

cprime

Food Recipe Generator Using HTML,CSS,JS



NAME : T MANOJ

EMPID : LYAKE2KHS

Introduction

This Food Recipe Generator is a web application that allows users to search for recipes based on meal names. It fetches recipe data from TheMealDB API and displays the results in an attractive, user-friendly interface. The application provides recipe details including ingredients, instructions, and other relevant information when a user selects a specific recipe.

Key Features

1. **Search Functionality:** Users can search for recipes by entering meal names
2. **Recipe Display:** Shows recipe cards with images, names, categories, and origins
3. **Detailed View:** Popup modal displays complete recipe details including ingredients and preparation instructions
4. **Responsive Design:** Works well on different screen sizes
5. **Visual Appeal:** Features gradient animations and clean layout

Technical Components

1. Frontend Structure

- **HTML:** Provides the basic structure of the application
- **CSS:** Handles styling and animations
- **JavaScript:** Implements the interactive functionality

2. Key Files

- `index.html`: Main page structure
- `style.css`: All styling rules
- `script.js`: All JavaScript functionality

3. External Services

- **TheMealDB API:** Free meal database API used to fetch recipe data
- **Font Awesome:** Provides icons used in the interface

Step-by-Step Process

1. User Interaction Flow

1. User enters a meal name in the search box
2. Clicks the search button or presses enter
3. Application fetches matching recipes from TheMealDB API
4. Displays recipe cards in a grid layout
5. User can click any recipe to view detailed information
6. Detailed view appears in a popup modal
7. User can close the modal to return to the recipe grid

2. Code Execution Flow

1. **Search Initiation:**
 - a. Event listener on search button triggers search function
 - b. Input value is validated (not empty)
 - c. `getrecipes()` function is called with search term
2. **API Request:**
 - a. Fetch request sent to TheMealDB API
 - b. Response converted to JSON
 - c. Loading message shown during fetch
3. **Recipe Display:**
 - a. For each recipe in response, create a recipe card
 - b. Each card contains:
 - i. Recipe image
 - ii. Meal name
 - iii. Area/cuisine
 - iv. Category
 - v. "View Recipe" button
4. **Detailed View:**
 - a. Clicking "View Recipe" triggers `openRecipepopup()`
 - b. Popup modal displays:
 - i. Meal name
 - ii. List of ingredients with measurements
 - iii. Preparation instructions
5. **Error Handling:**
 - a. Network errors or empty results show appropriate messages

- b. Input validation prevents empty searches

Key Functions

1. `getrecipes(data)`

- Fetches recipes from API based on search term
- Handles loading state and error messages
- Creates and appends recipe cards to the DOM

2. `fetchIngredients(meal)`

- Processes the meal object to extract ingredients and measurements
- Returns formatted HTML list of ingredients

3. `openRecipepopup(meal)`

- Populates the recipe details modal with:
 - Meal name
 - Ingredients list
 - Preparation instructions
- Displays the modal

4. Event Listeners

- Search button click handler
- Recipe close button handler
- Individual recipe button click handlers

