PROFESSIONAL TRAINING REPORT

entitled

FOOD RECIPE GENERATOR

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Blockchain Technology

by

Grandhi Harshini

41613009

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A++" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHISALAI,
CHENNAI – 600119

OCTOBER 2023

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with A++ Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai – 600 119
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training is the bonafide work of **Ms. Grandhi Harshini** who carried out the project entitled **"FOOD RECIPE GENERATOR"** under my supervision from June 2023 to October 2023.

Internal Guide
Ms. K. Ishwarya M.Tech(Ph.D)

Head of the Department
Dr. S. VIGNESHWARI, M.E., Ph.D.,

Submitted for Viva voce Examination held on ____________________

Internal Examiner **External Examiner**

DECLARATION

I, **Grandhi Harshini (41613009),** hereby declare that the Professional Training Report-I entitled **“FOOD RECIPE GENERATOR”** done by me under the guidance of **Ms. K. Ishwarya M.Tech(Ph.D),** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering with specialization in Artificial Intelligence.

DATE:

PLACE: **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

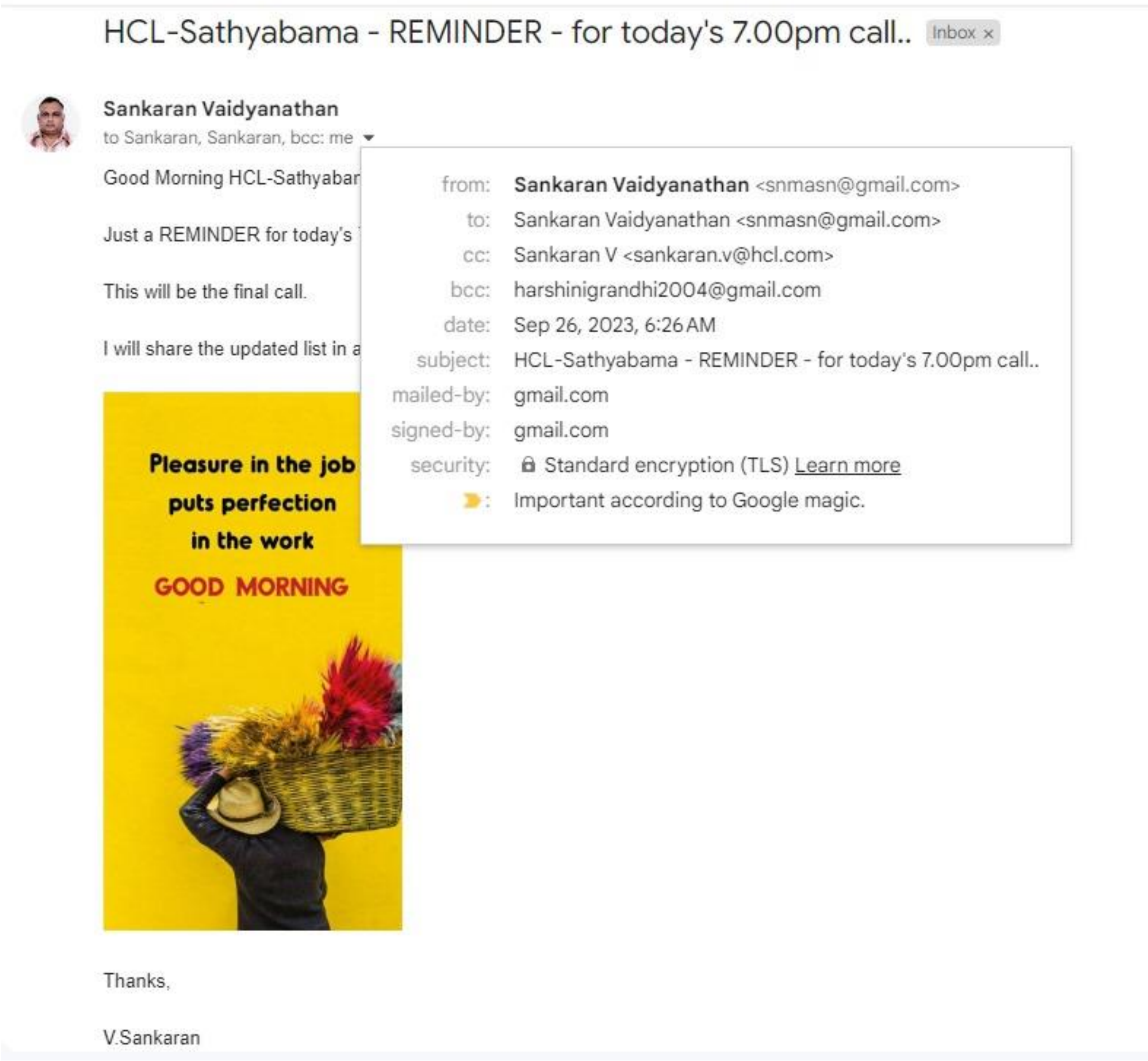
I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D.**, **Dean**, School of Computing, **Dr. S.Vigneshwari M.E., Ph.D., Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Internal Guide **Ms. K. Ishwarya M.Tech(Ph.D)** for his/her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my phase-1 professional Training.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

SAMPLE COURSE CERTIFICATE



HCL-Sathyabama - REMINDER - for today's 7.00pm call.. Inbox ×

Sankaran Vaidyanathan
to Sankaran, Sankaran, bcc: me

Good Morning HCL-Sathyabar

Just a REMINDER for today's 7

This will be the final call.

I will share the updated list in a

from: Sankaran Vaidyanathan <snmasn@gmail.com>
to: Sankaran Vaidyanathan <snmasn@gmail.com>
cc: Sankaran V <sankaran.v@hcl.com>
bcc: harshinigrandhi2004@gmail.com
date: Sep 26, 2023, 6:26 AM
subject: HCL-Sathyabama - REMINDER - for today's 7.00pm call..
mailed-by: gmail.com
signed-by: gmail.com
security: 🔒 Standard encryption (TLS) Learn more
Important according to Google magic.

Thanks,

V.Sankaran

ABSTRACT

The Food Recipe Generator seamlessly blends technology and culinary delight, incorporating HTML, CSS, and JavaScript to create an interactive and visually appealing platform. This innovative project extends its scope with solidity, introducing blockchain to the gastronomic realm. Users can effortlessly generate recipes based on available ingredients, while a sophisticated nutrient calculator ensures a healthy balance. The inclusion of a user-friendly wish list feature allows individuals to curate their culinary aspirations. In a unique twist, users earn exclusive NFTs by adding recipes to their wish list, creating a rewarding experience for both the palate and the digital domain. This multifaceted platform harmonizes simplicity with sophistication, catering to the modern epicurean's craving for both culinary inspiration and technological integration.

Keywords: culinary delight, Food Recipe Generator, HTML, CSS, wish, users, recipes, platform, list, visually, unique, twist

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The project at hand is a comprehensive food recipe generator, seamlessly blending HTML, Bootstrap, CSS, and JavaScript to construct an intuitive user interface. Distinctive features include a sophisticated login mechanism, an intelligent nutrient calculator, a personalized wish list, and a revolutionary recipe generator. To fortify the system's integrity, blockchain technology, implemented through Solidity, underpins the entire architecture. Upon recipe generation, a noteworthy aspect is the automatic addition of the recipe to the blockchain, fostering an immutable and transparent ledger of culinary creations. Users are further empowered with the option to mint NFTs, encapsulating their unique recipes in the blockchain, thus immortalizing their culinary prowess.

1.2 PURPOSE

The purpose of this project transcends the conventional boundaries of recipe repositories. It aspires to create an ecosystem where culinary creativity converges with cutting-edge technology. By seamlessly integrating blockchain, the project ensures the utmost security, transparency, and authenticity in the representation of culinary endeavours.

The user-friendly interface, fortified by Bootstrap and other technologies, makes culinary exploration accessible to individuals with varying degrees of expertise. The nutrient calculator adds a health-conscious dimension, while the wish list and NFT minting options introduce a personalized and immutable aspect to the culinary journey. In essence, this project is a testament to the harmonious coexistence of gastronomic creativity and technological advancement. It heralds a new era where each recipe is not merely a set of instructions but an indelible entry in a decentralized ledger, showcasing the evolution of culinary artistry in the digital age.

CHAPTER 2

LITERATURE REVIEW

In the vast landscape of technological integration, the amalgamation of blockchain technology with the artistry of culinary creation represents a frontier that is still largely unexplored in contemporary literature. While the existing body of knowledge extensively examines the decentralized and secure aspects of blockchain in financial transactions and supply chain management, there exists a discernible void when it comes to the intersection of blockchain and gastronomy. This literature survey aims to illuminate the transformative potential of this convergence and delineate the pioneering nature of the current project.

Current literature predominantly delves into the mechanics of blockchain, emphasizing its role in ensuring trust and transparency in diverse sectors. However, the culinary landscape, with its rich tapestry of flavours, techniques, and cultural nuances, has yet to fully harness the potential benefits of decentralized ledger systems. This project, therefore, endeavours to bridge this gap by applying blockchain principles to the realm of recipe generation, fostering an environment of heightened security, transparency, and collaborative knowledge sharing.

An examination of the existing literature reveals a paucity of discourse on the fusion of blockchain and culinary arts. This scarcity is indicative of a broader trend in research, where the creative dimensions of various fields remain somewhat divorced from the transformative capacities of blockchain technology. In essence, the literature survey underscores the innovative nature of this project, positioning it as a pioneering effort that extends beyond the current boundaries of interdisciplinary exploration.

The novelty of this project lies in its synthesis of user-friendly interfaces and cutting-edge blockchain technology, creating a space where culinary enthusiasts can not only explore and share recipes but also contribute to a secure and immutable ledger of gastronomic creativity. The literature survey, therefore, serves as a precursor to understanding the uncharted territories this project navigates, highlighting the unique contribution it makes to the evolving discourse on blockchain technology.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 OBJECTIVE OF THE PROJECT

The primary objective of this project is to create a user-centric food recipe generator with enhanced security features through blockchain integration. The system aims to provide a seamless experience for users to explore, create, and share recipes while ensuring the integrity and transparency of the culinary knowledge stored in the blockchain. The key goals include implementing a user-friendly interface, a nutrient calculator for health-conscious choices, a wish list for personalized experiences, and a recipe generator that leverages user inputs. Additionally, the integration of Solidity-based blockchain technology aims to secure the data and enable users to mint NFTs for their unique culinary creations.

3.2 REQUIREMENTS

3.2.1 *HARDWARE REQUIREMENTS*

The hardware infrastructure for this project is designed to accommodate a scalable and responsive web application. The hardware requirements include:

Server Infrastructure: A robust server capable of handling user requests, database interactions, and blockchain transactions efficiently. The server should have sufficient processing power, RAM, and storage to support concurrent user interactions.

Blockchain Node: To integrate blockchain functionality, a dedicated node is required. This node will be responsible for executing Smart Contracts written in Solidity, ensuring the security and transparency of the data stored in the blockchain.

-**Secure Storage**: As the project involves sensitive user data and blockchain interactions, a secure storage solution is essential to safeguard user information, recipes, and blockchain-related data.

3.2.2 *SOFTWARE REQUIREMENTS*

The software requirements encompass the technologies and platforms necessary for the development, deployment, and functionality of the food recipe generator. These include:

Frontend Technologies:

- HTML, CSS, and Bootstrap for creating an intuitive and responsive user interface.
- JavaScript for dynamic and interactive user experiences.
- Web3.js library for interacting with the Ethereum blockchain.

Backend Technologies:

- Node.js or another suitable backend framework for server-side logic and API integration.
- Database Management System (e.g., MySQL, MongoDB) for storing user data, recipes, and related information.

Blockchain Integration:

- Solidity for developing Smart Contracts that govern blockchain interactions.
- Ethereum or a compatible blockchain network for deploying the Smart Contracts and recording recipe transactions.

Security Measures:

- Secure Socket Layer (SSL) for encrypted communication between the user and the server.
- Authentication and authorization mechanisms to control user access and protect sensitive data.
- Regular security audits and updates to mitigate potential vulnerabilities.

