FOOD-GEN

PROJECT REPORT

Submitted by

SRIRANGANATHAN M

(Reg No:9923151058)

Under the guidance of

Dr. K. BALASUBRAMANIAN

In partial fulfilment of the requirements for award of the degree of

MASTER OF COMPUTER APPLICATIONS

2023 – 2025

DEPARTMENT OF COMPUTER APPLICATIONS

SCHOOL OF COMPUTING

KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION

MAY-2024

KALASALINGAM

Academy of Research and Education

DEEMED TO BE UNIVERSITY

Estd.U/S 3 of UGC Act 1956.

Anand Nagar, Krishnankoil - 626126. Srivilliputtur (Via), Virudhunagar (Dt), Tamil Nadu | info@kalasalingam.ac.in | www.kalasalingam.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER APPLICATIONS

BONAFIDE CERTIFICATE

This is to certify that this project report titled "**FOOD-GEN"** is the bonafide work of **SRIRANGANATHAN M (Reg No: 9923151050)** carried out in partial fulfilment of the requirement of the award of the degree of the Master of Computer Applications at Kalasalingam Academy of Research and Education under my supervision.

Project Guide & Head of the Department

Dr. K. Balasubramanian

Head of the Department

Department of Computer Applications

KARE

Submitted for the project viva-voce examination held on ______________

ABSTRACT

Food-Gen: A Mobile Application for Recipe Discovery

Food-Gen is a mobile application designed to simplify the process of finding recipes for various dishes using a user-friendly interface developed with Flet and powered by Python. This innovative application caters to individuals who often find themselves in need of a recipe for specific foods but are unsure where to start. Food-Gen allows users to effortlessly select a food item they desire and instantly access detailed recipes.

The core functionality of Food-Gen revolves around an intuitive selection process where users can browse through a wide range of food categories and individual dishes. Upon selecting a desired food item, the application retrieves and displays a comprehensive recipe, including ingredients, preparation steps, and cooking times. This ensures that users have all the necessary information at their fingertips to prepare their chosen dish successfully.

Food-Gen leverages a robust backend system to maintain a vast and diverse database of recipes, ensuring that users can find recipes for both common and exotic dishes. The integration of Flet for the user interface provides a seamless and visually appealing experience, enhancing user engagement and satisfaction. Additionally, the application is designed to be responsive, making it accessible on various mobile devices, thus catering to the on-the-go needs of modern users.

In summary, Food-Gen aims to transform the way users discover and interact with recipes, offering a quick, efficient, and enjoyable solution for anyone looking to expand their culinary repertoire. With its easy-to-navigate interface and comprehensive recipe database, Food-Gen is poised to become an essential tool for food enthusiasts and home cooks alike.

TABLE OF CONTENT

1. INTRODUCTION

In today's fast-paced world, cooking at home has become both a necessity and a pleasure for many individuals. However, the task of finding the right recipe for a specific dish can be daunting, especially for those with busy schedules or limited culinary knowledge. Recognizing this challenge, we developed Food-Gen, a mobile application designed to make recipe discovery simple and accessible for everyone.

Food-Gen harnesses the power of Python and the intuitive capabilities of the Flet framework to deliver a seamless user experience. The application provides users with an extensive selection of recipes, allowing them to easily find and prepare meals that match their cravings and dietary preferences. By focusing on user-friendly navigation and comprehensive recipe information, Food-Gen aims to demystify the cooking process and inspire confidence in the kitchen.

The application's interface is designed to be intuitive and engaging, enabling users to browse through food categories and specific dishes with ease. Once a food item is selected, Food-Gen quickly presents a detailed recipe, complete with ingredient lists, step-by-step instructions, and cooking tips. This ensures that users, regardless of their cooking expertise, can successfully prepare delicious meals.

Food-Gen also addresses the need for variety in home cooking by offering a diverse range of recipes from different cuisines and dietary categories. This diversity not only caters to various tastes and preferences but also encourages users to explore and experiment with new dishes. Through its innovative design and comprehensive content, Food-Gen aspires to become an indispensable kitchen companion for users around the globe.

In conclusion, Food-Gen represents a significant advancement in recipe discovery applications, combining cutting-edge technology with an intuitive user experience. By making it easier for users to find and follow recipes, Food-Gen supports the growing trend of home cooking and promotes a more enjoyable and efficient culinary experience.

2. SYSTEM OVERVIEW

2.1 Existing System

Currently, many individuals rely on various sources to find recipes, including cookbooks, food blogs, and general-purpose search engines. While these sources provide a wealth of information, they often lack the convenience and specificity needed for quick and easy recipe discovery. Users may have to sift through numerous irrelevant search results or navigate poorly organized websites, leading to a time-consuming and sometimes frustrating experience.

Limitations:

- **Time-Consuming Searches:** Users often spend a significant amount of time searching for the right recipe, filtering through irrelevant results.
- **Lack of Personalization:** Existing systems rarely offer personalized recipe suggestions based on user preferences and dietary needs.
- **Static Content:** Traditional cookbooks and many websites provide static content that lacks interactive features and real-time updates.

2.2 Proposed System

Food-Gen addresses these limitations by offering a streamlined, user-centric mobile application that makes recipe discovery effortless. The application features:

- **Intuitive User Interface:** Developed using Flet, the interface is designed for ease of use, allowing users to quickly browse and select recipes.
- **Personalized Recommendations:** Advanced algorithms provide personalized recipe suggestions based on user preferences and past interactions.