Styling Highlights

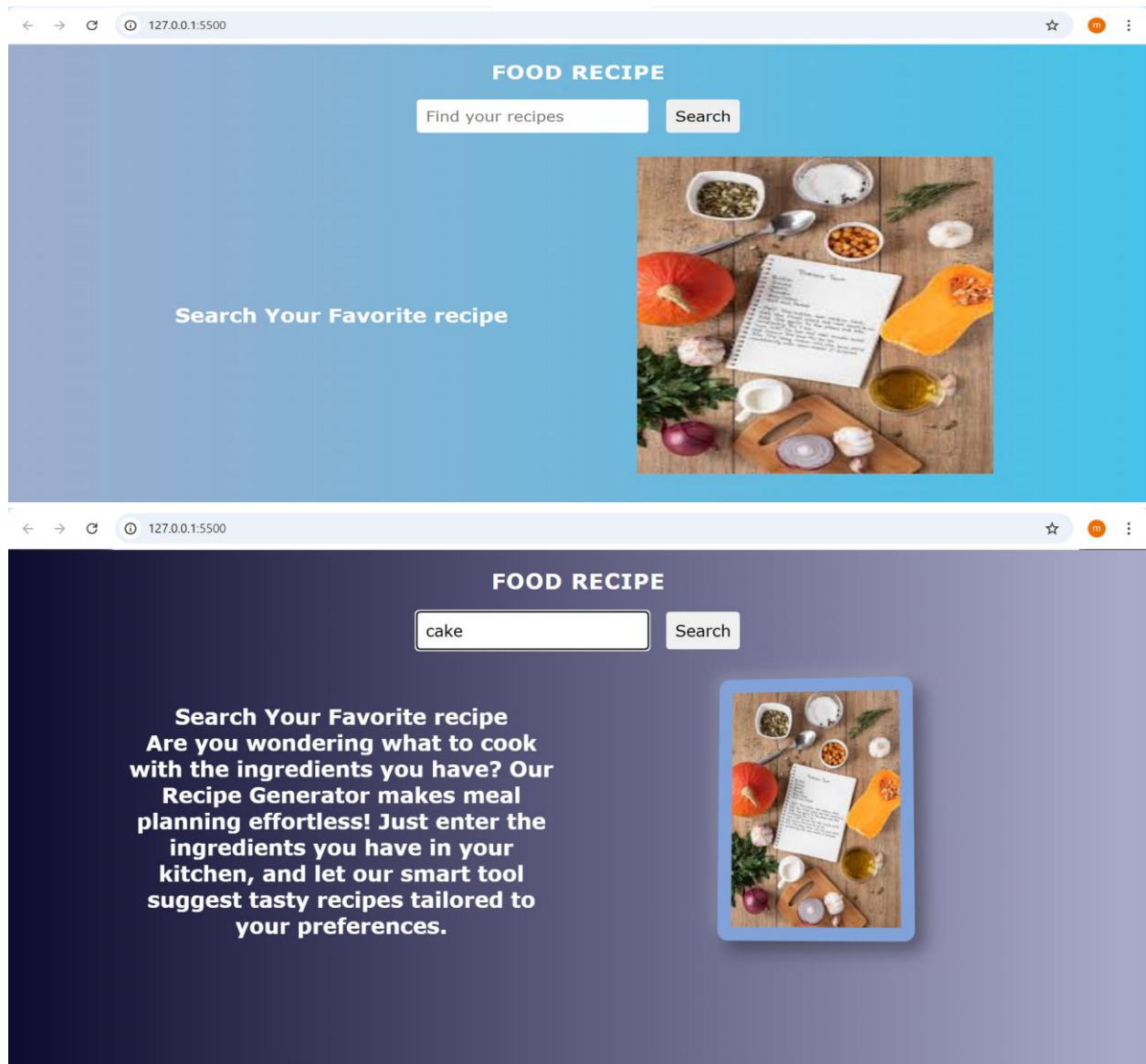
1. **Animated Gradient Background:**
 - a. Smooth gradient animation for visual appeal
 - b. Applied to both body and navigation
2. **Recipe Cards:**
 - a. Clean, card-based layout
 - b. Hover effects for interactivity

- c. Consistent styling
- 3. **Modal Popup:**
 - a. Centered on screen
 - b. Scrollable for long content
 - c. Custom scrollbar styling

Deployment Considerations

1. **API Limitations:**
 - a. TheMealDB has rate limits
 - b. Consider caching responses for better performance
2. **Responsiveness:**
 - a. Currently uses simple responsive grid
 - b. Could be enhanced with media queries
3. **Accessibility:**
 - a. Basic accessibility implemented
 - b. Could be improved with ARIA attributes

Result :



FOOD RECIPE



Pancakes

American - Dish
Belongs to Dessert

[View Recipe](#)

Rock Cakes

British - Dish
Belongs to Dessert

[View Recipe](#)

Carrot Cake

British - Dish
Belongs to Dessert

[View Recipe](#)

Dundee cake

British - Dish
Belongs to Dessert

[View Recipe](#)

Parkin Cake

British - Dish
Belongs to Dessert

[View Recipe](#)

Eccles Cakes

British - Dish
Belongs to Dessert

[View Recipe](#)

FOOD RECIPE



Pancakes

American - Dish
Belongs to Dessert

[View Recipe](#)

ROCK CAKES

Ingredients:

- 225g Self-raising Flour
- 75g Caster Sugar
- 1 tsp Baking Powder
- 125g Butter
- 150g Dried Fruit
- 1 Eggs
- 1 tbs Milk
- 2 tsp Vanilla Extract

Preheat oven to 180C/350F/Gas 4 and line a baking tray with baking parchment.

[View Recipe](#)

Carrot Cake

British - Dish
Belongs to Dessert

[View Recipe](#)

FOOD RECIPE



Pancakes

American - Dish
Belongs to Dessert

[View Recipe](#)

Preheat oven to 180C/350F/Gas 4 and line a baking tray with baking parchment.

Mix the flour, sugar and baking powder in a bowl and rub in the cubed butter until the mixture looks like breadcrumbs, then mix in the dried fruit.

In a clean bowl, beat the egg and milk together with the vanilla extract.

Add the egg mixture to the dry ingredients and stir with a spoon until the mixture just comes together as a thick lump.