Development Tools:

- Code editor (e.g., Visual Studio Code) for writing and managing code.
- Version control system (e.g., Git) for collaborative development and code versioning.

Ganache: A personal blockchain for Ethereum development, Ganache is essential for local testing and simulation of blockchain transactions.

Git:

- Version control is facilitated through Git, allowing collaborative development, tracking changes, and maintaining a comprehensive history of the project.

Node.js:

- The project leverages Node.js for server-side scripting, ensuring efficient and scalable handling of backend processes.

Docker Files:

- Docker is employed for containerization, encapsulating the application and its dependencies into containers for seamless deployment across various environments.

Truffle:

- Truffle is a development framework for Ethereum, aiding in the streamlined deployment of smart contracts and facilitating interactions with the blockchain.

Blockchain Integration (Solidity):

- The use of Solidity, a programming language for writing smart contracts on the Ethereum blockchain, is fundamental to achieving the project's goal of immutability and transparency.

By meeting these hardware and software requirements, the project aims to deliver a robust, secure, and user-friendly food recipe generator with blockchain integration.

CHAPTER 4

DESIGN DESCRIPTION OF PROPOSED PROJECT

4.1 EXISTING METHODOLOGY

The current culinary landscape lacks a cohesive digital platform that seamlessly integrates recipe generation, user interaction, and the robust security afforded by blockchain technology. Traditional recipe platforms often rely on centralized databases, presenting challenges such as data vulnerabilities, limited user engagement features, and a lack of transparency in the recipe-sharing process. Without the integration of blockchain, these systems are susceptible to data tampering, and the authenticity of culinary creations remains uncertain.

4.2 PROPOSED SYSTEM

The proposed system is a revolutionary advancement in the domain of culinary technology, bridging the existing gaps and setting new standards for user engagement, security, and transparency. Key features of the proposed system include:

1. Blockchain Integration:

- The implementation of blockchain technology using Solidity ensures a decentralized and tamper-proof ledger for all recipes. Each culinary creation becomes an immutable entry in the blockchain, guaranteeing authenticity and preserving the creative process.

2. User-Friendly Interface:

- The system boasts a user-centric design with HTML, CSS, Bootstrap, and JavaScript, offering an intuitive interface for users of all levels of culinary expertise. Navigation is seamless, promoting a delightful user experience.

3. Nutrient Calculator:

- A sophisticated nutrient calculator is incorporated to cater to health-conscious users. This feature enables users to make informed choices about their culinary creations, aligning with diverse dietary preferences and requirements.

4. Wish List:

- The introduction of a personalized wish list adds a layer of customization. Users can curate a collection of favorite recipes, creating a tailored culinary journey within the platform.

5. Recipe Generator:

- The heart of the system lies in its dynamic recipe generator. Leveraging user inputs, this feature generates personalized recipes, encouraging experimentation and exploration in the kitchen.

6. NFT Minting:

- A distinctive offering is the ability for users to mint NFTs of their recipes. This feature elevates culinary creations to the realm of digital art, allowing users to showcase and share their unique recipes as non-fungible tokens on the blockchain.

The proposed system, with its amalgamation of cutting-edge technology and user-centric design, addresses the limitations of the existing culinary landscape. By leveraging blockchain, the project not only secures and authenticates culinary creations but also transforms the act of sharing recipes into a decentralized and transparent experience. The culmination of these features positions the proposed system as a pioneering force in the convergence of gastronomy and technology, promising a paradigm shift in how we approach, share, and safeguard our culinary knowledge.

4.1.1 *Ideation Map/System Architecture*

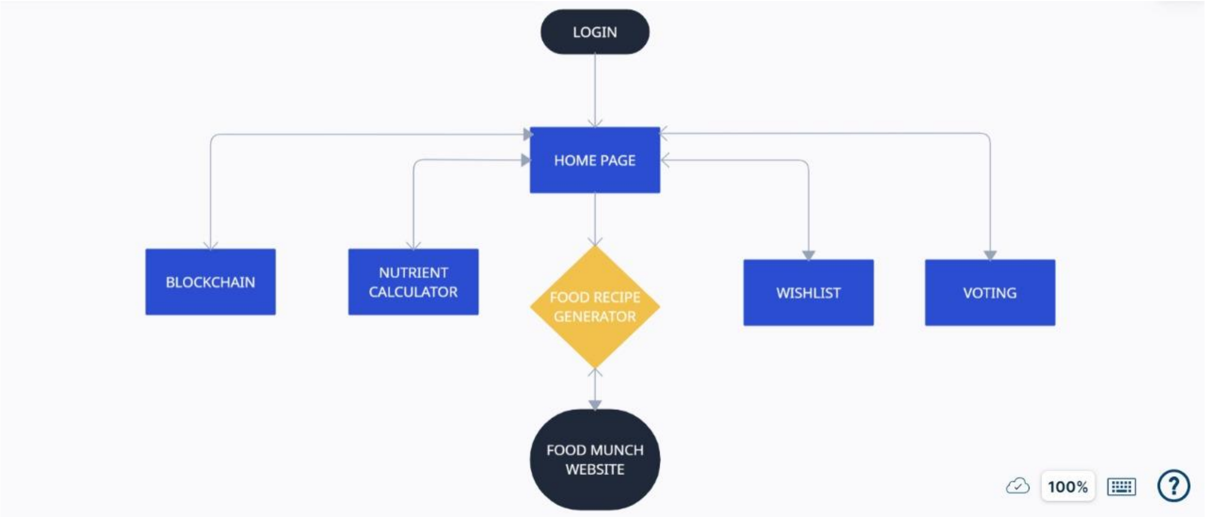


Fig. 1 Conceptual representation

4.1.2 *Various Stages*

The development of the CV Builder project involves several crucial stages, each contributing to its functionality and user experience:

1. Conceptualization and Research: This initial phase focuses on brainstorming ideas and conducting market research. Identifying user needs, understanding market trends, and studying competitor platforms help define the project's scope and features.

2. Planning and Requirement Analysis: Detailed planning involves outlining project objectives, defining user stories, and specifying technical requirements. Clear objectives and a deep understanding of user needs form the basis for subsequent development.

3. Design and Prototyping: User interface (UI) and user experience (UX) are designed during this stage. Wireframes and prototypes are created, providing a visual representation of the platform's layout, functionality, and user interactions.

4. Frontend and Backend Development: The frontend, using HTML, CSS, and JavaScript, focuses on creating the platform's visual elements and user interactions. Backend development involves server-side scripting, database integration, and ensures data management and security.

5. Blockchain Integration: The integration of Solidity smart contracts and Web3.js facilitates blockchain interactions. Storing CV data securely on the blockchain ensures data immutability, enhancing security and credibility.

6. User Authentication and Security: Implementing robust user authentication mechanisms and encryption protocols ensures user data security. Protection against common web vulnerabilities, such as SQL injection and XSS attacks, strengthens the platform's security.

7. CV Creation Interface: Developing an intuitive drag-and-drop interface allows users to create and customize their CVs seamlessly. Real-time previews enhance the user experience, enabling instant feedback and adjustments.

8. CV Printing Functionality: Allowing users to request high-quality prints directly from the platform enhances the practicality of the service, bridging the digital and physical aspects of resume creation.

9. NFT Minting Feature: Enabling users to mint their CVs as Non-Fungible Tokens (NFTs) adds a unique dimension, enhancing the CV's value and authenticity as a digital asset.

1**0. User Testing and Feedback:** Rigorous testing, including user acceptance testing, identifies potential issues and ensures the platform's functionality and usability. User feedback is invaluable, leading to iterative improvements.

11. Deployment and Launch: After successful testing and refinements, the platform is deployed to the chosen hosting environment, making it accessible to users. A well-orchestrated launch strategy ensures a smooth introduction to the user base.

By progressing through these stages meticulously, the CV Builder project achieves its objectives: providing a secure, user-friendly, and innovative platform for creating, validating, and printing professional resumes while pioneering the integration of blockchain technology in the field of career services.

4.1.3 *Internal or Component design structure*

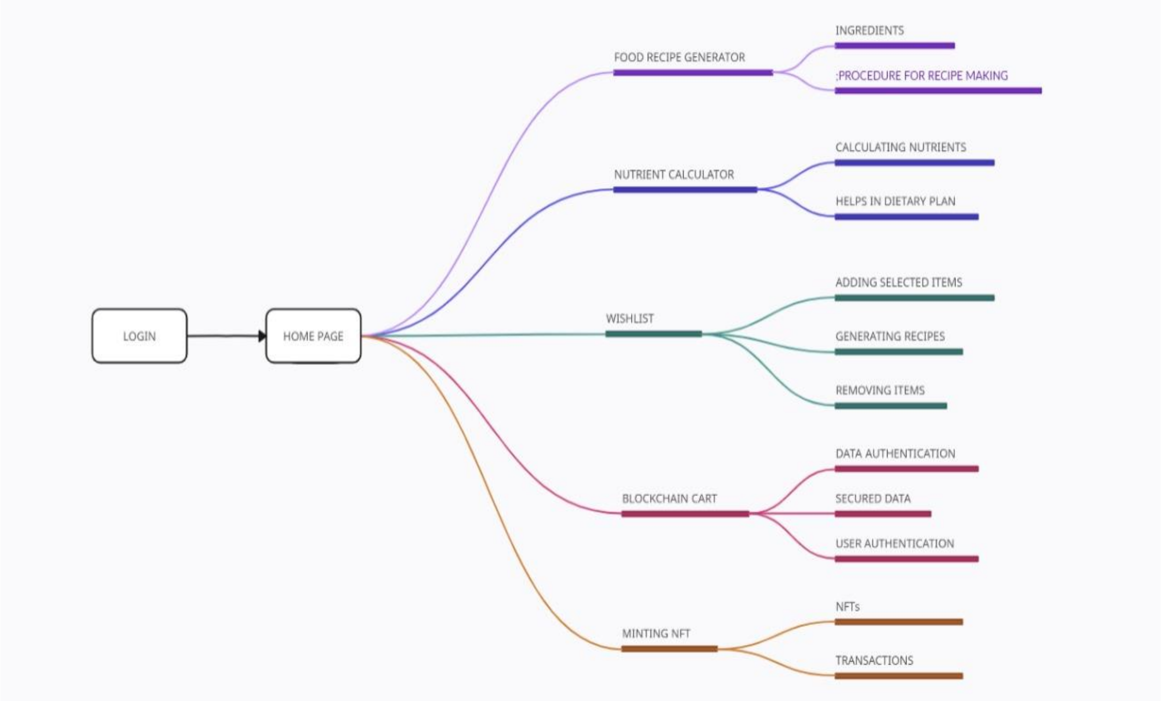


Fig 2. Structure of the website

4.1.4 *working principles*

The food recipe generator operates on principles of user-centricity and innovation. Through an intuitive interface, users can create accounts and curate personalized wish lists. Advanced algorithms power the platform to generate customized recipes based on user preferences. A built-in nutrient calculator promotes nutritional awareness. The project integrates Solidity smart contracts and blockchain technology, ensuring the security and immutability of recipes. Users can mint NFTs for their favorite recipes, enhancing their value and authenticity. The transparent blockchain ledger records every recipe and NFT, fostering trust and traceability. This amalgamation of user personalization, culinary expertise, nutritional insights, and blockchain security forms the core principles, delivering a sophisticated and trustworthy culinary experience to users.

4.2 FEATURES

- **User-Friendly Interface:** The platform offers an intuitive and visually appealing user interface, ensuring an effortless experience for users. Employing HTML, Bootstrap, CSS, and JavaScript, the design prioritizes accessibility and ease of use.

- **Authentication and Personalization:** The inclusion of a login feature enables users to create accounts, enhancing personalization. Users can maintain a wish list, storing preferred recipes for future reference.

- **Nutrient Calculator:** The platform integrates a nutrient calculator, allowing users to analyze the nutritional content of recipes. This feature empowers users to make informed dietary choices.