Advantages:

- **Efficiency:** Quickly find relevant recipes without the need for extensive searching.
- **Personalization:** Receive tailored recipe suggestions that match individual dietary preferences and tastes.
- **Convenience:** Access a comprehensive database of recipes on the go, making meal planning and preparation easier.
- **Inspiration:** Discover new and diverse recipes, encouraging culinary exploration and experimentation.

3.1 SYSTEM SPECIFICATIONS

3.1 Hardware Specifications

Processor	:	AMD Ryzen 5
Ram	:	4GB DDR4
Storage	:	512GB M.2 NVMe™ PCIe® 3.0 SSD
Display	:	14.0-inch Full HD (1920 x 1080) IPS-level Panel
Keyboard	:	Chiclet Keyboard with 1.4mm key travel
Touchpad	:	Touchpad with support for Number Pad
Processor Speed	:	3.3 GHz
Android version	:	11 Funtouch OS

3.2 Software Specifications

Operating System	:	Windows 10 Pro
Technology Used	:	Python 3.12.3 & Flet 0.22.0

4. SYSTEM DESCRIPTION

4.1 Python 3.12.3

Python 3.12 is the latest stable release of the Python programming language, with a mix of changes to the language and the standard library. The library changes focus on cleaning up deprecated APIs, usability, and correctness. Of note, the distutils package has been removed from the standard library. Filesystem support in os and pathlib has seen a number of improvements, and several modules have better performance.

The language changes focus on usability, as f-strings have had many limitations removed and ‘Did you mean …’ suggestions continue to improve. The new type parameter syntax and type statement improve ergonomics for using generic types and type aliases with static type checkers.

New syntax features:

- PEP 695, type parameter syntax and the type statement

New grammar features:

- PEP 701, f-strings in the grammar

Interpreter improvements:

- PEP 684, a unique per-interpreter GIL
- PEP 669, low impact monitoring
- Improved ‘Did you mean …’ suggestions for NameError, ImportError, and SyntaxError exceptions

Python data model improvements:

- PEP 688, using the buffer protocol from Python

Significant improvements in the standard library:

- The pathlib.Path class now supports subclassing
- The os module received several improvements for Windows support
- A command-line interface has been added to the sqlite3 module isinstance() checks against runtime-checkable protocols enjoy a speed up of between two and 20 times

- The asyncio package has had a number of performance improvements, with some benchmarks showing a 75% speed up.
- A command-line interface has been added to the uuid module
- Due to the changes in PEP 701, producing tokens via the tokenize module is up to 64% faster.

Security improvements:

- Replace the builtin hashlib implementations of SHA1, SHA3, SHA2-384, SHA2-512, and MD5 with formally verified code from the HACL* project. These builtin implementations remain as fallbacks that are only used when OpenSSL does not provide them.

4.2 Package

Flet:

Flet is a framework for building real-time web, desktop and mobile applications in Python.

No more complex architecture with JavaScript frontend, REST API backend, database, cache, etc. With Flet you just write a monolith stateful app in Python only and get multi-user, realtime Single-Page Application (SPA) or a mobile app.

To start developing with Flet, you just need your favorite IDE or text editor. No SDKs, no thousands of dependencies, no complex tooling - Flet has built-in web server with assets hosting and desktop clients.

Flet UI is built with Flutter, so your app looks professional and can be delivered to any platform. Flet simplifies Flutter model by combining smaller "widgets" into ready-to-use "controls" with imperative programming model. You get all the power of Flutter without having to learn Dart!

Flet app is deployed as a regular web app and can be instanly accessed with a browser or installed as a PWA on a mobile device. Web app also exposes an API that can be used by a Flet client (planned for future releases) running on iOS and Android and providing native mobile experience.

4.3 Command Lines

Pip Install flet:

Installing ‘flet’ provides a rich User Interface (UI) framework to quickly build interactive web, desktop and mobile apps in Python without prior knowledge of web technologies like HTTP, HTML, CSS or JavaScript. You build UI with controls based on Flutter widgets to ensure your programs look cool and professional.

Requirements:

Python 3.7 or above on Windows, Linux or macOS

Installation:

```
pip install flet
```

Run as a desktop app:

The following command will start the app in a native OS window:

```
flet run main.py
```

Run as a web app:

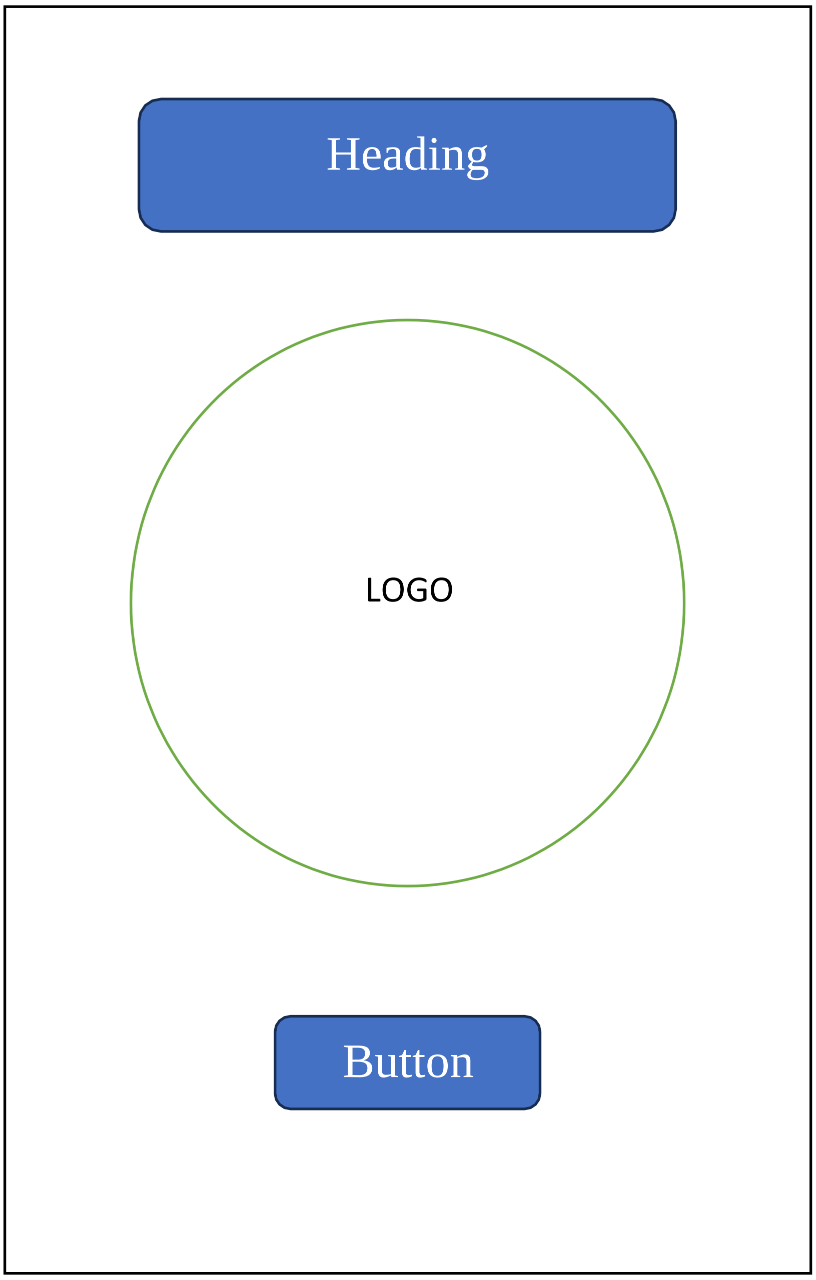
The following command will start the app as a web app:

```
flet run --web main.py
```