[View Recipe](#)

Carrot Cake

British - Dish
Belongs to Dessert

[View Recipe](#)

Conclusion :

A **Recipe Generator** is a powerful tool for anyone looking to simplify meal planning and discover new dishes effortlessly. By entering available ingredients, users can receive personalized recipe suggestions, making cooking both fun and efficient.

With features like **ingredient-based search, quick recipes, healthy options, and step-by-step instructions**, this tool enhances the cooking experience for both beginners and experienced chefs. Whether you're trying to reduce food waste, eat healthier, or just find inspiration for your next meal, a **Recipe Generator** is the perfect kitchen companion.

APPENDIX :

Index.html :

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="style.css">

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.7.2/css/all.min.css" integrity="sha512-Ew84Mr4kqVGRNSglGL/F/aIDqQb7xQ2vcrdlwxfjThSH8CSR7PBEakCr51Ck+w+/U6swU2Im1vVX0SVk9ABhg==" crossorigin="anonymous" />

  <title>FOOD RECIPE GENERATOR</title>

</head>

<body>

  <header>

    <nav>

      <h1>FOOD RECIPE</h1>
```

<form>

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

</form>

</nav>

</header>


```
<main>
```

<section>

<div class="recipe-generator">

Search Your Favorite recipe

Are you wondering what to cook with the ingredients you have? Our Recipe Generator makes meal planning effortless! Just enter the ingredients you have in your kitchen, and let our smart tool suggest tasty recipes tailored to your preferences.



A placeholder for a recipe image, represented by a large rectangular area with a light gray background and a thin black border. The text within this area is a base64-encoded image source string.

</div>

<div class="recipe-details">

<button type="button" class="recipe-close-btn">

<i class="fas fa-times"></i>

</button>

</div>

```
</section>

</main>

<script src="script.js"></script>

</body>

</html>
```

Style.css :

```
*{
  margin: 0;
  padding: 0;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  box-sizing: border-box;
}

@keyframes gradientAnimation {
  0% {
    background-position: 0% 50%;
  }
  50% {
    background-position: 100% 50%;
  }
  100% {
    background-position: 0% 50%;
  }
}

body{
  background-color: #2b1d0f;
  color: #fff;
  height: 100vh;
```

```
        background: linear-gradient(90deg, rgba(2,0,36,1) 0%, rgba(169,169,201,1)
35%, rgba(0,212,255,1) 100%);
        background-size: 300% 300%;
        animation: gradientAnimation 6s ease infinite;
    }
    header nav {

        background: linear-gradient(90deg, rgba(2,0,36,1) 0%, rgba(169,169,201,1)
35%, rgba(0,212,255,1) 100%);
        background-size: 300% 300%;
        animation: gradientAnimation 6s ease infinite;

        padding: 20px;
        display: flex;
        justify-content: space-between;
        align-items: center;
        z-index: 1;
    }
    header nav h1 {
        letter-spacing: 1px;
        font-weight: 800;
    }
    header nav form{
        display: flex;
        justify-content: center;
        margin-top: 20px;
    }
    input{
        flex-grow: 1;
        margin-right: 20px;
    }
    input[type="text"],button[type='submit']{
        border: none;
        font-size: 18px;
        padding: 10px;
        border-radius: 4px;
    }
    .recipe-close-btn:hover,.recipe button:hover,[type='submit']:hover{
        background-color: rgb(148, 191, 48);
    }
```

```
        cursor: pointer;
    }
}
.recipe-generator{
    text-align: center;

    display: grid;
    grid-template-columns: repeat(auto-fit,minmax(300px,1fr));
    grid-gap: 40px;
    width:80%;
    margin:10px auto;
    place-items: center;
}
.recipe{
    background-color: #fff;
    display: flex;
    flex-direction: column;
    color:black;
    border: radius 6px;
    box-shadow: 0 5px 10px rgba(78,73,73,0.1), -5px -5px 10px rgba(34,34,34,0.5);
    cursor: pointer;
    max-width:300px;
    transition: transform 0.3s ease-in-out;
}
.recipe:hover{
    transform: scale(1.02);
}
.recipe img{
    height: 300px;
}
.recipe button{
    font-size:20px;
    padding:10px;
    border-radius: 5px;
    cursor:pointer;
    margin:18px auto;
    border:none;
    background-color: #f44336;
    color: #fff;
}
```

```
}

header nav{
    flex-direction: column;
}

.recipe h3{
    font-size: 24px;
    margin-block: 5px;
}

.recipe p{
    font-size: 20px;
    color: #4a4a4a;
    margin: 5px 0;
}

.recipe-details{
    display: none;
    position: fixed;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    -webkit-transform: translate(-50%, -50%);
    background-color: #694c2f;
    border-radius: 12px;
    width: 40%;
    height: 60%;
    font-size: 20px;
    transition: all 0.5s ease-in-out;
    overflow-y: scroll;
}