- **Recipe Generation**: Through user inputs and preferences, JavaScript powers the recipe generator. Dynamic algorithms create recipes tailored to individual requests, ensuring a personalized culinary experience.

- **Blockchain Integration:** Utilizing Solidity smart contracts, the platform integrates blockchain technology for enhanced security. Each generated recipe is cryptographically hashed and permanently stored on the blockchain. This ensures data integrity, providing an immutable record of recipes.

- **NFT Minting:** Users are given the choice to mint NFTs representing their favorite recipes. This groundbreaking feature adds a layer of ownership and uniqueness to recipes, establishing them as valuable digital assets.

- **Blockchain Security:** The integration of blockchain technology ensures robust security. Smart contracts validate transactions, securing the platform against tampering and unauthorized access.

- **Transparent Record-Keeping:** Every generated recipe and NFT creation is transparently recorded on the blockchain. This transparent ledger enhances trust, enabling users to trace the origin and ownership of recipes.

By incorporating these principles, the project delivers a seamless, secure, and personalized culinary experience, while pioneering the integration of blockchain and NFTs in the realm of food recipe generation.

4.2.1 *Novelty of the proposal*

The novelty of our food recipe generator project lies in its intersection of culinary innovation, user empowerment, and cutting-edge blockchain technology. Unlike conventional recipe platforms, our project offers a deeply personalized experience, allowing users to curate their culinary journey through wish lists and tailored recipe suggestions. What sets us apart is the integration of blockchain and NFTs. By leveraging Solidity smart contracts, every recipe is cryptographically secured, ensuring data integrity and security. The ability to mint NFTs for recipes transforms them into exclusive digital assets, a pioneering concept in the culinary realm, enhancing their uniqueness and value. Moreover, our platform promotes nutritional awareness through an embedded nutrient calculator, aligning with the growing emphasis on health-conscious dining. Additionally, the transparent and immutable blockchain ledger serves as a testament to the authenticity of recipes, instilling trust among users. This amalgamation of personalized culinary exploration, blockchain-enhanced security, and the concept of recipe NFTs makes our project a groundbreaking venture in the gastronomic landscape, redefining how individuals engage with recipes and explore the world of flavors with confidence and sophistication.

CHAPTER 5

CONCLUSION

In the culinary tapestry of innovation and technology, our Food Recipe Generator emerges as a transformative masterpiece. With a user-friendly interface and personalized features, it redefines how culinary enthusiasts explore recipes, ensuring a seamless and delightful experience for users of all backgrounds. Our project isn't just about cooking; it's a journey into healthier lifestyles. The integration of a nutrient calculator empowers users to make informed dietary choices, enhancing their well-being.

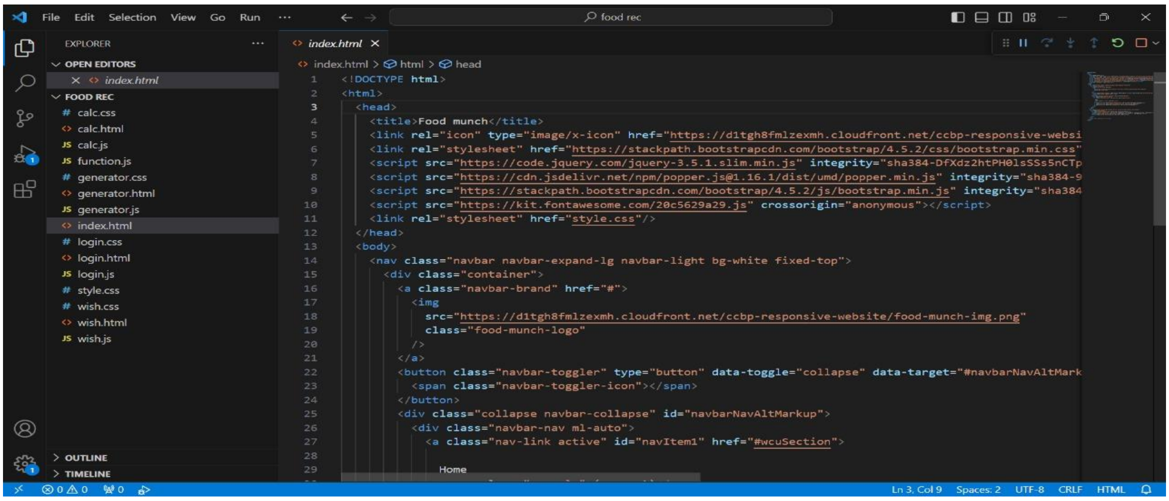
What truly sets our project apart is the integration of blockchain technology and NFTs. Each recipe, meticulously crafted, isn't just a digital entity but a unique, verifiable asset. Through blockchain's security and NFT minting, we ensure the authenticity and provenance of every recipe, fostering trust and confidence among users. In essence, our Food Recipe Generator isn't just a digital platform; it's a culinary companion, an innovative amalgamation of flavours and code, marking the future of gastronomic exploration.

In this digital culinary landscape, we've not only created a recipe generator but an immersive experience, where every click is a step into the heart of digital gastronomy. Our project stands as a testament to creativity, collaboration, and a shared passion for the art of cooking. It's not just a platform; it's a flavour-filled journey, leaving a lasting impact on how we approach, share, and savour recipes in the digital age.

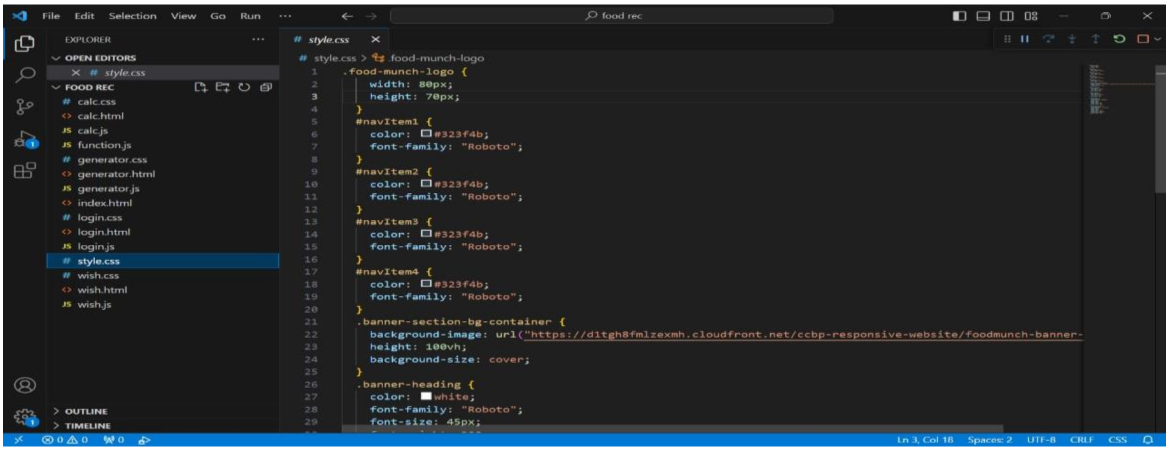
APPENDIX

A. SOURCE CODE

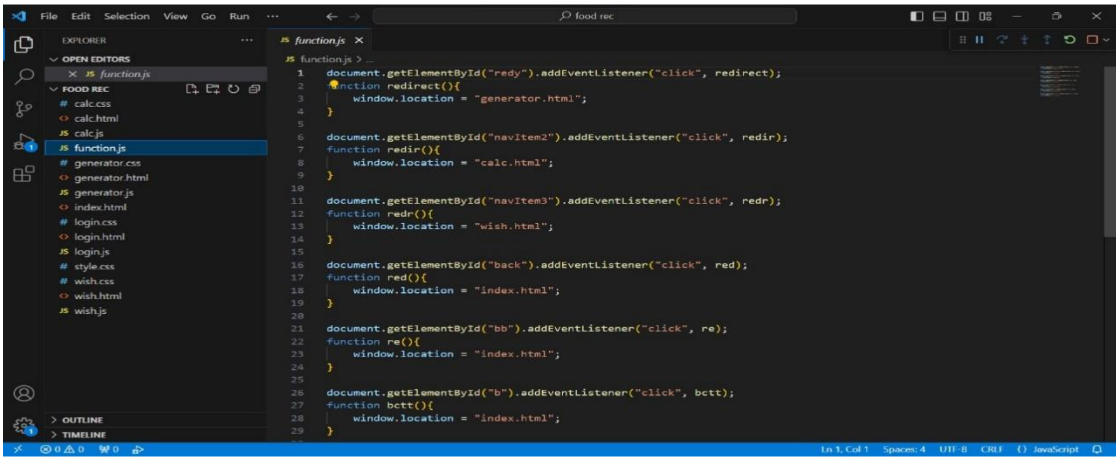
HOME PAGE



```
<!DOCTYPE html>
<html>
  <head>
    <title>Food munch</title>
    <link rel="icon" type="image/x-icon" href="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-responsive-websi
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-DfXdz2htPH0lsSSs5nCTp
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-9
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384
    <script src="https://kit.fontawesome.com/20c5629a29.js" crossorigin="anonymous"></script>
    <link rel="stylesheet" href="style.css"/>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-light bg-white fixed-top">
      <div class="container">
        <a class="navbar-brand" href="#">
          <img
            src="https://d1tgh8fmlzexmh.cloudfront.net/ccbp-responsive-website/food-munch-img.png"
            class="food-munch-logo"
          />
        </a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavAltMark
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
          <div class="navbar-nav ml-auto">
            <a class="nav-link active" id="navItem1" href="#wcuSection">

              Home
```

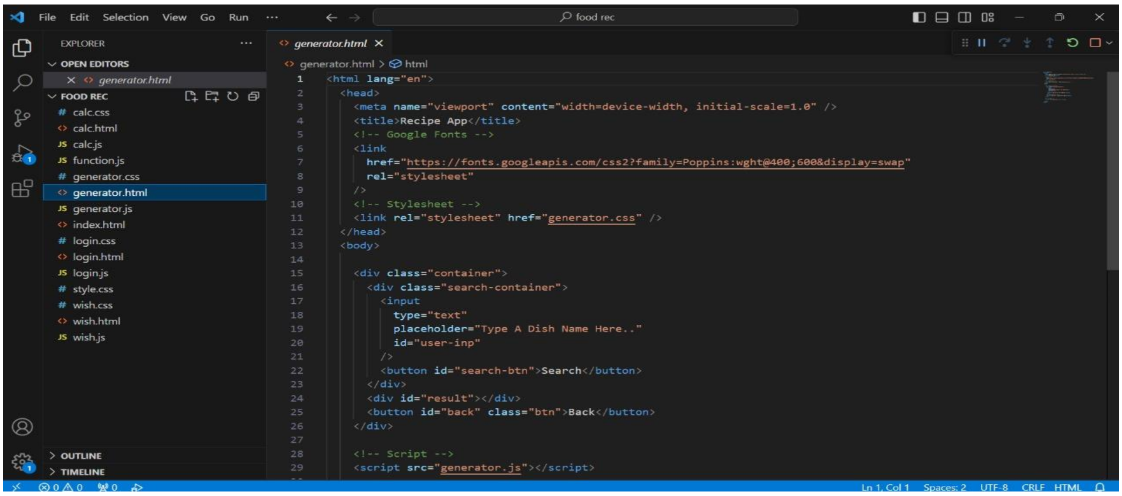


```
.food-munch-logo {
  width: 80px;
  height: 70px;
}
#navItem1 {
  color: #323f4b;
  font-family: "Roboto";
}
#navItem2 {
  color: #323f4b;
  font-family: "Roboto";
}
#navItem3 {
  color: #323f4b;
  font-family: "Roboto";
}
#navItem4 {
  color: #323f4b;
  font-family: "Roboto";
}
.banner-section-bg-container {
  background-image: url("https://d1tgh8fmlzexmh.cloudfront.net/ccbp-responsive-website/foodmunch-banner-
  height: 100vh;
  background-size: cover;
}
.banner-heading {
  color: white;
  font-family: "Roboto";
  font-size: 45px;
```