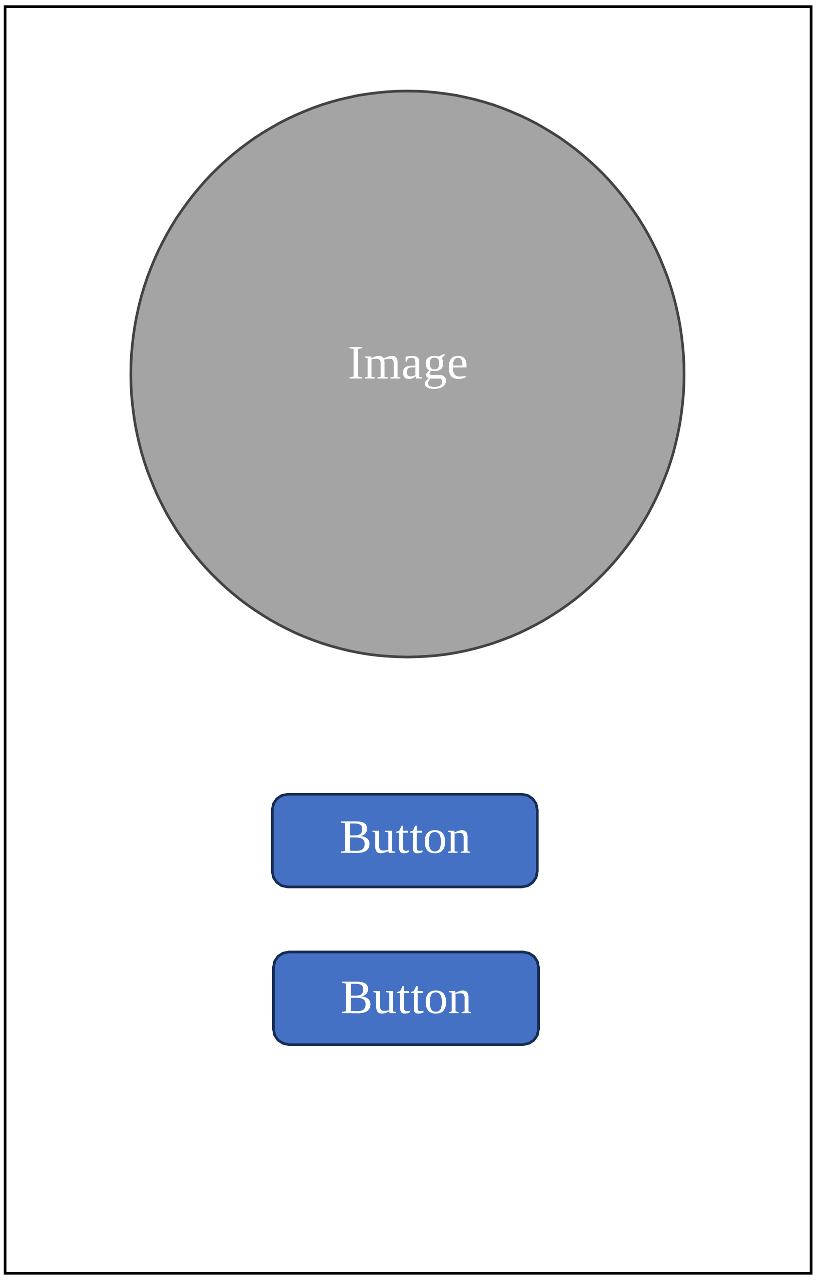
5 SYSTEM DESIGN

5.1 Application Design

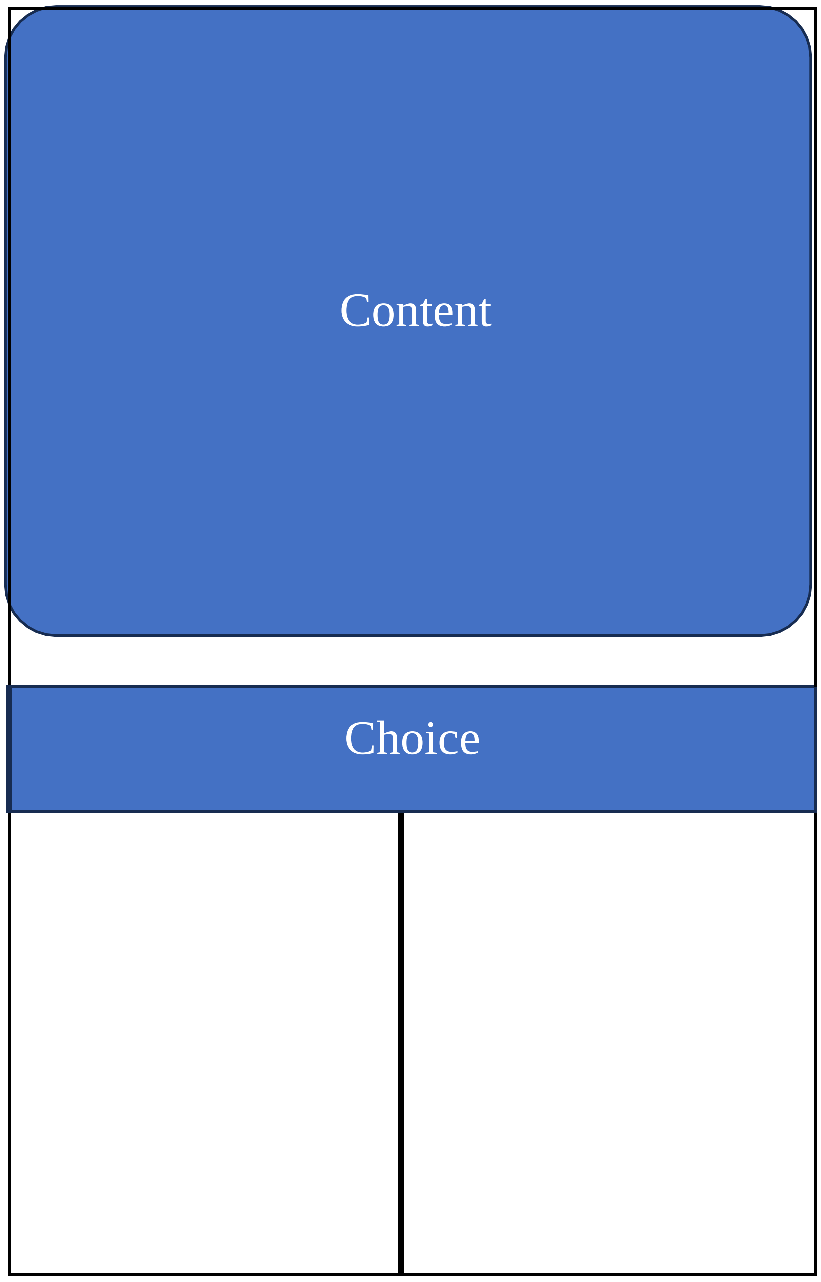
5.1.1 Logo Page:



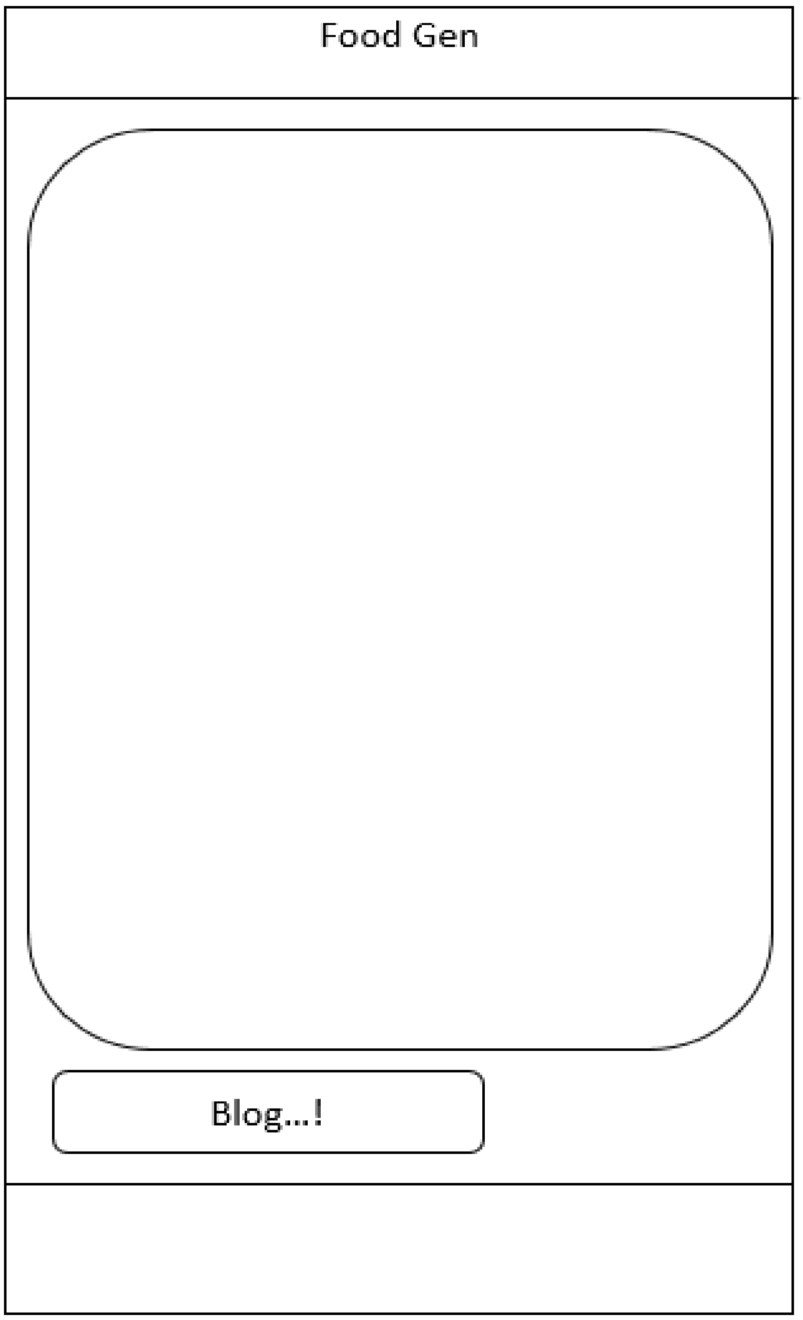
5.1.2 Language Page:



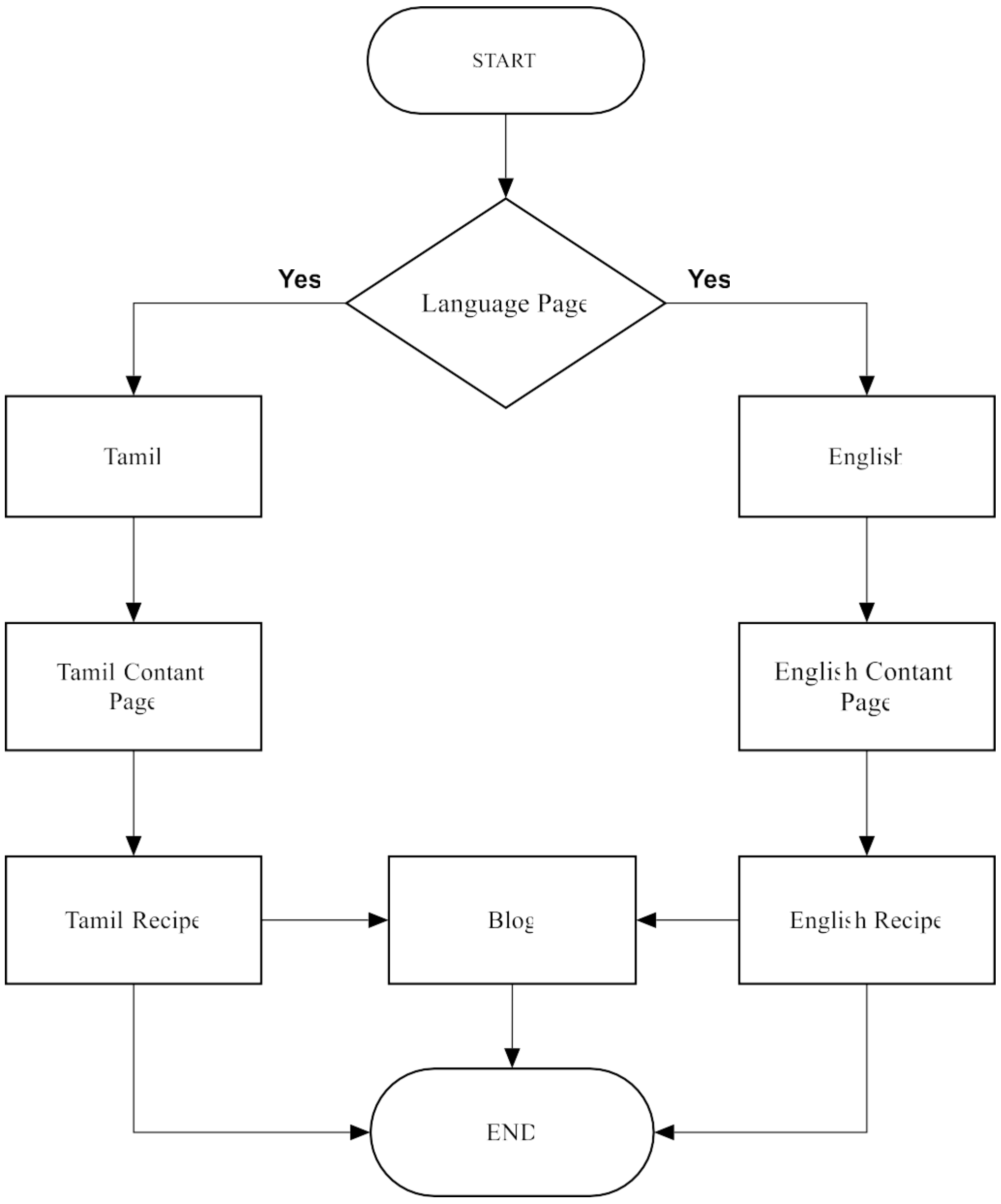
5.1.3 Home Page:



5.1.4 Recipe Page:



5.2 Data Flow Diagram



The data flow diagram which starts at logo page, once you click START button it moves to second page which contain Language, you can choose your suitable language (Tamil & English). Once you select Language it's move to Contant page and then Recipe. If you need more you can go to Blog to see more details.

