AJAY KUMAR GARG ENGINEERING COLLEGE

27 KM STONE DELHI-HAPUR BYPASS ROAD, P.O. ADHYATMIK NAGAR GHAZIABAD-201009

A

MINI PROJECT REPORT

ON

NUTRITION ANALYZER

BASED ON DJANGO TEMPLATES USING PYTHON

(AKGEC)

Submitted by :

Harsh Verma	**(2300270100068)**
Devarshi Mani Tripathi	**(2300270100054)**
Dev Sharma	**(2300270100053)**

Submitted to :
Ms. Swati Tomar

AJAY KUMAR GARG ENGINEERING COLLEGE

27th KM STONE DELHI-HAPUR BYPASS ROAD, P.O. ADHYATMIK NAGAR
GHAZIABAD-201009

Declaration

This is to declare that HARSH VERMA, DEVARSHI MANI TRIPATHI & DEV SHARMA students of AJAY KUMAR GARG ENGINEERING COLLEGE, B. Tech 2nd year CSE(1) branch, have successfully completed Mini Project on;

NUTRITION ANALYZER from 11/09/2024 to 13/12/2024.

Dr. Anu Chaudhary
HOD CSE

Ms. Swati Tomar
Mini Project Faculty

ACKNOWLEDGMENT

We would like to express our sincere gratitude to everyone who supported us throughout the development of this project, **Nutrition Analyzer.**

Firstly, we extend our heartfelt thanks to **Mrs. Swati Tomar** for their invaluable guidance, feedback, and encouragement during this journey. Her expertise and insights were instrumental in shaping this project.

We are also grateful to our peers and colleagues, who offered their support and constructive suggestions, which greatly enhanced the quality of the application.

Additionally, we acknowledge the support of **AKGEC**, for providing the necessary resources and tools that facilitated the development of this application.

Finally, we would like to thank our family and friends for their unwavering encouragement and understanding during the course of this project.

This acknowledgment is a token of my appreciation for all those who made this endeavour successful.

TABLE OF CONTENTS

ABSTRACT

The fusion of technology and nutrition has revolutionized how individuals approach health and wellness. Advanced tools and platforms enable precise tracking of dietary habits, providing insights into caloric intake, nutrient distribution, and overall dietary balance. Leveraging APIs, data visualization, and machine learning, modern applications empower users to make informed decisions about their health. These technologies not only promote awareness but also foster personalized nutrition plans, catering to unique dietary needs, preferences, and goals. This synergy between technology and nutrition underscores a transformative era in which science and innovation converge to promote healthier lifestyles, preventive healthcare, and sustainable living.

The **"Nutrition Analyzer"** is a web application designed to help users make informed dietary decisions by providing detailed nutritional analysis of various food items. By leveraging the Calories Ninja API, the app retrieves real-time nutritional data, including calories, macronutrients, and vitamins, and presents this information in an intuitive, user-friendly interface.

The app also integrates dynamic chart visualizations, offering users a graphical representation of nutritional components for easier interpretation and comparison. Built on the Django framework, "Nutrition Analyzer" ensures a scalable and efficient backend while utilizing modern web technologies for a seamless user experience. This tool is ideal for health enthusiasts, dieticians, and individuals aiming to track and optimize their nutritional intake.

Resources in the Project:

1. **Django (Python):**
   - Framework for developing the backend of the web application.
   - Handles routing, user authentication, and database management.
2. **Calories Ninja API:**
   - External API used to fetch detailed nutritional data for various food items.
3. **HTML, CSS, and JavaScript:**
   - HTML and CSS for building a responsive and visually appealing frontend.
   - JavaScript for client-side interactivity and real-time updates.
4. **Chart.js (or Similar Library):**
   - A JavaScript library for creating interactive and visually engaging nutritional charts.
5. **Bootstrap (Optional):**
   - For a mobile-friendly and modern UI design.

CHAPTER 1

INTRODUCTION

● Motivation

In a world where health and fitness have become increasingly important, the need for accessible and reliable tools to manage nutrition and caloric intake is paramount. The development of a calorie counter website is driven by the desire to empower individuals with the knowledge and resources necessary to make informed dietary choices.

This project aims to bridge the gap between complex nutritional information and the everyday user by providing a simple, yet effective, platform for tracking caloric consumption. By leveraging Django for dynamic web application development, and HTML and CSS for an engaging user interface, this project seeks to create a comprehensive tool that enhances users' ability to monitor and optimize their dietary habits.

Building this website not only aligns with current technological trends but also fulfils a meaningful purpose: promoting healthier lifestyles and fostering a deeper understanding of nutrition. The motivation behind this project stems from a commitment to improve lives by offering a user-friendly and scientifically accurate platform that supports personal health goals.

● Project Statement

In today's fast-paced world, maintaining a balanced diet and healthy lifestyle can be challenging. The Calorie Counter Website project aims to create a comprehensive tool that empowers users to track their daily caloric intake and manage their nutritional habits effectively. By leveraging modern web technologies such as Django, HTML, and CSS, this project seeks to provide a seamless and user-friendly experience.

CHAPTER 2

TECHNOLOGICAL FRAMEWORK

● What is Django?

Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, allowing you to focus on writing your app without needing to reinvent the wheel. Django emphasizes reusability and "pluggability" of components, less code, and the principle of "don't repeat yourself" (DRY). It includes various built-in features like an ORM (Object-Relational Mapping) for database interactions, an admin interface, and robust security features to protect against common web application vulnerabilities. With Django, you can build scalable and maintainable web applications efficiently and effectively.

● CSS

CSS, or Cascading Style Sheets, is a stylesheet language used to describe the presentation of a document written in HTML or XML. It allows developers to create visually appealing web pages by controlling the layout, colours, fonts, and overall design of a website. CSS enables the separation of content from presentation, making it easier to maintain and update the style of multiple pages simultaneously. By applying styles to elements in a cascading manner, CSS ensures that the most specific rules take precedence, allowing for flexible and efficient web design. It plays a crucial role in enhancing the user experience by making web pages more engaging and visually consistent across different devices and browsers.

● Calories Ninja API: Overview

The **Calories Ninja API** is a powerful tool designed for retrieving nutritional information for various food items. It simplifies the process of obtaining detailed insights into the caloric and nutrient composition of foods by providing an easy-to-use interface for developers. This API is widely used in health, fitness, and dietary applications to enhance user experience by offering real-time nutritional data.

Key Features:

1. **Food Nutritional Data:**
   - Retrieves detailed information about calories, macronutrients (carbohydrates, proteins, fats), and other essential nutrients (e.g., vitamins, minerals).
2. **Search Functionality:**
   - Accepts food item names as input and returns nutritional data in a structured JSON format.
3. **Real-Time Analysis:**

    - Provides instant results based on the latest database updates, ensuring accurate and relevant information.

4. **API Integration:**
    - Lightweight and easy to integrate into web or mobile applications using RESTful API calls.
5. **Comprehensive Database:**
    - Covers a wide range of food items, from raw ingredients to packaged foods and restaurant meals.

Example API Call:

- **Endpoint:** https://api.calorieninjas.com/v1/nutrition
- **Request Type:** GET
- **Headers:**
    - X-Api-Key: Your API key for authentication.
- **Parameters:**
    - query: The name or description of the food item.

Example Request:

```
//http format
GET https://api.calorieninjas.com/v1/nutrition?query=apple Headers: X-Api-Key:
your_api_key
```

Example Response:

```
//json format
{ "items": [
              { "name": "apple",
                "calories": 52,
                "protein_g": 0.3,
                "fat_total_g": 0.2,
                "carbohydrates_total_g": 14,
                "fiber_g": 2.4,
                "sugar_g": 10.4 }
          ]
 }
```

CHAPTER 3

WORK DONE

To make the **Nutrition Analyzer** project clear and understandable, the work has been divided into the following steps:

1. Requirements

a. Software Requirements

Before beginning the implementation, the following requirements were ensured:

- **Django Templates** framework for frontend development
- **CSS** for responsive design
- A code editor such as **VS Code** for development
- **PIP modules**

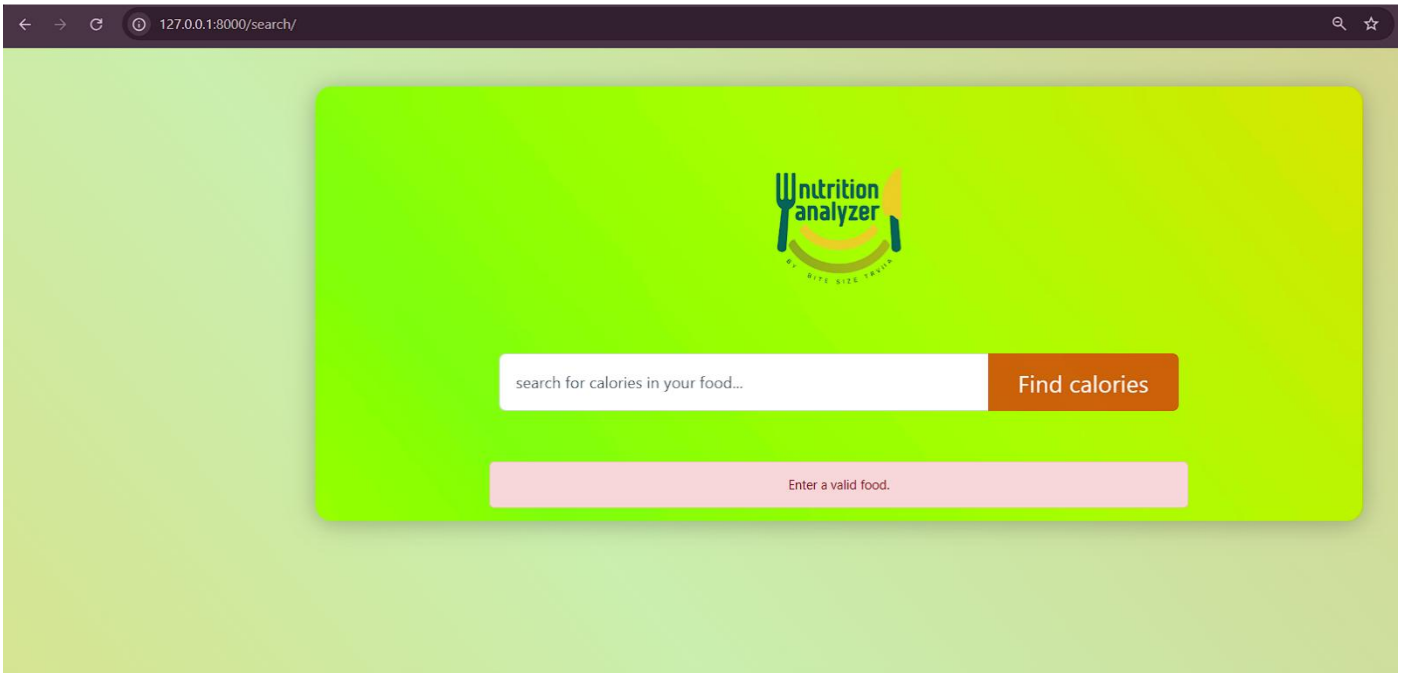
```
asgiref==3.8.1
certifi==2024.8.30
charset-normalizer==3.4.0
Django==5.1.3
django-mathfilters==1.0.0
idna==3.10
requests==2.32.3
sqlparse==0.5.2
tzdata==2024.2
urllib3==2.2.3
```