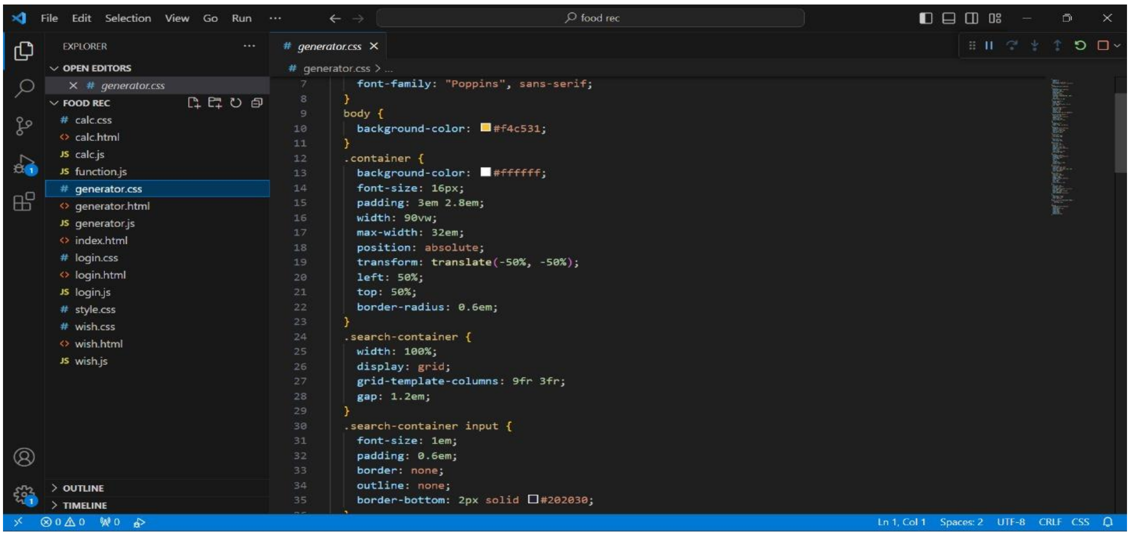


```
document.getElementById("redy").addEventListener("click", redirect);
function redirect(){
    window.location = "generator.html";
}

document.getElementById("navItem2").addEventListener("click", redir);
function redir(){
    window.location = "calc.html";
}

document.getElementById("navItem3").addEventListener("click", redr);
function redr(){
    window.location = "wish.html";
}

document.getElementById("back").addEventListener("click", red);
function red(){
    window.location = "index.html";
}

document.getElementById("bb").addEventListener("click", re);
function re(){
    window.location = "index.html";
}

document.getElementById("b").addEventListener("click", bctt);
function bctt(){
    window.location = "index.html";
}
```

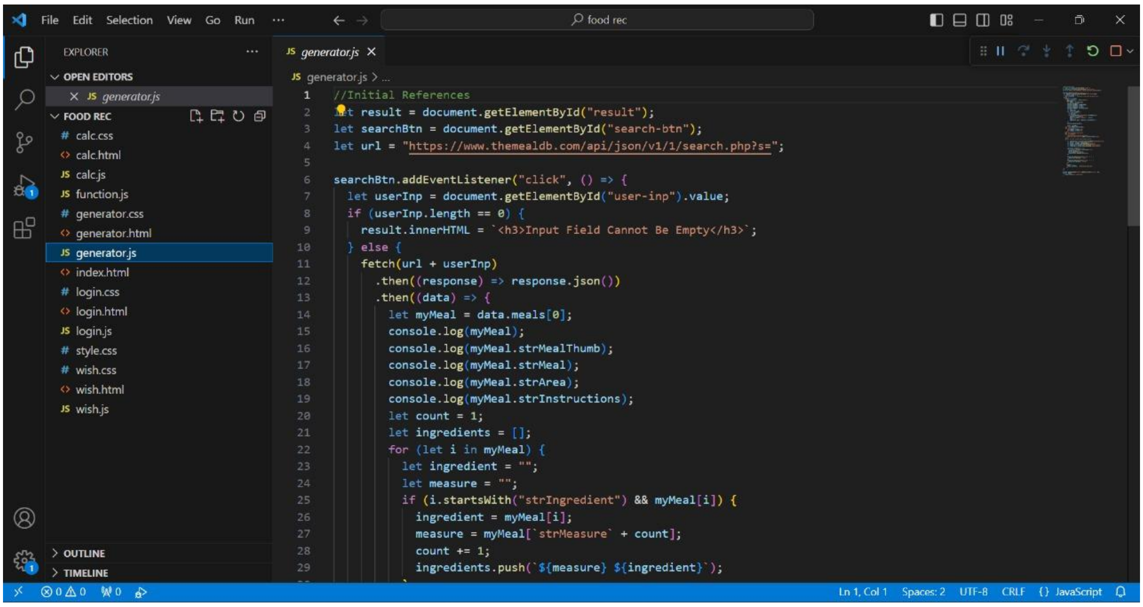
RECIPE GENERATOR PAGE



```
<html lang="en">
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Recipe App</title>
    <!-- Google Fonts -->
    <link
      href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap"
      rel="stylesheet"
    />
    <!-- Stylesheet -->
    <link rel="stylesheet" href="generator.css" />
  </head>
  <body>

    <div class="container">
      <div class="search-container">
        <input
          type="text"
          placeholder="Type A Dish Name Here.."
          id="user-inp"
        />
        <button id="search-btn">Search</button>
      </div>
      <div id="result"></div>
      <button id="back" class="btn">Back</button>
    </div>

    <!-- Script -->
    <script src="generator.js"></script>
```

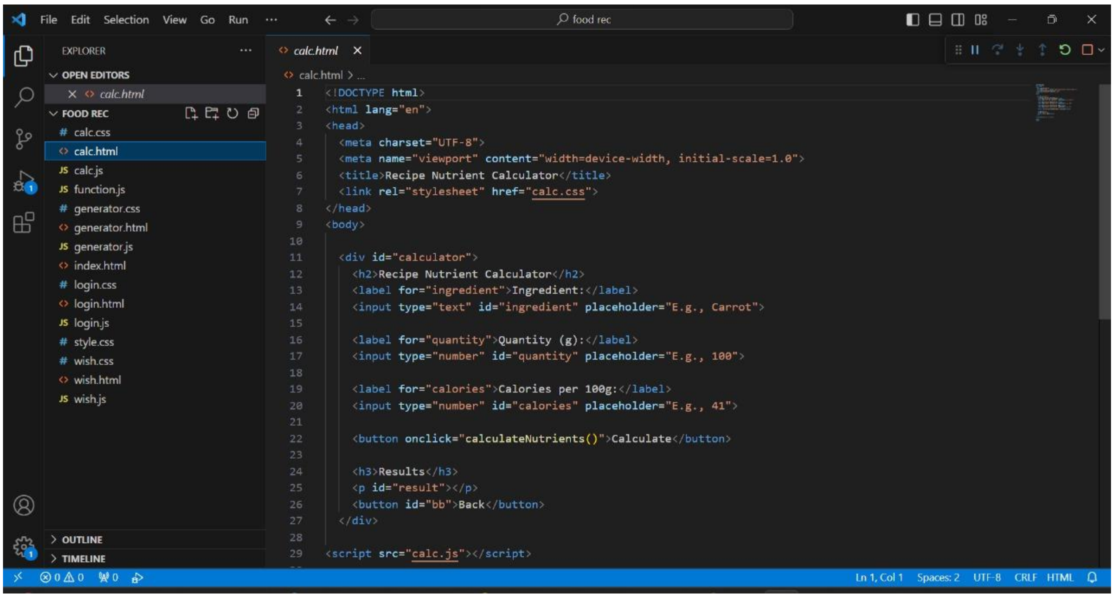


```
  font-family: "Poppins", sans-serif;
}
body {
  background-color: #f4c531;
}
.container {
  background-color: #ffffff;
  font-size: 16px;
  padding: 3em 2.8em;
  width: 90vw;
  max-width: 32em;
  position: absolute;
  transform: translate(-50%, -50%);
  left: 50%;
  top: 50%;
  border-radius: 0.6em;
}
.search-container {
  width: 100%;
  display: grid;
  grid-template-columns: 9fr 3fr;
  gap: 1.2em;
}
.search-container input {
  font-size: 1em;
  padding: 0.6em;
  border: none;
  outline: none;
  border-bottom: 2px solid #202030;
```

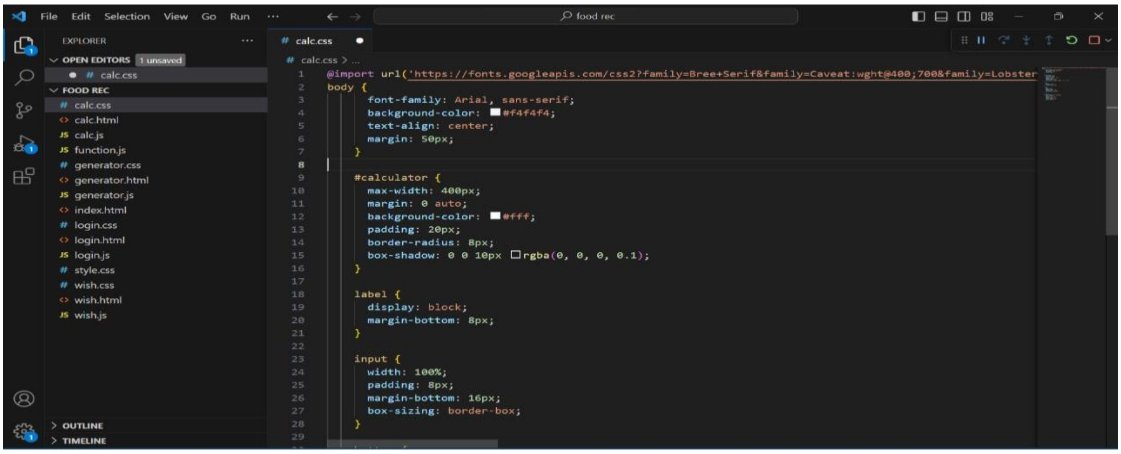


```javascript
//Initial References
let result = document.getElementById("result");
let searchBtn = document.getElementById("search-btn");
let url = "https://www.themealdb.com/api/json/v1/1/search.php?s=";

searchBtn.addEventListener("click", () => {
  let userInp = document.getElementById("user-inp").value;
  if (userInp.length == 0) {
    result.innerHTML = `<h3>Input Field Cannot Be Empty</h3>`;
  } else {
    fetch(url + userInp)
      .then((response) => response.json())
      .then((data) => {
        let myMeal = data.meals[0];
        console.log(myMeal);
        console.log(myMeal.strMealThumb);
        console.log(myMeal.strMeal);
        console.log(myMeal.strArea);
        console.log(myMeal.strInstructions);
        let count = 1;
        let ingredients = [];
        for (let i in myMeal) {
          let ingredient = "";
          let measure = "";
          if (i.startsWith("strIngredient") && myMeal[i]) {
            ingredient = myMeal[i];
            measure = myMeal[`strMeasure` + count];
            count += 1;
            ingredients.push(`${measure} ${ingredient}`);
```