5 SYSTEM IMPLEMENTATION

Implementing Food-Gen involves several stages, from initial planning and design to development, testing, and deployment. Below is a detailed overview of the key phases involved in the system implementation process.

5.1. Planning and Requirements Gathering:

- **Objective Definition:** Clearly define the objectives of Food-Gen, focusing on ease of recipe discovery and user satisfaction.
- **Requirements Gathering:** Identify and document functional and non-functional requirements. This includes understanding user needs, desired features, and technical constraints.

5.2. Design:

- **System Architecture:** Design the system architecture, including the frontend (user interface) and Blog.
- **User Interface Design:** Create wireframes and prototypes using Flet to visualize the layout and user flow. Focus on intuitive navigation and user-friendly design.

5.3. Development:

- **Frontend Development:** Develop the user interface using Flet, ensuring responsiveness and compatibility across different mobile devices.
- **Algorithm Implementation:** Develop and integrate algorithms for personalized recipe recommendations based on user preferences.

7 APPENDIX

7.1 Sample Coding:

7.1.1 Main.py

```
import flet as ft
from recipe_e import *
from recipe_t import *

def main(page: ft.Page):
    page.window_width = 400
    page.window_height = 800
    page.padding = 0
    page.theme_mode = ft.ThemeMode.LIGHT
    page.fonts = {
        "Nightcore": "/fonts/Nightcore.ttf",
        "Oswald-Bold": "/fonts/Oswald-Bold.ttf",
        "Ancient": "/fonts/Ancient Ad.ttf",
        "Orange Squash": "/fonts/Orange Squash Free.ttf",
        "Sticky":"/fonts/Sticky Memos Demo Two.ttf",
        "Tamilbold":"/fonts/NotoSansTamil-Bold.ttf",
        "Tamil":"/fonts/NotoSansTamil-Medium.ttf",
        "Poppinsbold":"/fonts/Poppins-Bold.ttf",
        "Poppins":"/fonts/Poppins-Medium.ttf"
    }
    page.theme = ft.Theme(font_family="Poppins")
    page.update()

    #appbar
    appbar = ft.CupertinoAppBar(
        leading=ft.IconButton(ft.icons.KEYBOARD_BACKSPACE,on_click=lambda _:
page.go("/lan")),
        bgcolor=ft.colors.SURFACE_VARIANT,
        trailing=ft.Icon(ft.icons.FASTFOOD_OUTLINED),
      middle=ft.Text("FoodGen",font_family="Poppinsbold",size=28),
    )

    appbar_e = ft.CupertinoAppBar(
        leading=ft.IconButton(ft.icons.KEYBOARD_BACKSPACE,on_click=lambda _:
page.go("/eng")),
        bgcolor=ft.colors.SURFACE_VARIANT,
        trailing=ft.Icon(ft.icons.FASTFOOD_OUTLINED),
      middle=ft.Text("FoodGen",font_family="Poppinsbold",size=28),
    )

    appbar_t = ft.CupertinoAppBar(
        leading=ft.IconButton(ft.icons.KEYBOARD_BACKSPACE,on_click=lambda _:
page.go("/tam")),
        bgcolor=ft.colors.SURFACE_VARIANT,
        trailing=ft.Icon(ft.icons.FASTFOOD_OUTLINED),
        middle=ft.Text("FoodGen",font_family="Poppinsbold",size=28),
    )


    # Page Routing start
    def route_change(route):
        page.views.clear()
        page.theme_mode = ft.ThemeMode.LIGHT
        page.theme = ft.theme.Theme(color_scheme_seed='#A233A2')
        page.views.append(
            ft.View(
                "/",
                [
                    splash_screen
                ],padding=0
            )
        )
```

7.1.2 recipe_e:

```
import flet as ft

#******INDIAN ENGLISH RECIPE ******#
goat_e = ft.Column(
[
    ft.Container(height=10),
    ft.Text(value="<GOAT CURRY>", font_family="Ancient", size=30),
    ft.Container(height=5),
    ft.Image(src="/images/goat.jpg", width=300),
    ft.Container(height=10),
    ft.Text("Recipe",font_family="Sticky",size=50),
    ft.Container(height=5),
    ft.Container(
        ft.Text('''4 to 8 dried red chilies (use low to spicy kind as per preference, deseed if preferred, or 1 to 1.5 tsp chili powder)
2 tablespoon coriander seeds
1 star anise (small)
1 teaspoon cumin seeds
1 teaspoon fennel seeds
5 green cardamoms (elaichi)
4 to 5 cloves
2 two inch cinnamon stick
½ teaspoon pepper corn
1 to 2 small portions kalpasi / stone flower/ dagad phool (optional)
3 sprigs of curry leaves'''),
        padding=ft.padding.only(left=10,right=10,bottom=20),
        width=400,
        height=400,
    )
   ], horizontal_alignment='center',scroll=ft.ScrollMode.HIDDEN
```

```
        )
        goat = ft.Container(
            content=goat_e,
            gradient=ft.LinearGradient(
                begin=ft.alignment.top_center,
                end=ft.alignment.bottom_center,
                colors=('#E4EEE9', '#93A5CE')
            ),
            width=400,
            height=800,
            padding=5
)
```