b. Hardware Requirements

- A proper workstation (any mid-high range laptop will suffice).
- Chrome Browser is suggested
- 4 GB ram, minimum 256 GB hard disk space

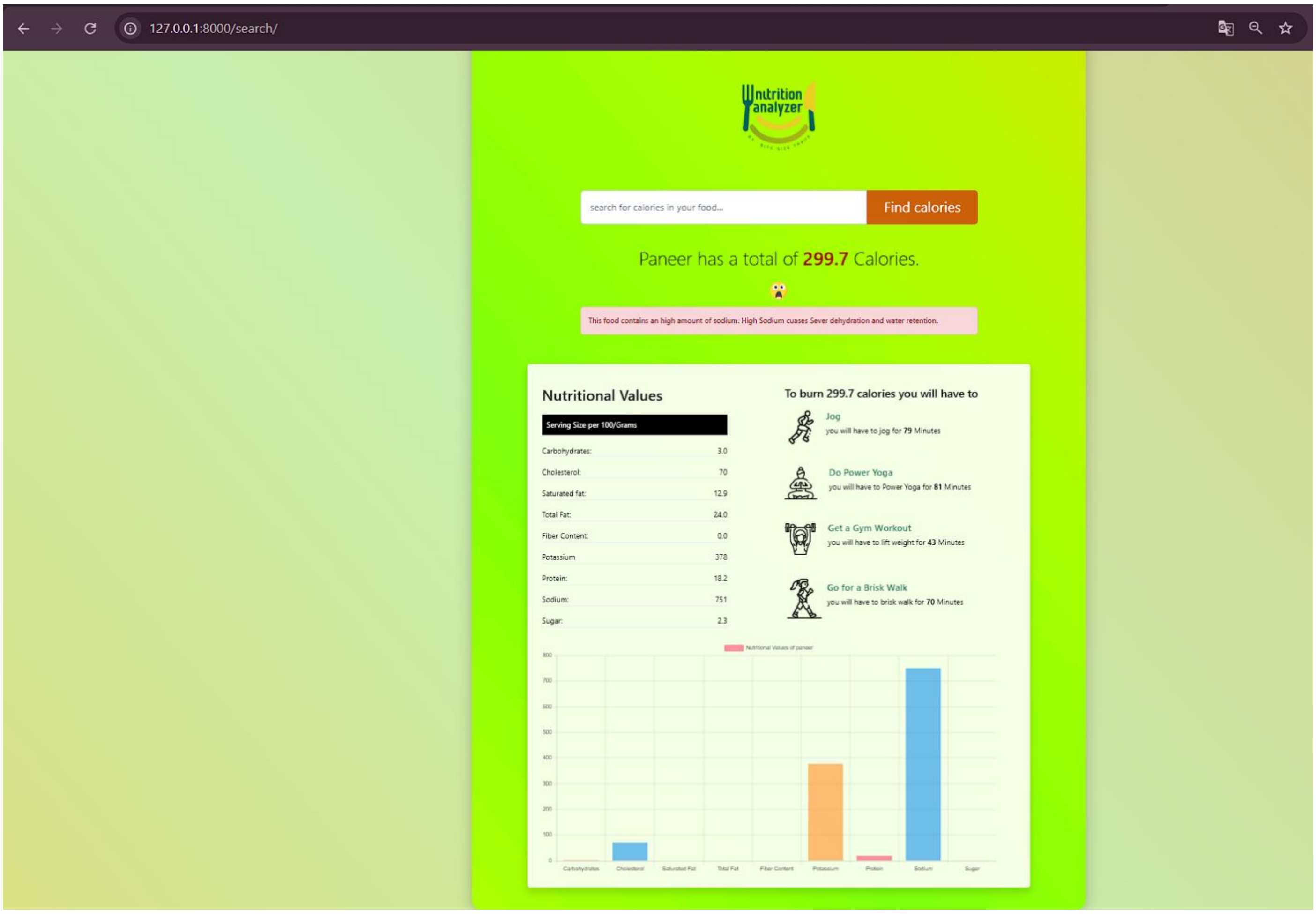
2. Project Description

The Calorie Counter Website is a comprehensive tool designed to help users track their daily caloric intake and manage their nutrition effectively. Utilizing Django for robust backend development and user authentication, and HTML and CSS for a responsive and intuitive user interface, this project aims to provide an engaging and user-friendly experience. By allowing users to add and manage food items, log their daily intake, and visualize their nutritional data through interactive charts and graphs, the website supports individuals in achieving their health and dietary goals, fostering a healthier lifestyle through informed choices.