.recipe-details::-webkit-scrollbar{
    width: 10px;
}

.recipe-details::-webkit-scrollbar-thumb{
    background: #b5b5ba;
    border-radius: 16px;
    --webkit-border-radius: 16px;
}

.recipe-details-content{
    padding: 30px;
}

.recipe-name{
    text-align: center;
    text-transform: uppercase;
}
```

```
    text-decoration: underline;
}
.recipeName, .ingredientList, .recipeInstructions{
    margin-bottom: 20px;
}
.ingredientList{
    margin-top: 10px;
    margin-left: 20px;
}

.recipeInstructions{
    line-height: 30px;
    white-space: pre-line;
}
.recipe-close-btn{
    border: none;
    font-size: 18px;
    padding: 8px;
    border-radius: 4px;
    background-color: #f44336;
    color: #fff;
    position: absolute;
    top: 20px;
    right: 20px;
    width: 30px;
    height: 30px;
    display: flex;
    justify-content: center;
    align-items: center;
}
.recipe_image{
    display: flex;

    justify-content: center;
    align-items: center;
    background: rgb(130, 163, 217);
    padding: 15px;
    border-radius: 12px;
    box-shadow: 10px 10px 20px rgba(0, 0, 0, 0.4),
                -5px -5px 15px rgba(255, 255, 255, 0.2);
    max-width: 420px;
}
```



```

margin: auto;
transform: perspective(1000px) rotateX(5deg) rotateY(-5deg);
transition: all 0.3s ease-in-out;
cursor:pointer ;
}
.recipe_image:hover {
  cursor: pointer;
  transform: perspective(1000px) rotateX(0deg) rotateY(0deg) scale(1.05); /*
Lifts on hover */
  box-shadow: 15px 15px 30px rgba(0, 0, 0, 0.5),
              -7px -7px 20px rgba(255, 255, 255, 0.3);
}

```

Script.js :

```

const search_input=document.querySelector(".search-input");
const search_btn=document.querySelector(".search-btn");
const recipe_generator=document.querySelector(".recipe-generator");
const recipe_details_content=document.querySelector(".recipe-details-content");
const recipe_close_btn=document.querySelector(".recipe-details");

// const openRecipepopup()

const getrecipes =async (data) => {
  recipe_generator.innerHTML="Fetching recipes.."
  try{
    const dt=await
fetch(`https://www.themealdb.com/api/json/v1/1/search.php?s=${data}`);
    const query = await dt.json();
    recipe_generator.innerHTML=""
    query.meals.forEach(meal => {

      const recipediv=document.createElement('div');
      recipediv.classList.add('recipe');

      recipediv.innerHTML=`
        
        <h3>${meal.strMeal}</h3>
        <p><span>${meal.strArea}</span> - Dish</p>
        <p>Belongs to <span>${meal.strCategory}</span></p>

```

```

        const button=document.createElement('button');
        button.textContent="View Recipe";
        recipediv.appendChild(button);
        button.addEventListener('click',()=>{
            openRecipepopup(meal);
        })
        recipe_generator.appendChild(recipediv);
    });

}
catch (error){
    recipe_generator.innerHTML="Error in Fetching Recipes..!try another"
}

}

const fetchIngredients =(meal)=>{
    let ingredientsList="";
    for(let i=1;i<=20;i++){
        const ingredient = meal[`strIngredient${i}`];
        if(ingredient){
            const measure = meal[`strMeasure${i}`];
            ingredientsList += `<li>${measure} ${ingredient}</li>`
        }
        else{
            break;
        }
    }
    return ingredientsList;
}

const openRecipepopup=(meal)=>{
    recipe_details_content.innerHTML=`
        <h2 class="recipeName">${meal.strMeal}</h2>
        <h3>Ingredients:</h3>
        <ul class="ingredientList">${fetchIngredients(meal)}</ul>
        <div class="recipeInstructions"><h3><p
>${meal.strInstructions}</p></h3></div>

```

```
        recipe_details_content.parentElement.style.display='block'
    }

    recipe_close_btn.addEventListener('click',()=>{
        recipe_details_content.parentElement.style.display="none";
    })

    search_btn.addEventListener('click',(e)=>{
        e.preventDefault();
        const data =search_input.value.trim();
        if(!data){
            recipe_generator.innerHTML=`<h2>Type the Meal in the Search Box.</h2>`
            return
        }
        getrecipes(data);
        console.log("clicked!")
    })
```