7.1.3 recipe_t:

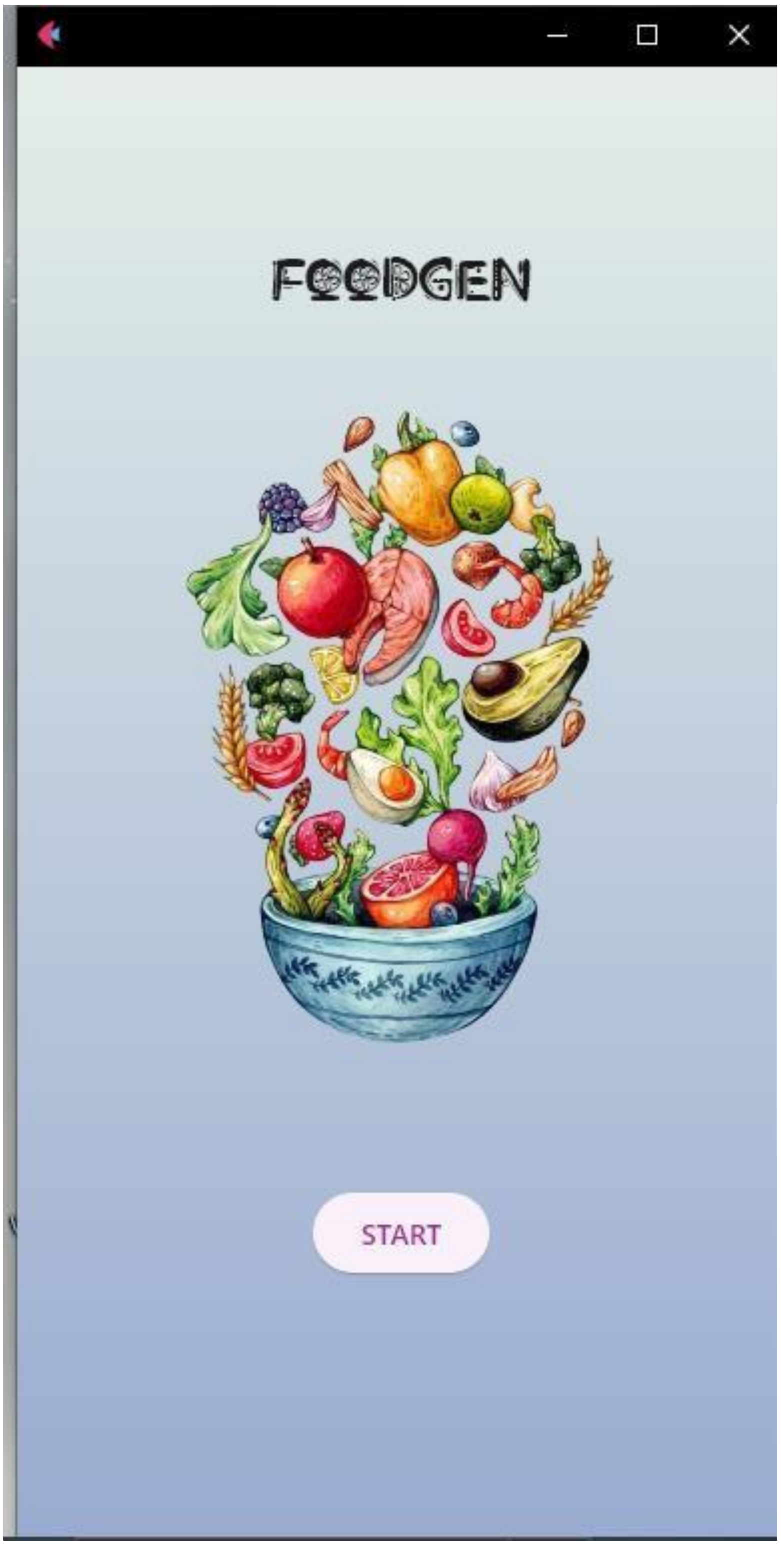
```
        import flet as ft

        #..............................India.................................#
        goat_t = ft.Column(
            [
                ft.Container(height=10),
                ft.Text(value="<ஆட்டுக்கறி>", font_family="Ancient", size=30),
                ft.Container(height=5),
                ft.Image(src="/images/goat.jpg", width=300),
                ft.Container(height=10),
                ft.Text("செய்
                முறை",font_family="Sticky",size=50),
                ft.Container(height=5),
                ft.Container(
                    ft.Text('''1) 4 முதல் 8 காய் ந்த சிவப்பு மிளகாய் (விரும் பினால்
        குறைந்த முதல் காரமான வறக, விருப்பப்பட்டால் டீசீட் அல்லF
        1முதல் 1.5 ததக்கரண் டி மிளகாய் ளே்)
        2) 2 ததக்கரண் டி சகாத்தமல்லி விறதகள்
        3) 1 நடெத்திர தொம்பு (சிறிய F)
        4) 1 ததக்கரண் டி சீரகம்
        5) 1 ததக்கரண் டி சபரூஞ் சீரகம் விறதகள
        6) 5 பெறெ ஏலக்காய் (எறலெசி)
        7) 4 முதல் 5 கிராம்பு
        8) 2 இரண் டு அங்குல இலவங்கப்பட்றட
        9) ½ ததக்கரண் டி மிளகு தொளம்
        10) 1 முதல் 2 சிறிய பகுதிகள் கல்பசி / கல் மலர் / தகட் பூல
        (விரும்பினால்)
        11) கறிதவப்பிறல 3 தளிரகள்'''),
                    padding=ft.padding.only(left=10,right=10,bottom=20),
                    width=400,
                    height=420,
                )
               ], horizontal_alignment='center',scroll=ft.ScrollMode.HIDDEN
        )goa_t = ft.Container( content=goat_t, gradient=ft.LinearGradient(
            begin=ft.alignment.top_center,
            end=ft.alignment.bottom_center,
            colors=('#E4EEE9', '#93A5CE')
          ),
          width=400,
          height=800,
          padding=5
  )
```