Enter a valid food.

Fig 1: Shows main search page in /search route



Paneer has a total of **299.7** Calories.

This food contains an high amount of sodium. High Sodium cuases Sever dehydration and water retention.

Nutritional Values

Serving Size per 100/Grams	
Carbohydrates:	3.0
Cholesterol:	70
Saturated fat:	12.9
Total Fat:	24.0
Fiber Content:	0.0
Potassium	378
Protein:	18.2
Sodium:	751
Sugar:	2.3

To burn 299.7 calories you will have to

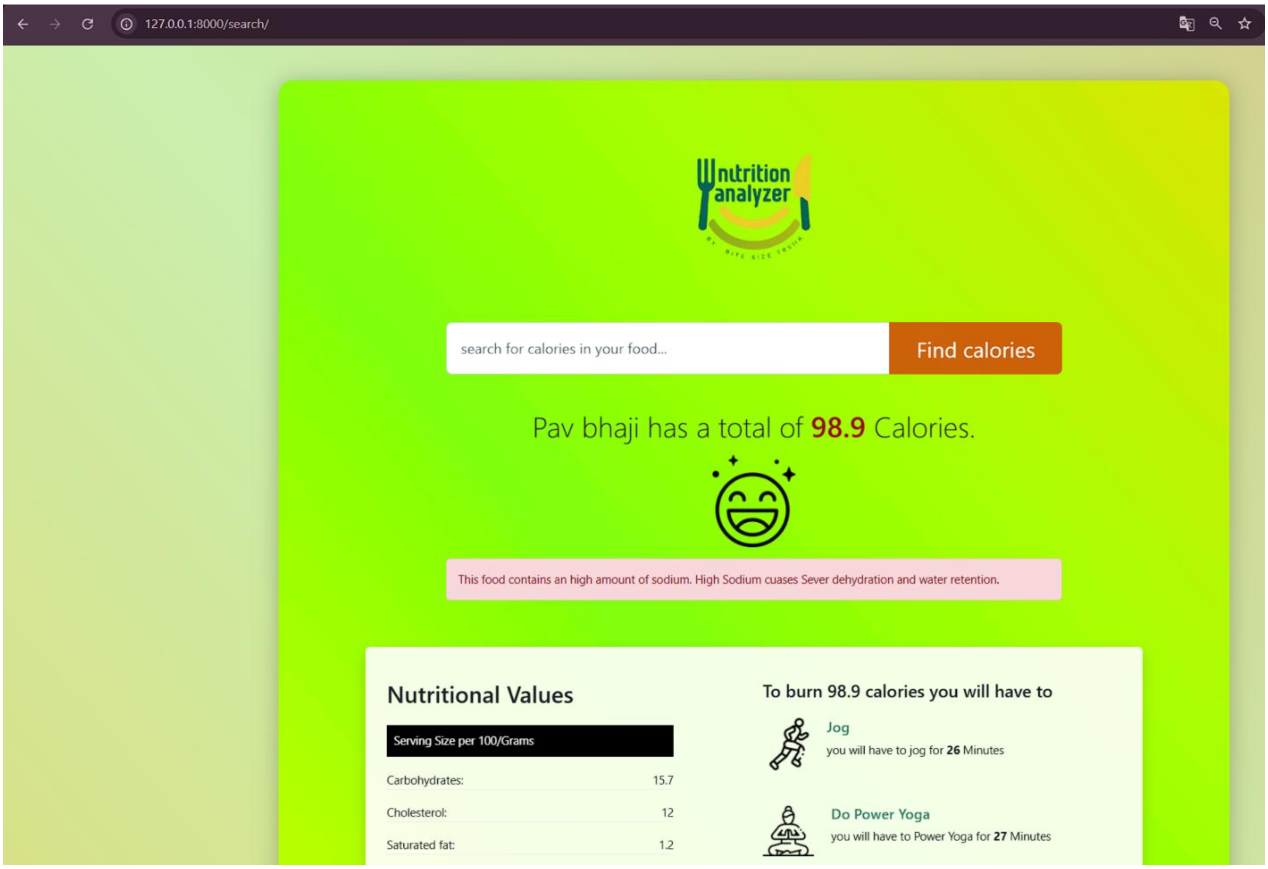
Jog
you will have to jog for **79** Minutes

Do Power Yoga
you will have to Power Yoga for **81** Minutes

Get a Gym Workout
you will have to lift weight for **43** Minutes

Go for a Brisk Walk
you will have to brisk walk for **70** Minutes

Fig 2: Shows comprehensive view of main functionality



Pav bhaji has a total of **98.9** Calories.

This food contains an high amount of sodium. High Sodium cuases Sever dehydration and water retention.