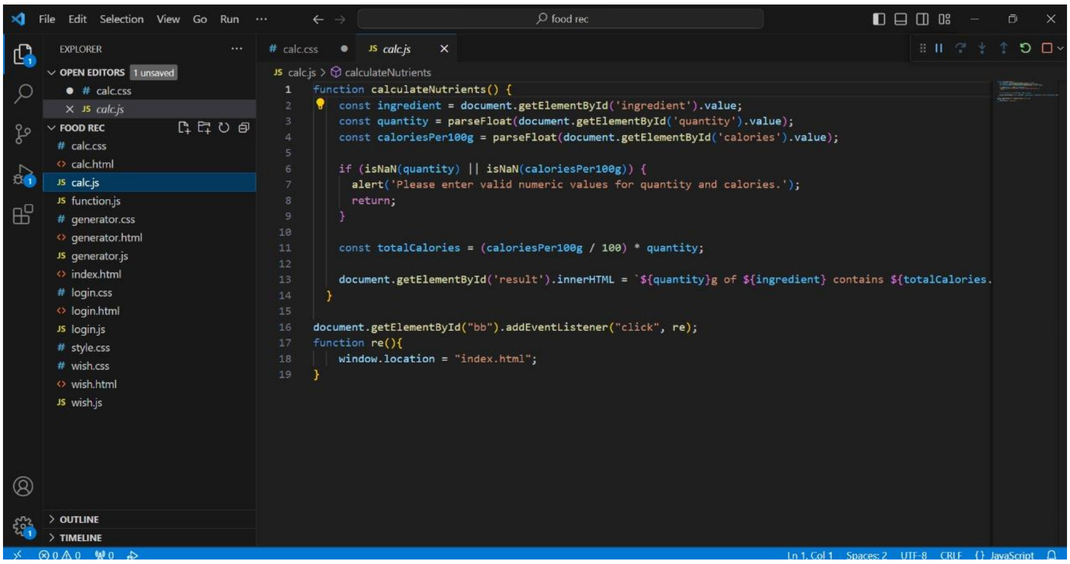
NUTRIENT CALCULATOR PAGE



```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Recipe Nutrient Calculator</title>
  <link rel="stylesheet" href="calc.css">
</head>
<body>

  <div id="calculator">
    <h2>Recipe Nutrient Calculator</h2>
    <label for="ingredient">Ingredient:</label>
    <input type="text" id="ingredient" placeholder="E.g., Carrot">

    <label for="quantity">Quantity (g):</label>
    <input type="number" id="quantity" placeholder="E.g., 100">

    <label for="calories">Calories per 100g:</label>
    <input type="number" id="calories" placeholder="E.g., 41">

    <button onclick="calculateNutrients()">Calculate</button>

    <h3>Results</h3>
    <p id="result"></p>
    <button id="bb">Back</button>
  </div>

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

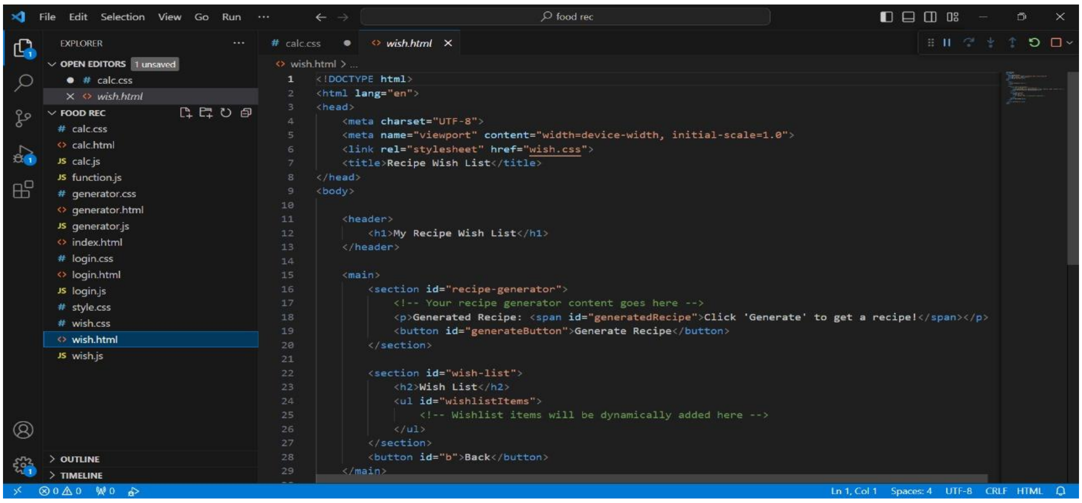


```css
@import url('https://fonts.googleapis.com/css2?family=Bree+Serif&family=Caveat:wght@400;700&family=Lobster
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    text-align: center;
    margin: 50px;
  }

  #calculator {
    max-width: 400px;
    margin: 0 auto;
    background-color: #fff;
    padding: 20px;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }

  label {
    display: block;
    margin-bottom: 8px;
  }

  input {
    width: 100%;
    padding: 8px;
    margin-bottom: 16px;
    box-sizing: border-box;
  }
```

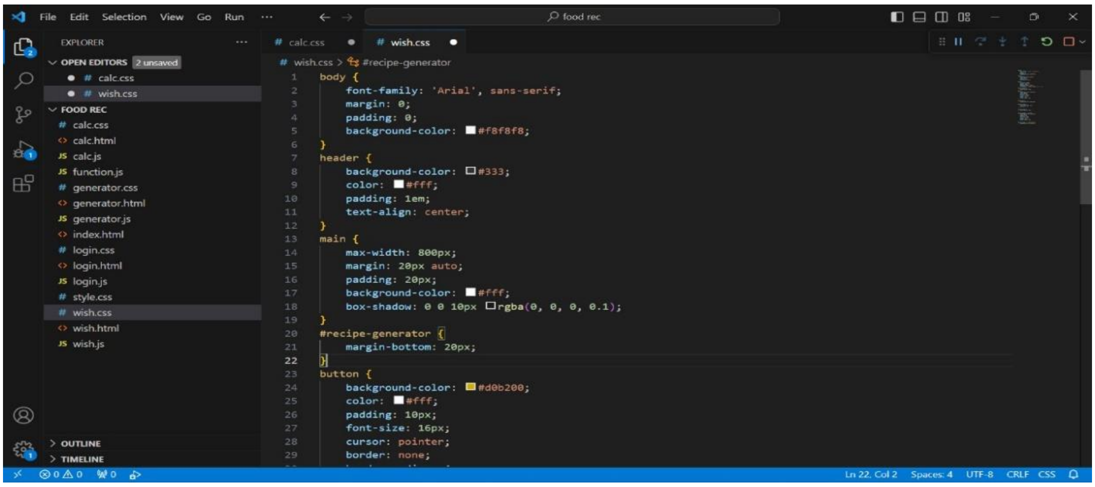


```javascript
function calculateNutrients() {
    const ingredient = document.getElementById('ingredient').value;
    const quantity = parseFloat(document.getElementById('quantity').value);
    const caloriesPer100g = parseFloat(document.getElementById('calories').value);

    if (isNaN(quantity) || isNaN(caloriesPer100g)) {
      alert('Please enter valid numeric values for quantity and calories.');
      return;
    }

    const totalCalories = (caloriesPer100g / 100) * quantity;

    document.getElementById('result').innerHTML = `${quantity}g of ${ingredient} contains ${totalCalories.
  }

document.getElementById("bb").addEventListener("click", re);
function re(){
    window.location = "index.html";
}
```

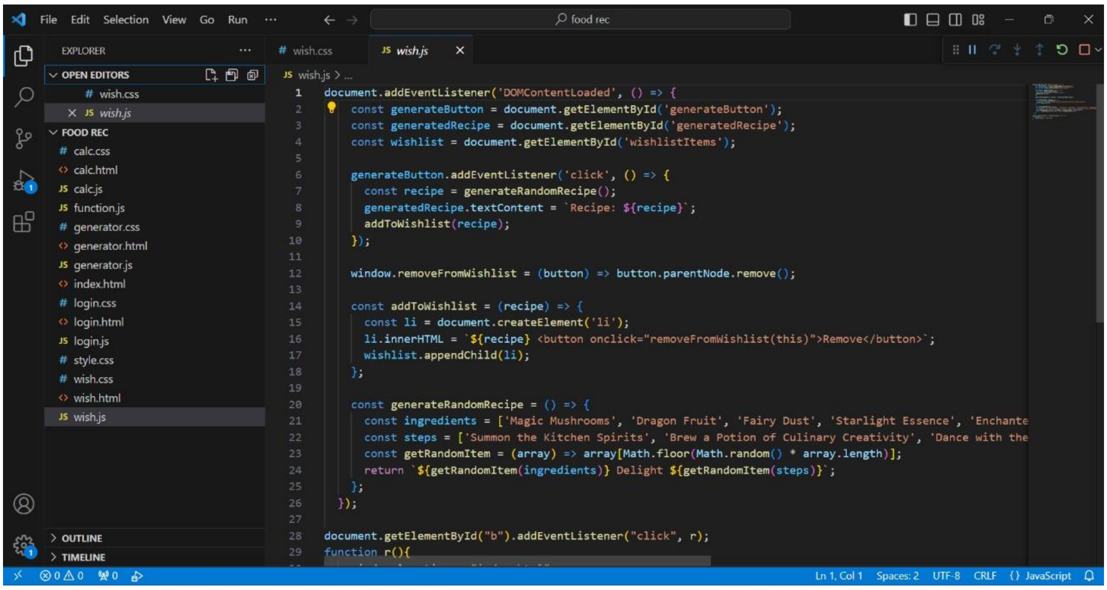
WISHLIST PAGE



```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="wish.css">
    <title>Recipe Wish List</title>
</head>
<body>

    <header>
        <h1>My Recipe Wish List</h1>
    </header>

    <main>
        <section id="recipe-generator">
            <!-- Your recipe generator content goes here -->
            <p>Generated Recipe: <span id="generatedRecipe">Click 'Generate' to get a recipe!</span></p>
            <button id="generateButton">Generate Recipe</button>
        </section>

        <section id="wish-list">
            <h2>Wish List</h2>
            <ul id="wishlistItems">
                <!-- Wishlist items will be dynamically added here -->
            </ul>
        </section>
        <button id="b">Back</button>
    </main>
```

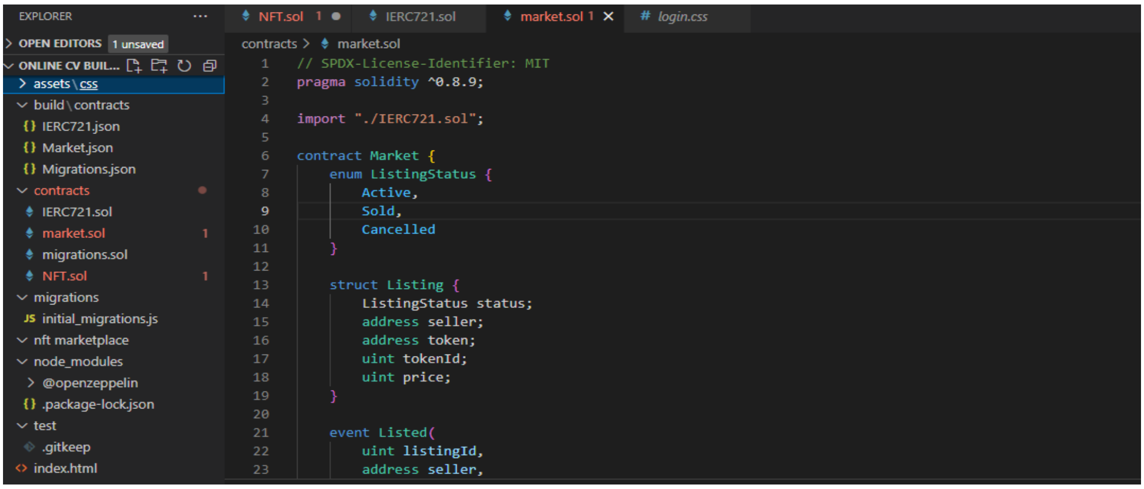


