;
}

.red {
background-color: red;
}

.green {
background-color: green;
}

.activelightrd {
background-color: red;
box-shadow: 0px 0px 5px #ff0, 0px 0px 10px red, 0px 0px 15px red;
animation: flashRed 0.5s infinite;
}

.activelightgreen {
background-color: #00ff00;
```

```
box-shadow: 0px 0px 5px #ff0, 0px 0px 10px #0f0, 0px 0px 15px #0f0;
animation: flashGreen 0.5s infinite;
}

@keyframes flashRed {
0% {
    box-shadow: 0px 0px 5px #ff0, 0px 0px 10px red, 0px 0px 15px red;
}
50% {
    box-shadow: 0px 0px 10px #f00, 0px 0px 15px #f00, 0px 0px 20px
#f00;
}
100% {
    box-shadow: 0px 0px 5px #ff0, 0px 0px 10px red, 0px 0px 15px red;
}
}

@keyframes flashGreen {
0% {
    box-shadow: 0px 0px 5px #ff0, 0px 0px 10px #0f0, 0px 0px 15px #0f0;
}
50% {
    box-shadow: 0px 0px 10px #0f0, 0px 0px 15px #0f0, 0px 0px 20px
#0f0;
}
```

```
}  
  
100% {  
    box-shadow: 0px 0px 5px #ff0, 0px 0px 10px #0f0, 0px 0px 15px #0f0;  
}  
}  
  
.alert-centered {  
position: fixed;  
top: 50%;  
left: 50%;  
transform: translate(-50%, -50%);  
}  
  
.alert-large {  
font-size: 1.8rem;  
}
```

### savedRecipes.css

```
body {  
background: linear-gradient(to bottom, #89ffde 0%, #8f4bff 100%)  
no-repeat  
    fixed;  
background: url("green.jpg") no-repeat  
    center center fixed;
```

```
-webkit-background-size: cover;

-moz-background-size: cover;

-o-background-size: cover;

background-size: cover;

position: relative;

}

.card-title {

margin-bottom: 0;

}

.traffic-light {

width: 75px;

height: 30px;

background-color: black;

border-radius: 10px;

display: flex;

flex-direction: row;

justify-content: space-around;

align-items: center;

}

.light {

width: 20px;
```



```
height: 20px;
border-radius: 50%;
}

.red {
background-color: red;
}

.green {
background-color: green;
}

.activelightred {
background-color: red;
box-shadow: 0px 0px 5px #ff0, 0px 0px 10px red, 0px 0px 15px red;
animation: flashRed 0.5s infinite;
}

.activelightgreen {
background-color: #00ff00;
box-shadow: 0px 0px 5px #ff0, 0px 0px 10px #0f0, 0px 0px 15px #0f0;
animation: flashGreen 0.5s infinite;
}
```

```
@keyframes flashRed {
0% {
    box-shadow: 0px 0px 5px #ff0, 0px 0px 10px red, 0px 0px 15px red;
}
50% {
    box-shadow: 0px 0px 10px #f00, 0px 0px 15px #f00, 0px 0px 20px
#f00;
}
100% {
    box-shadow: 0px 0px 5px #ff0, 0px 0px 10px red, 0px 0px 15px red;
}
}

@keyframes flashGreen {
0% {
    box-shadow: 0px 0px 5px #ff0, 0px 0px 10px #0f0, 0px 0px 15px #0f0;
}
50% {
    box-shadow: 0px 0px 10px #0f0, 0px 0px 15px #0f0, 0px 0px 20px
#0f0;
}
100% {
    box-shadow: 0px 0px 5px #ff0, 0px 0px 10px #0f0, 0px 0px 15px #0f0;
}
```

```
}  
  
}  
  
.alert-centered {  
position: fixed;  
top: 50%;  
left: 50%;  
transform: translate(-50%, -50%);  
}  
  
.alert-large {  
font-size: 1.8rem;  
}
```