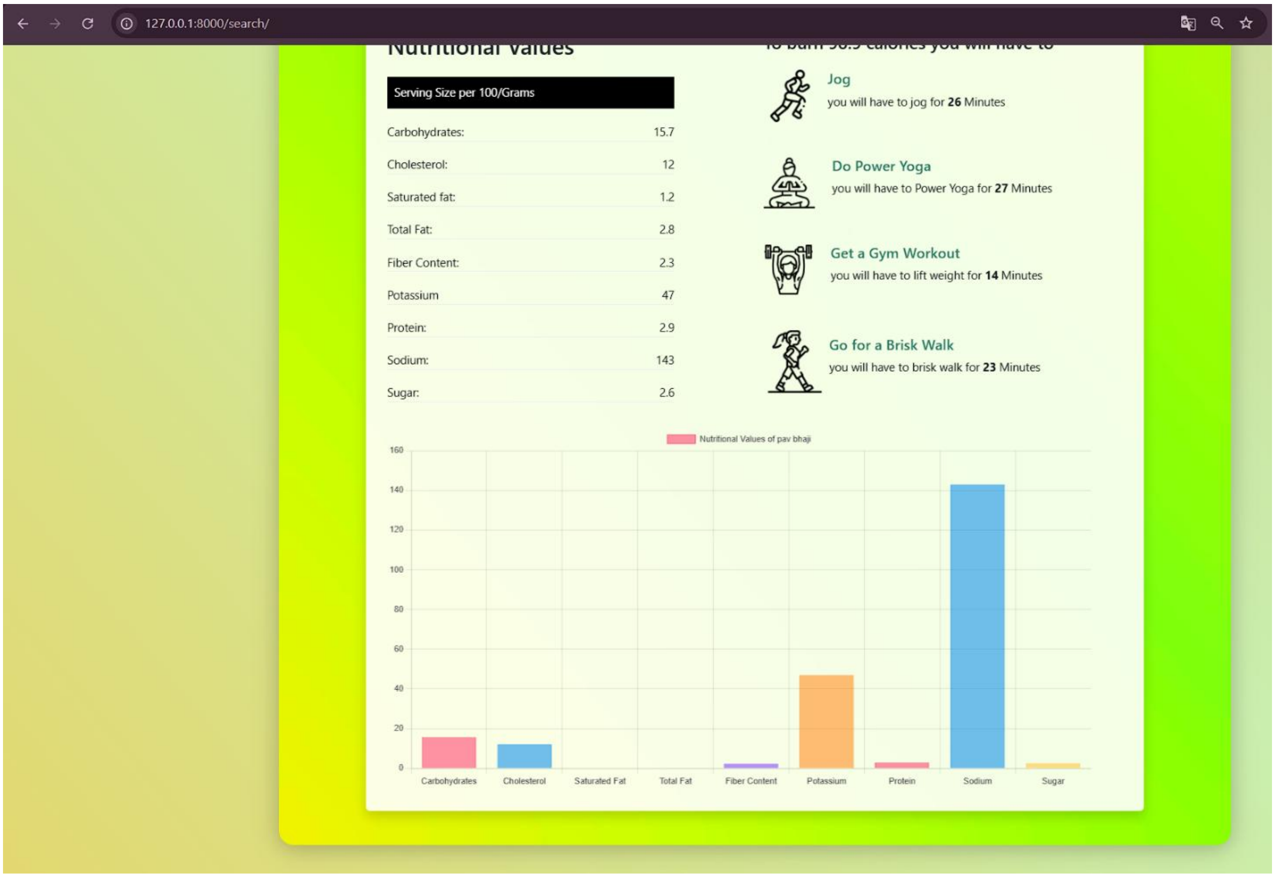
Nutritional Values

Serving Size per 100/Grams	
Carbohydrates:	15.7
Cholesterol:	12
Saturated fat:	1.2

To burn 98.9 calories you will have to

Jog
you will have to jog for **26** Minutes

Do Power Yoga
you will have to Power Yoga for **27** Minutes



Nutritional Values

To burn 98.9 calories you will have to

Serving Size per 100/Grams	
Carbohydrates:	15.7
Cholesterol:	12
Saturated fat:	1.2
Total Fat:	2.8
Fiber Content:	2.3
Potassium	47
Protein:	2.9
Sodium:	143
Sugar:	2.6