```css
body {
    font-family: 'Arial', sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f8f8f8;
}
header {
    background-color: #333;
    color: #fff;
    padding: 1em;
    text-align: center;
}
main {
    max-width: 800px;
    margin: 20px auto;
    padding: 20px;
    background-color: #fff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
#recipe-generator {
    margin-bottom: 20px;
}
button {
    background-color: #d0b200;
    color: #fff;
    padding: 10px;
    font-size: 16px;
    cursor: pointer;
    border: none;
```

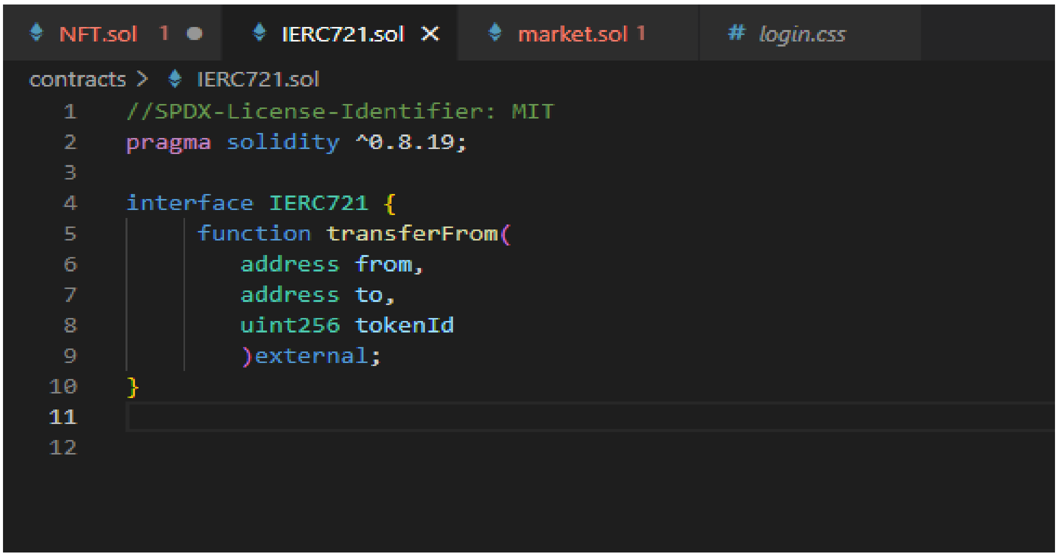


```javascript
document.addEventListener('DOMContentLoaded', () => {
  const generateButton = document.getElementById('generateButton');
  const generatedRecipe = document.getElementById('generatedRecipe');
  const wishlist = document.getElementById('wishlistItems');

  generateButton.addEventListener('click', () => {
    const recipe = generateRandomRecipe();
    generatedRecipe.textContent = `Recipe: ${recipe}`;
    addToWishlist(recipe);
  });

  window.removeFromWishlist = (button) => button.parentNode.remove();

  const addToWishlist = (recipe) => {
    const li = document.createElement('li');
    li.innerHTML = `${recipe} <button onclick="removeFromWishlist(this)">Remove</button>`;
    wishlist.appendChild(li);
  };

  const generateRandomRecipe = () => {
    const ingredients = ['Magic Mushrooms', 'Dragon Fruit', 'Fairy Dust', 'Starlight Essence', 'Enchante
    const steps = ['Summon the Kitchen Spirits', 'Brew a Potion of Culinary Creativity', 'Dance with the
    const getRandomItem = (array) => array[Math.floor(Math.random() * array.length)];
    return `${getRandomItem(ingredients)} Delight ${getRandomItem(steps)}`;
  };
});

document.getElementById("b").addEventListener("click", r);
function r(){
```

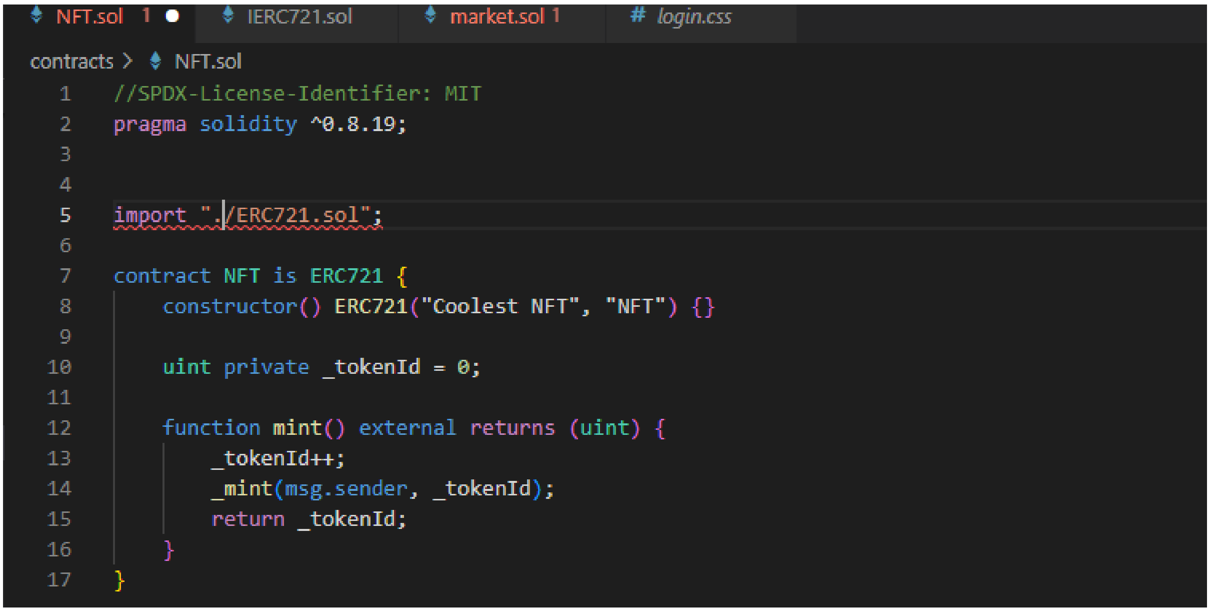
BLOCKCHAIN PAGE



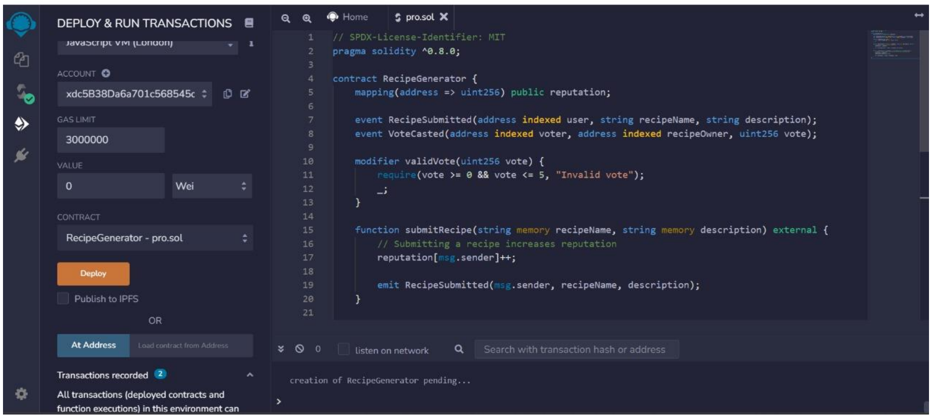
```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

import "./IERC721.sol";

contract Market {
    enum ListingStatus {
        Active,
        Sold,
        Cancelled
    }

    struct Listing {
        ListingStatus status;
        address seller;
        address token;
        uint tokenId;
        uint price;
    }

    event Listed(
        uint listingId,
        address seller,
```



```
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

interface IERC721 {
    function transferFrom(
        address from,
        address to,
        uint256 tokenId
        )external;
}

```



```
//SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;


import "./ERC721.sol";

contract NFT is ERC721 {
    constructor() ERC721("Coolest NFT", "NFT") {}

    uint private _tokenId = 0;

    function mint() external returns (uint) {
        _tokenId++;
        _mint(msg.sender, _tokenId);
        return _tokenId;
    }
}
```


DEPLOY & RUN TRANSACTIONS
Deploy
OR
At Address
Load contract from Address
Home
pro.sol
listen on network
Search with transaction hash or address

JavaScript VM (London)

ACCOUNT xdc5B38Da6a701c568545c

GAS LIMIT 3000000

VALUE 0 Wei

CONTRACT RecipeGenerator - pro.sol

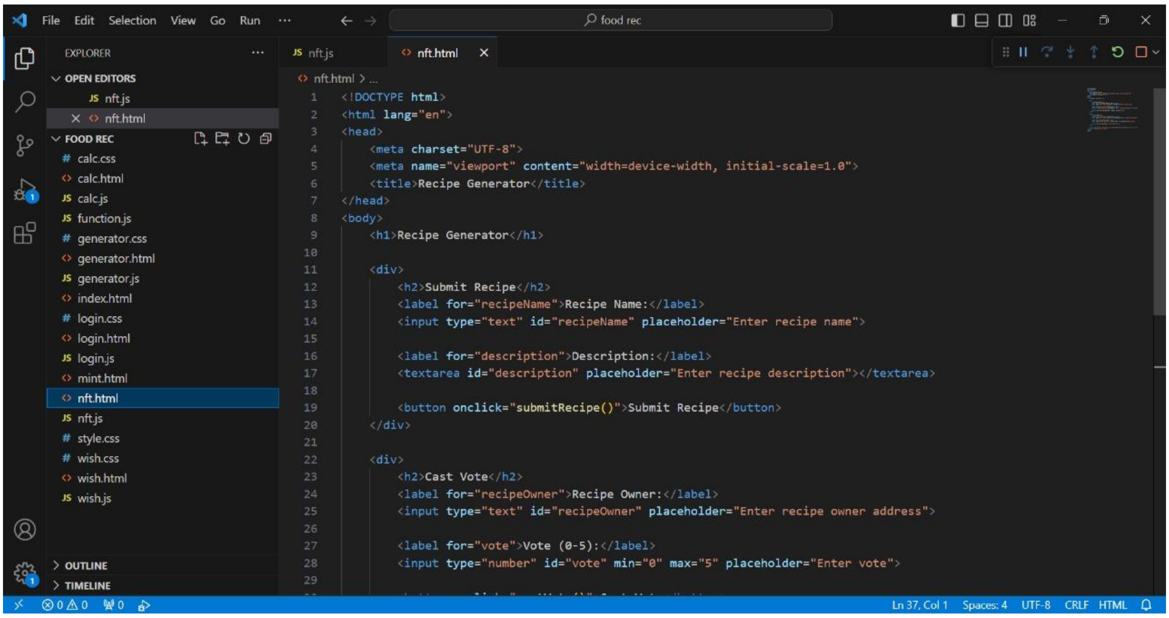
□ Publish to IPFS

Transactions recorded 2

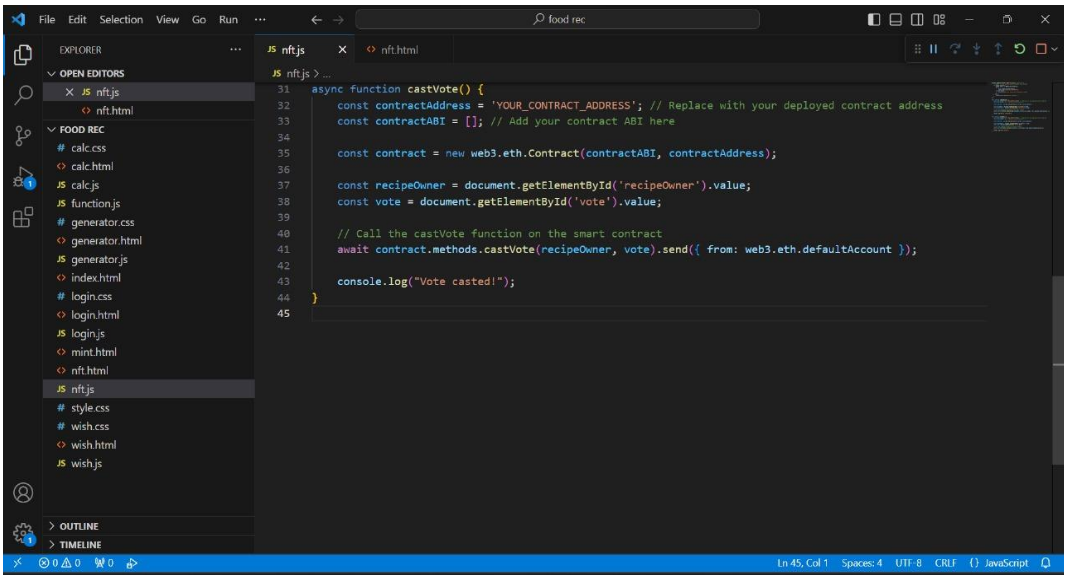
All transactions (deployed contracts and function executions) in this environment can

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract RecipeGenerator {
    mapping(address => uint256) public reputation;

    event RecipeSubmitted(address indexed user, string recipeName, string description);
    event VoteCasted(address indexed voter, address indexed recipeOwner, uint256 vote);

    modifier validVote(uint256 vote) {
        require(vote >= 0 && vote <= 5, "Invalid vote");
        _;
    }

    function submitRecipe(string memory recipeName, string memory description) external {
        // Submitting a recipe increases reputation
        reputation[msg.sender]++;

        emit RecipeSubmitted(msg.sender, recipeName, description);
    }

```

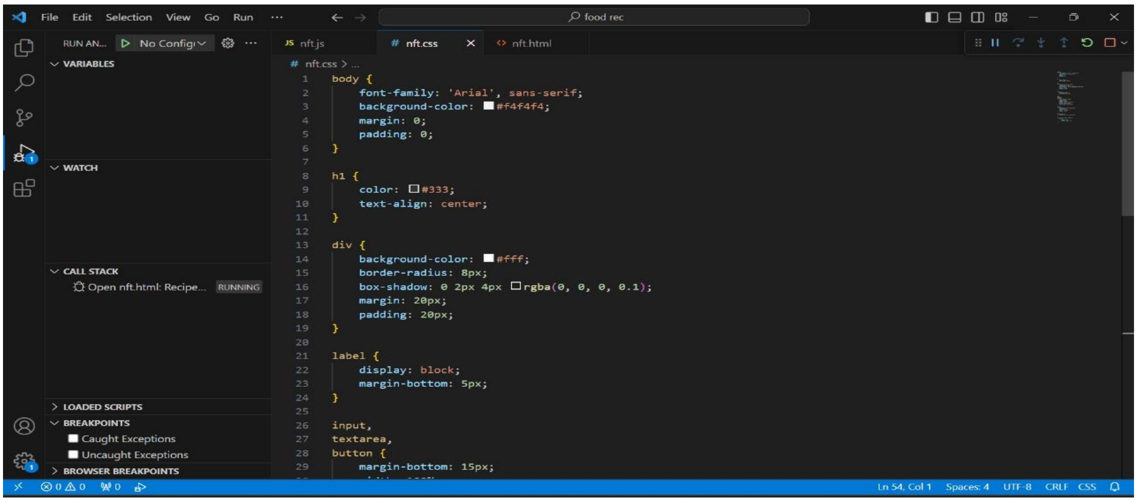
```
creation of RecipeGenerator pending...
```



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Recipe Generator</title>
</head>
<body>
    <h1>Recipe Generator</h1>

    <div>
        <h2>Submit Recipe</h2>
        <label for="recipeName">Recipe Name:</label>
        <input type="text" id="recipeName" placeholder="Enter recipe name">

        <label for="description">Description:</label>
        <textarea id="description" placeholder="Enter recipe description"></textarea>

        <button onclick="submitRecipe()">Submit Recipe</button>
    </div>

    <div>
        <h2>Cast Vote</h2>
        <label for="recipeOwner">Recipe Owner:</label>
        <input type="text" id="recipeOwner" placeholder="Enter recipe owner address">

        <label for="vote">Vote (0-5):</label>
        <input type="number" id="vote" min="0" max="5" placeholder="Enter vote">

```

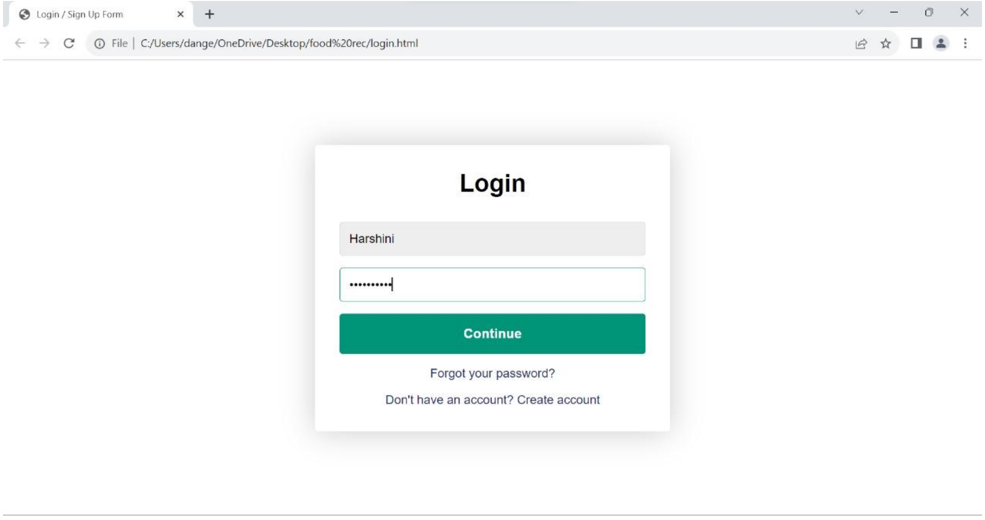


```javascript
async function castVote() {
    const contractAddress = 'YOUR_CONTRACT_ADDRESS'; // Replace with your deployed contract address
    const contractABI = []; // Add your contract ABI here

    const contract = new web3.eth.Contract(contractABI, contractAddress);

    const recipeOwner = document.getElementById('recipeOwner').value;
    const vote = document.getElementById('vote').value;

    // Call the castVote function on the smart contract
    await contract.methods.castVote(recipeOwner, vote).send({ from: web3.eth.defaultAccount });

    console.log("Vote casted!");
}

```



```css
body {
    font-family: 'Arial', sans-serif;
    background-color: ■#f4f4f4;
    margin: 0;
    padding: 0;
}

h1 {
    color: □#333;
    text-align: center;
}

div {
    background-color: ■#fff;
    border-radius: 8px;
    box-shadow: 0 2px 4px □rgba(0, 0, 0, 0.1);
    margin: 20px;
    padding: 20px;
}