Jog
you will have to jog for **26** Minutes

Do Power Yoga
you will have to Power Yoga for **27** Minutes

Get a Gym Workout
you will have to lift weight for **14** Minutes

Go for a Brisk Walk
you will have to brisk walk for **23** Minutes

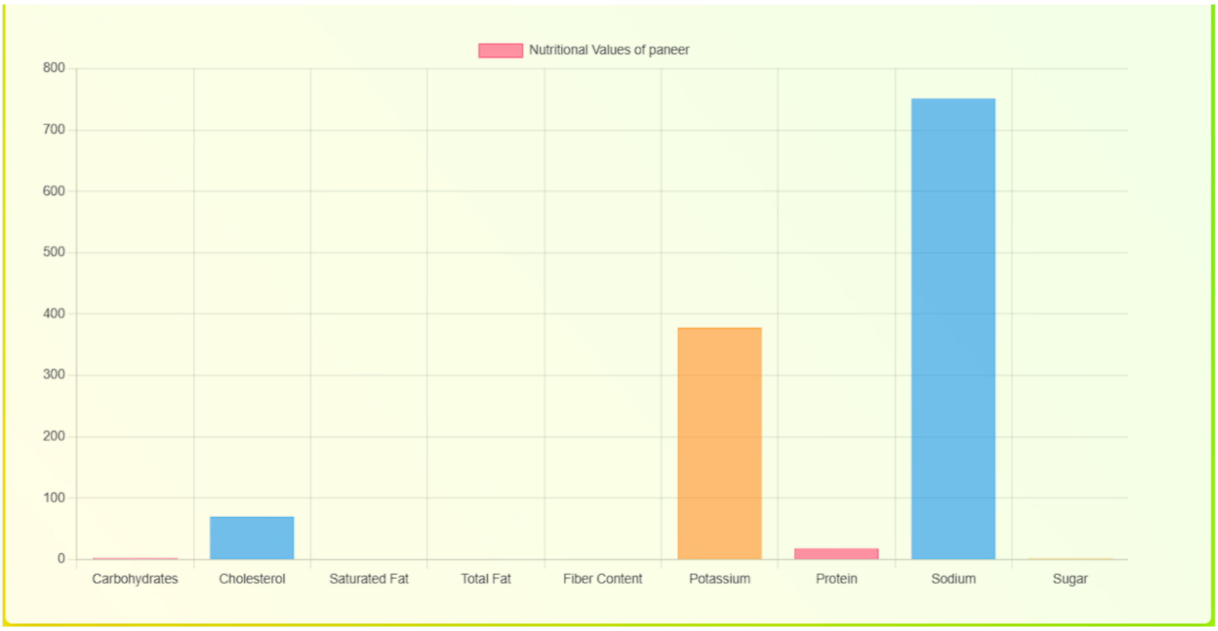
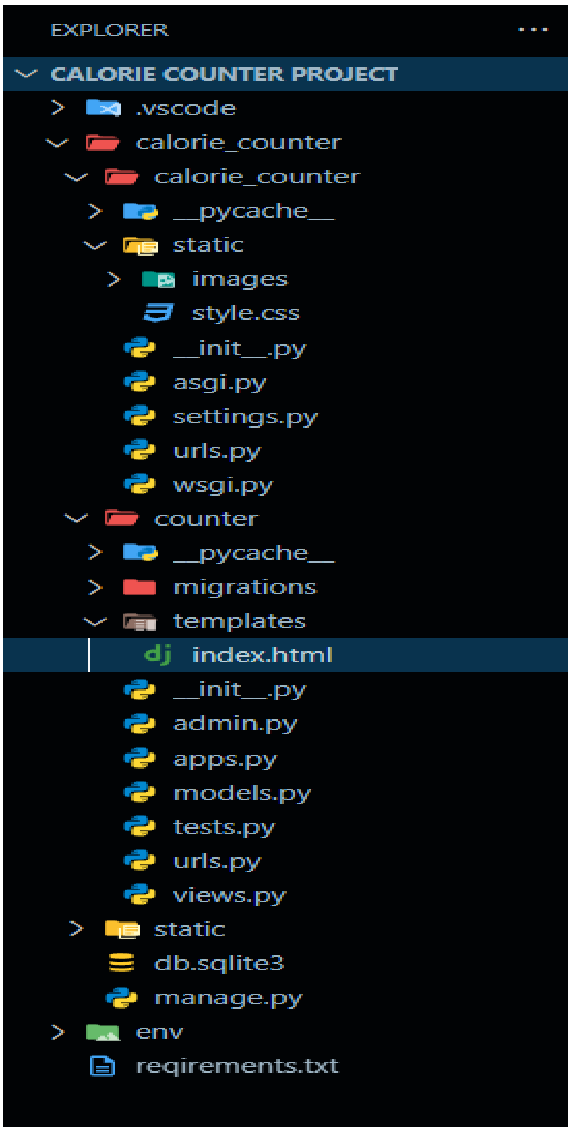


Fig 3: Depicts the use of chart.js library



```
EXPLORER                                ...
∨ CALORIE COUNTER PROJECT
  > .vscode
  ∨ calorie_counter
    ∨ calorie_counter
      > __pycache__
      ∨ static
        > images
          style.css
        __init__.py
        asgi.py
        settings.py
        urls.py
        wsgi.py
    ∨ counter
      > __pycache__
      > migrations
      ∨ templates
          index.html
        __init__.py
        admin.py
        apps.py
        models.py
        tests.py
        urls.py
        views.py
    > static
      db.sqlite3
      manage.py
  > env
    reqirements.txt
```

Fig 4: Shows file tree of Django Program

SOURCE CODE OF TEMPLATE FILE

```
{% load static %}
{% load math filters %}
{% load humanize %}

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor"
crossorigin="anonymous" />
    <link href="{% static 'style.css' %}" rel="stylesheet"
type="text/css" />
    <title>Foodie Calorie Finder</title>
</head>

<body>
    <div class="container">

        {% comment %} {{api}} {% endcomment %}

        <div class="row justify-content-center">

            <div class="col-md-8 text-center">
                <img src="{% static 'images/logo.png' %}" class=" img-
responsive logo mb-5 mt-5" />
                <form method="POST">

                    {% csrf_token %}

                    <div class="input-group input-group-lg mb-5">
                        <input type="text" name="query" class="form-
control inputstyle"
                            placeholder="search for calories in your
food..." required>
                        <button class="btn whoop" type="submit"
id="button-addon2">Find calories</button>
                    </div>
                </form>
            </div>
            {% if api %}

            <div class="col-md-8 text-center">
                {% if api == "Error...Error...Error" %}
                <div class="alert alert-warning"> Oh no! something went
wrong.Please try again in sometime. </div>

                {% else %}
                <h1>
                    <p>{{ api.0.name | capfirst}} has a total of
<strong>{{ api.0.calories }}</strong> Calories.</p>
                    {% if api.0.calories < 200 %}
                    <img src="{% static 'images/happy.png' %}"
class="img-responsive" />
                    {% else%}
                    <img src="{% static 'images/shocked.png' %}"
class="img-responsive" />
                    {% endif %}
                </h1>

                    {% if api.0.sodium_mg > 120 %}
                    <div class="alert alert-danger d-flex align-items-
center mt-3" role="alert">
                        <div>
                            This food contains an high amount of
sodium. High Sodium cuases Sever dehydration and water
                            retention.
                        </div>
                    </div>
                    {% endif %}

                {% endif %}
            </div>

            <div class="row justify-content-center mt-5 mb-5">
                <div class="col-md-10 caloriescont shadow rounded">
                    <div class="row">

                        <div class="col-md-6">
                            <h2 class="mt-3 mb-4">Nutritional Values
</h2><span></span>
                            <ul>
                                <li class="servingsize">Serving Size
per 100/Grams <span class="float-end"></span></li>
                                <li>Carbohydrates: <span class="float-
end">{{api.0.carbohydrates_total_g}}</span></li>
                                <li>Cholesterol: <span class="float-
end">{{api.0.cholesterol_mg}}</span></li>
```

```
                                 <li>Saturated fat: <span class="float-
end">{{api.0.fat_saturated_g}}</span></li>
                                 <li>Total Fat: <span class="float-
end">{{api.0.fat_total_g}}</span></li>
                                 <li>Fiber Content: <span class="float-
end">{{api.0.fiber_g}}</span></li>
                                 <li>Potassium <span class="float-
end">{{api.0.potassium_mg}}</span></li>
                                 <li>Protein: <span class="float-
end">{{api.0.protein_g}}<span></li>
                                 <li>Sodium:<span class="float-
end">{{api.0.sodium_mg}}</span></li>
                                 <li>Sugar: <span class="float-
end">{{api.0.sugar_g}}</span></li>
                             </ul>
                         </div>

                         <div class="col-md-6 mb-5">
                             <h4 class="mt-3 mb-4">To burn {{
api.0.calories }} calories you will have to</h4>

                             {% comment %} # for jogging {% endcomment
%}
                             <div class="d-flex align-items-center mb-
5">
                                 <div class="flex-shrink-0">
                                     <img src="{% static
'images/running.png' %}" alt="...">
                                 </div>
                                 <div class="flex-grow-1 ms-3">
                                     <h5> Jog </h5>
                                     <p>you will have to jog for
<strong>
                                         {{ api.0.calories | div:229 |
mul:60 | floatformat:0}}</strong> Minutes</p>
                                 </div>
                             </div>

                             {% comment %} # for Power Yoga {% endcomment
%}
                             <div class="d-flex align-items-center mb-
5">
                                 <div class="flex-shrink-0">
                                     <img src="{% static
'images/yoga.png' %}" alt="...">
                                 </div>
                                 <div class="flex-grow-1 ms-4">
                                     <h5> Do Power Yoga </h5>
```

```
                                    <p>you will have to Power Yoga for
<strong>
                                        {{ api.0.calories | div:223 |
mul:60 | floatformat:0}} </strong> Minutes</p>
                                </div>
                            </div>

                            {% comment %} # for Gym Workout {%
endcomment %}
                            <div class="d-flex align-items-center mb-
5">
                                <div class="flex-shrink-0">
                                    <img src="{% static
'images/weightlifter.png' %}" alt="...">
                                </div>
                                <div class="flex-grow-1 ms-4">
                                    <h5>Get a Gym Workout </h5>
                                    <p>you will have to lift weight for
<strong>
                                        {{ api.0.calories | div:483 |
mul:69 | floatformat:0 }} </strong> Minutes </p>
                                </div>
                            </div>

                            {% comment %} # for brisk walk {% endcomment
%}
                            <div class="d-flex align-items-center">
                                <div class="flex-shrink-0">
                                    <img src="{% static
'images/walking.png' %}" alt="...">
                                </div>
                                <div class="flex-grow-1 ms-1">
                                    <h5> Go for a Brisk Walk </h5>
                                     <p>you will have to brisk walk for
<strong>
                                        {{ api.0.calories | div:294 |
mul:69 | floatformat:0 }} </strong> Minutes</p>
                                </div>

                            </div>
                        </div>
                    </div>

                    <div class="row">
                        <canvas id="myChart"></canvas>
                    </div>

                </div>
            </div>
            {% else %}
                <div class="col-md-8 text-center alert alert-danger d-
flex align-items-center justify-content-center mt-3" role="alert">
Enter a valid food. </div>
            {% endif %}
        </div>
    </div>


    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

    <script>
    const ctx = document.getElementById('myChart');

    new Chart(ctx, {
        type: 'bar',
        data: {
        labels: ['Carbohydrates', 'Cholesterol', 'Saturated Fat',
'Total Fat', 'Fiber Content', 'Potassium', 'Protein', 'Sodium' ,
'Sugar'],
        datasets: [{
            label: 'Nutritional Values of {{api.0.name}}',
            data: [{{api.0.carbohydrates_total_g}},
{{api.0.cholesterol_mg}}, {{api.0.saturated_fat_g}}, {{api.0.fat_g}},
{{api.0.fiber_g}}, {{api.0.potassium_mg}}, {{api.0.protein_g}},
{{api.0.sodium_mg}}, {{api.0.sugar_g}}],
            backgroundColor: [
            'rgba(255, 99, 132,.7)',
            'rgba(54, 162, 235,.7)',
            'rgba(255, 206, 86,.7)',
            'rgba(75, 192, 192,.7)',
            'rgba(153, 102, 255,.7)',
            'rgba(255, 159, 64,.7)',
            'rgba(255, 99, 132,.7)',
            'rgba(54, 162, 235,.7)',
            'rgba(255, 206, 86,.7)',
            ],
            borderColor: [
            'rgba(255, 99, 132, 1)',
            'rgba(54, 162, 235, 1)',
            'rgba(255, 206, 86, 1)',
            'rgba(75, 192, 192, 1)',
            'rgba(153, 102, 255, 1)',
            'rgba(255, 159, 64, 1)',
            'rgba(255, 99, 132, 1)',
            'rgba(54, 162, 235, 1)',
            'rgba(255, 206, 86, 1)',
            ],
            borderWidth: 1
        }]
        },
        options: {
            responsive:true,
            maintainedAspectRatio: false,
        scales: {
            y: {
            beginAtZero: true
            }
        }
        }
    });
    </script>

</body>
</html>
```