label {
    display: block;
    margin-bottom: 5px;
}

input,
textarea,
button {
    margin-bottom: 15px;
```

B. OUTPUT SCREENSHOTS



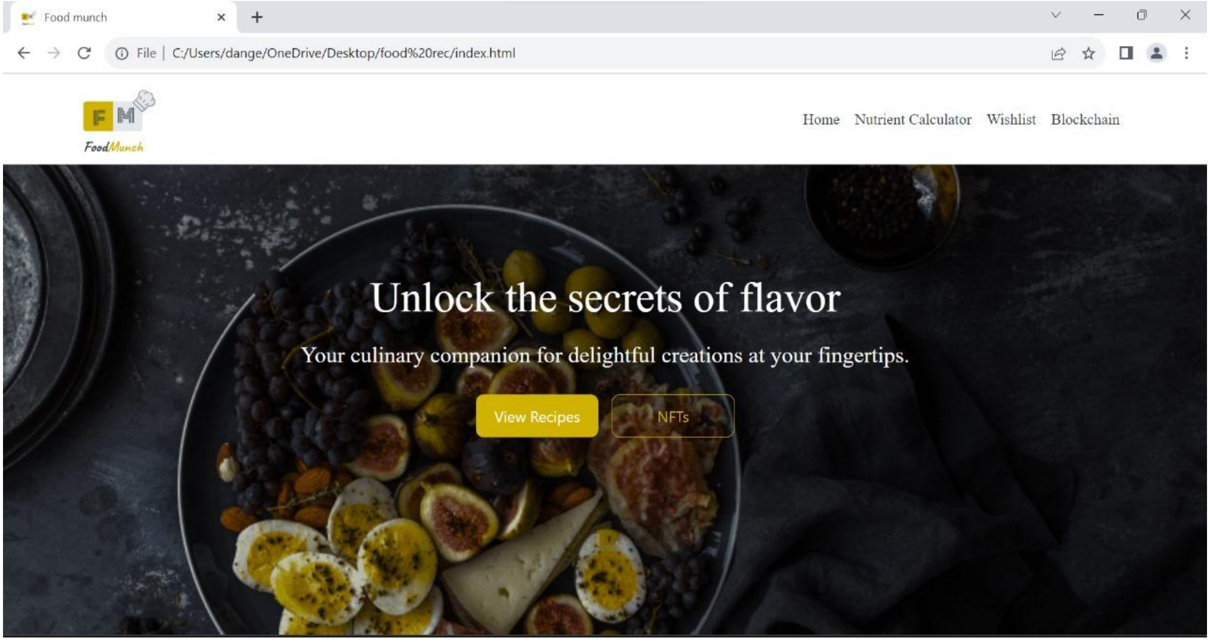
Login / Sign Up Form

File | C:/Users/dange/OneDrive/Desktop/food%20rec/login.html

Login

Harshini

••••••••••

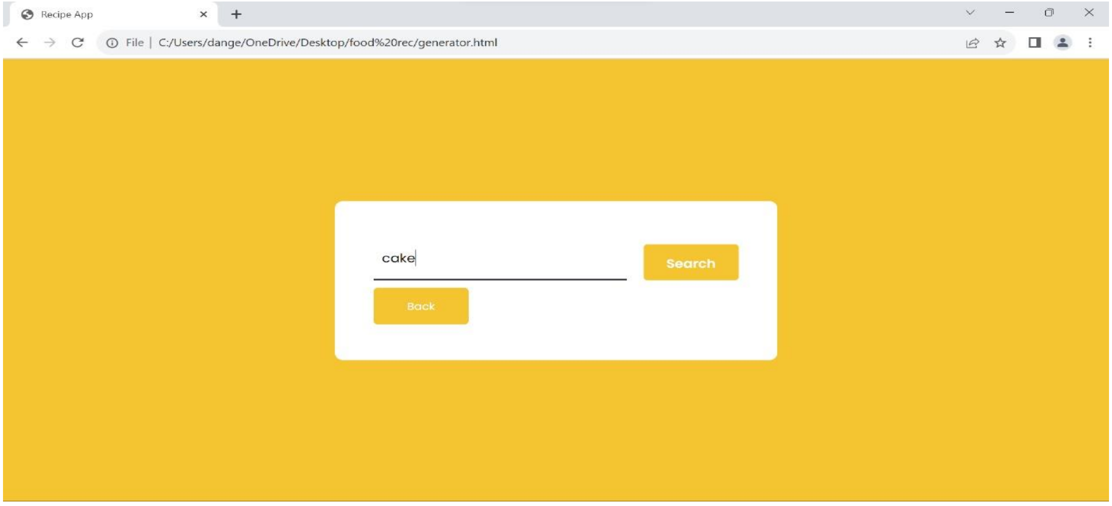


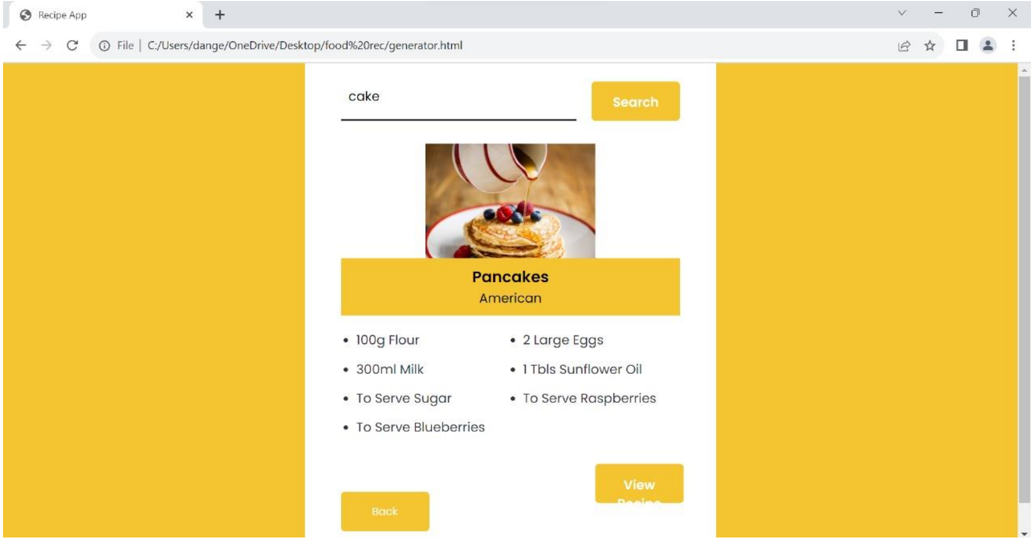
Food munch

File | C:/Users/dange/OneDrive/Desktop/food%20rec/index.html

Unlock the secrets of flavor

Your culinary companion for delightful creations at your fingertips.

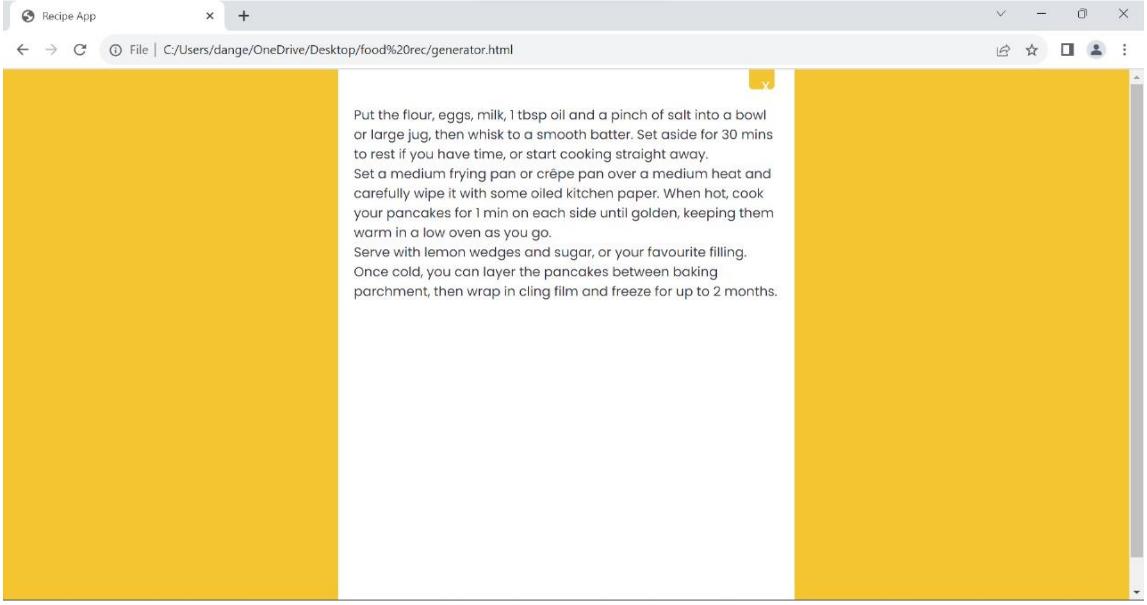

Recipe App
File | C:/Users/dange/OneDrive/Desktop/food%20rec/generator.html
cake
Search
Back


Recipe App
File | C:/Users/dange/OneDrive/Desktop/food%20rec/generator.html
cake
Search

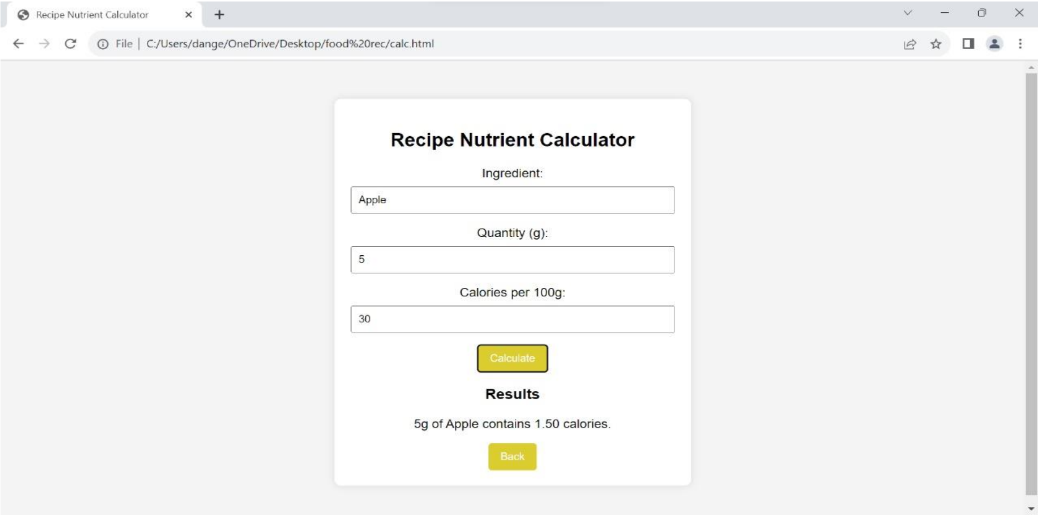
Pancakes

American

- 100g Flour
- 2 Large Eggs
- 300ml Milk
- 1 Tbls Sunflower Oil
- To Serve Sugar
- To Serve Raspberries
- To Serve Blueberries


Recipe App
File | C:/Users/dange/OneDrive/Desktop/food%20rec/generator.html
x

Put the flour, eggs, milk, 1 tbsp oil and a pinch of salt into a bowl or large jug, then whisk to a smooth batter. Set aside for 30 mins to rest if you have time, or start cooking straight away.
Set a medium frying pan or crêpe pan over a medium heat and carefully wipe it with some oiled kitchen paper. When hot, cook your pancakes for 1 min on each side until golden, keeping them warm in a low oven as you go.
Serve with lemon wedges and sugar, or your favourite filling. Once cold, you can layer the pancakes between baking parchment, then wrap in cling film and freeze for up to 2 months.


Recipe Nutrient Calculator
File | C:/Users/dange/OneDrive/Desktop/food%20rec/calc.html
Calculate
Back

Recipe Nutrient Calculator

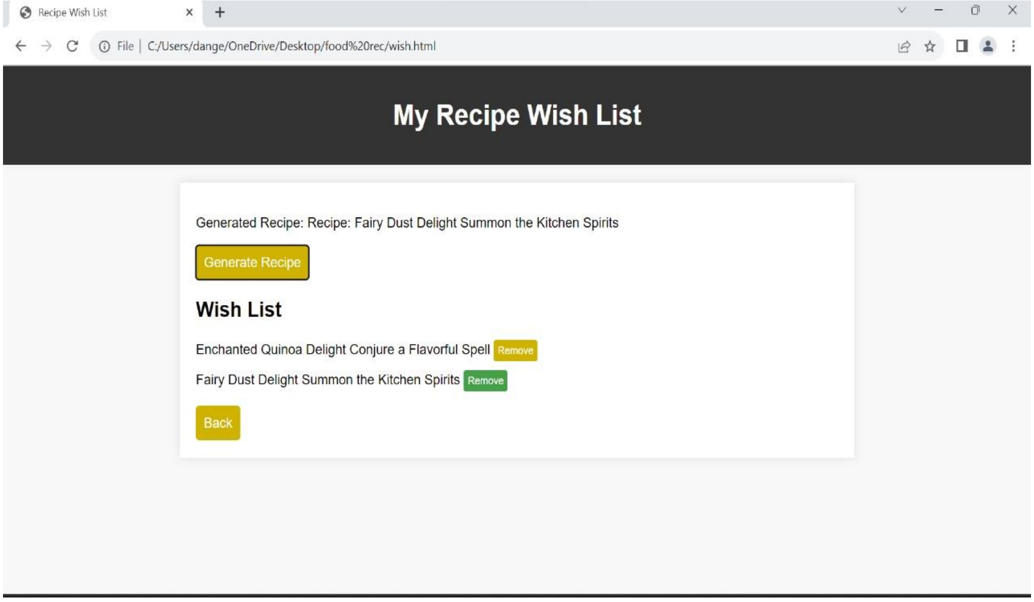
Ingredient: Apple

Quantity (g): 5

Calories per 100g: 30

Results

5g of Apple contains 1.50 calories.



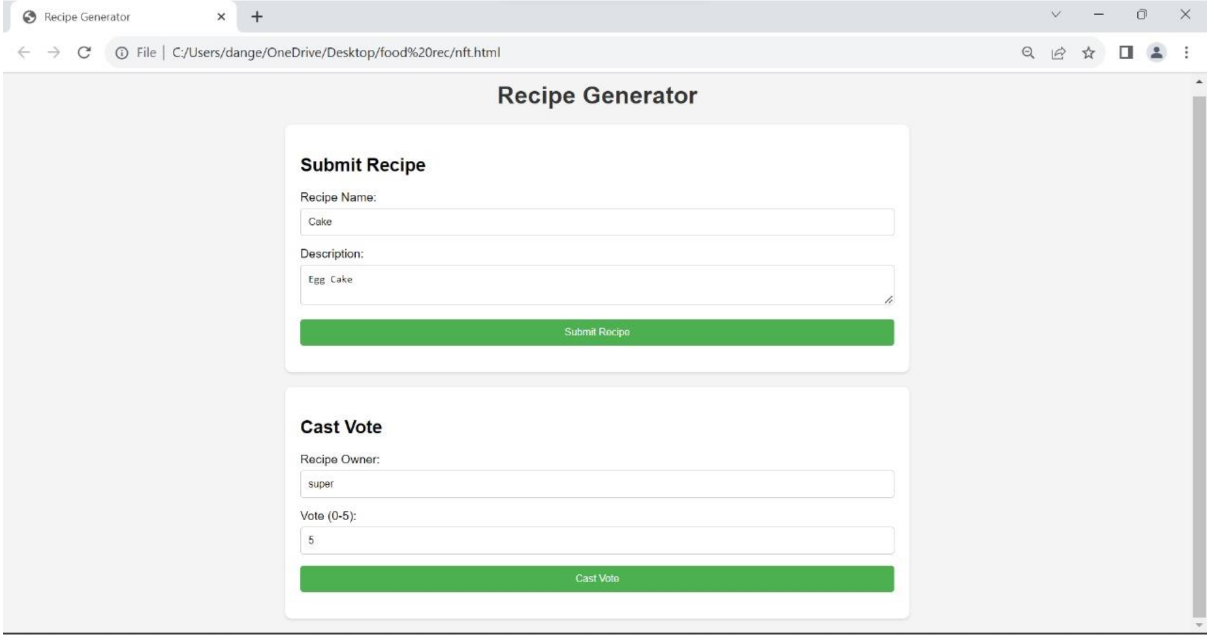
My Recipe Wish List

Generated Recipe: Recipe: Fairy Dust Delight Summon the Kitchen Spirits

Wish List

Enchanted Quinoa Delight Conjure a Flavorful Spell

Fairy Dust Delight Summon the Kitchen Spirits



Recipe Generator

Submit Recipe

Recipe Name: Cake

Description: Egg Cake

Cast Vote

Recipe Owner: super

Vote (0-5): 5

REFERENCES

1. [Edamam Recipe API](https://developer.edamam.com/edamam-recipe-api): A comprehensive API that provides access to a large database of recipes.

2. [Spoonacular API](https://spoonacular.com/food-api): Offers various food and recipe-related APIs, including a recipe search API.

3. [Food.com Datasets](https://www.food.com/about/data): Food.com provides access to its recipe database for non-commercial use.

4. [Yummly Dataset](https://www.yummly.com/dish/458465/download-yummly-recipes): A dataset of recipes available for download.

5. [NLTK](https://www.nltk.org/): Natural Language Toolkit for Python. Useful for text processing and analysis.

6. [spaCy](https://spacy.io/): Another powerful library for natural language processing.

7. [Beautiful Soup](https://www.crummy.com/software/BeautifulSoup/): A Python library for pulling data out of HTML and XML files.

8. [Scrapy](https://scrapy.org/): An open-source and collaborative web crawling framework for Python

9. Explore how smart contracts on blockchain platforms like Ethereum can be used to establish ownership and copyright of recipes. You might find resources on Ethereum's documentation helpful.

10. Consider using decentralized storage solutions like IPFS (InterPlanetary File System) to store recipes. This ensures data integrity and availability.