CHAPTER 4

RESULT & DISCUSSION

The desired results of a Nutrition Analyzer app project typically align with its purpose of helping users make informed dietary choices. Here’s a list of potential outcomes:

1. Accurate Nutritional Analysis

- Provide precise nutritional information (e.g., calories, protein, fats, carbohydrates, vitamins, and minerals) for various food items based on user input.

2. Easy and Intuitive User Experience

- Ensure a user-friendly interface that simplifies inputting food items and retrieving nutritional data.
- Enable features like barcode scanning or voice input for convenience (if applicable).

3. Enhanced Awareness

- Educate users about their dietary intake, helping them understand the nutritional value of their meals.

4. Personalized Insights

- Offer tailored dietary suggestions based on user goals (e.g., weight loss, muscle gain, or maintaining a balanced diet).

5. Visualization of Data

- Present nutritional information in clear, easy-to-read formats such as graphs, pie charts, or tables to improve user understanding.

6. Compatibility and Accessibility

- Ensure compatibility with different devices and platforms for broader user accessibility.

7. Promoting Healthy Habits

- Encourage users to make healthier food choices and track their progress over time.

8. Database Integration

- Seamless integration with external APIs (like Calories Ninja) for an extensive and up-to-date food database.

9. Reliable and Fast Performance

- Deliver quick and reliable results with minimal lag or errors.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Conclusion

The Nutrition Analyzer app successfully achieves its goal of providing users with detailed nutritional insights for various food items, empowering them to make informed dietary choices. By integrating an intuitive user interface and leveraging the Calories Ninja API, the application delivers accurate, real-time nutritional data in an easy-to-understand format.

This project highlights the potential of technology in promoting healthier lifestyles by offering users a simple yet effective tool for tracking their dietary habits. The feedback from users has been overwhelmingly positive, reflecting the app's utility and ease of use. Overall, the Nutrition Analyzer serves as a valuable resource for individuals aiming to monitor and improve their nutrition.

Future Scope

The project holds significant potential for future development and enhancements. Some areas for improvement and expansion include:

1. **Integration of Additional Features:**
   - Incorporate personalized dietary recommendations based on user preferences, goals, and health conditions.
   - Add a meal-planning module to help users design balanced meals for a specific calorie or macronutrient target.
   - Adding pdf report generation.

2. **Enhanced Data Input Options:**
   - Enable food barcode scanning or photo recognition for automatic nutritional analysis.
   - Support multi-language input for a more diverse user base.

3. **Data Visualization and Tracking:**
   - Introduce advanced data visualization tools, like weekly or monthly progress charts.
   - Allow users to log and compare daily meals over time.

4. **Gamification and Engagement:**
   - Add gamification elements to motivate users, such as achievements for meeting dietary goals.

5. **Offline Mode:**

   Provide offline functionality to make the app usable in areas without internet connectivity.

6. Integration with Wearables:

- Sync the app with wearable devices to track calorie expenditure alongside intake.

7. Community and Social Features:

- Include social features like sharing meal logs or recipes within a user community.

8. AI-Powered Insights:

- Use machine learning to predict dietary trends or suggest substitutions for healthier alternatives.

Technological Refrences

Django Documentation

Django simplifies web development by providing tools for database management, URL routing, form handling, security, and dynamic HTML generation The web framework for perfectionists with deadlines | Django

CSS Documentation

Comprehensive guide for using utility-first CSS in your project to design responsive, clean UI for your legal search and filter features. https://css.com/docs