7.1.4 routing:

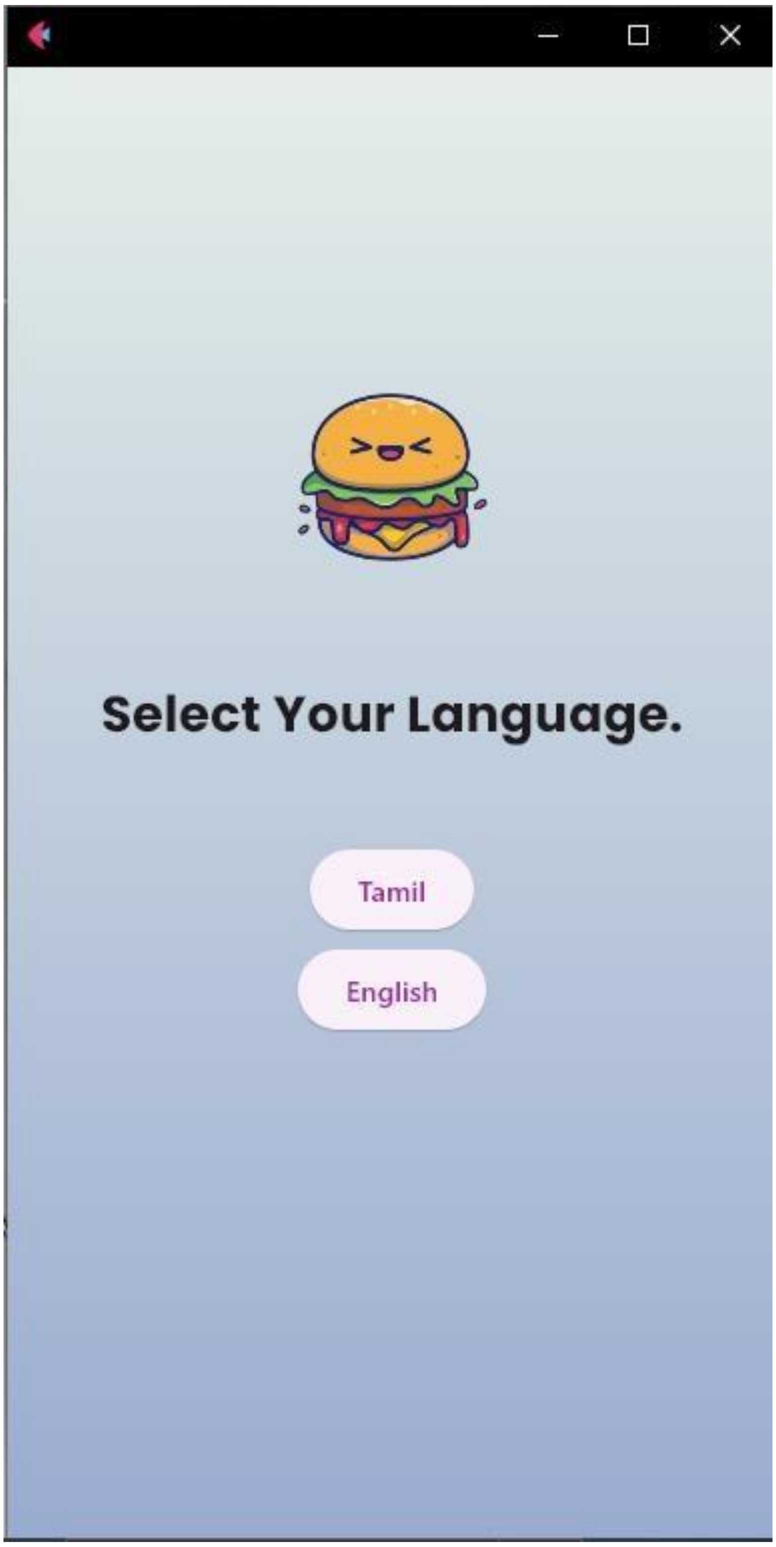
```
      def route_change(route):
          page.views.clear()
          page.theme_mode = ft.ThemeMode.LIGHT
          page.theme = ft.theme.Theme(color_scheme_seed='#A233A2')
          page.views.append(
            ft.View(
              "/",
              [
                splash_screen
              ],padding=0
            )
          )
          #.................................ENGLISH....................................#
          if page.route == "/lan":
            page.views.append(
            ft.View(
              "/lan",
              [
                langlogo
              ],
              padding=0,
              vertical_alignment="center",
              horizontal_alignment="center",
            )
          )
      )
        def view_pop(view):
          page.views.pop()
          top_view=page.views[-1]
          page.go(top_view.route)

        page.on_route_change=route_change
        page.on_view_pop=view_pop
  page.go(page.route)
```

7.2 Sample Screen Shot

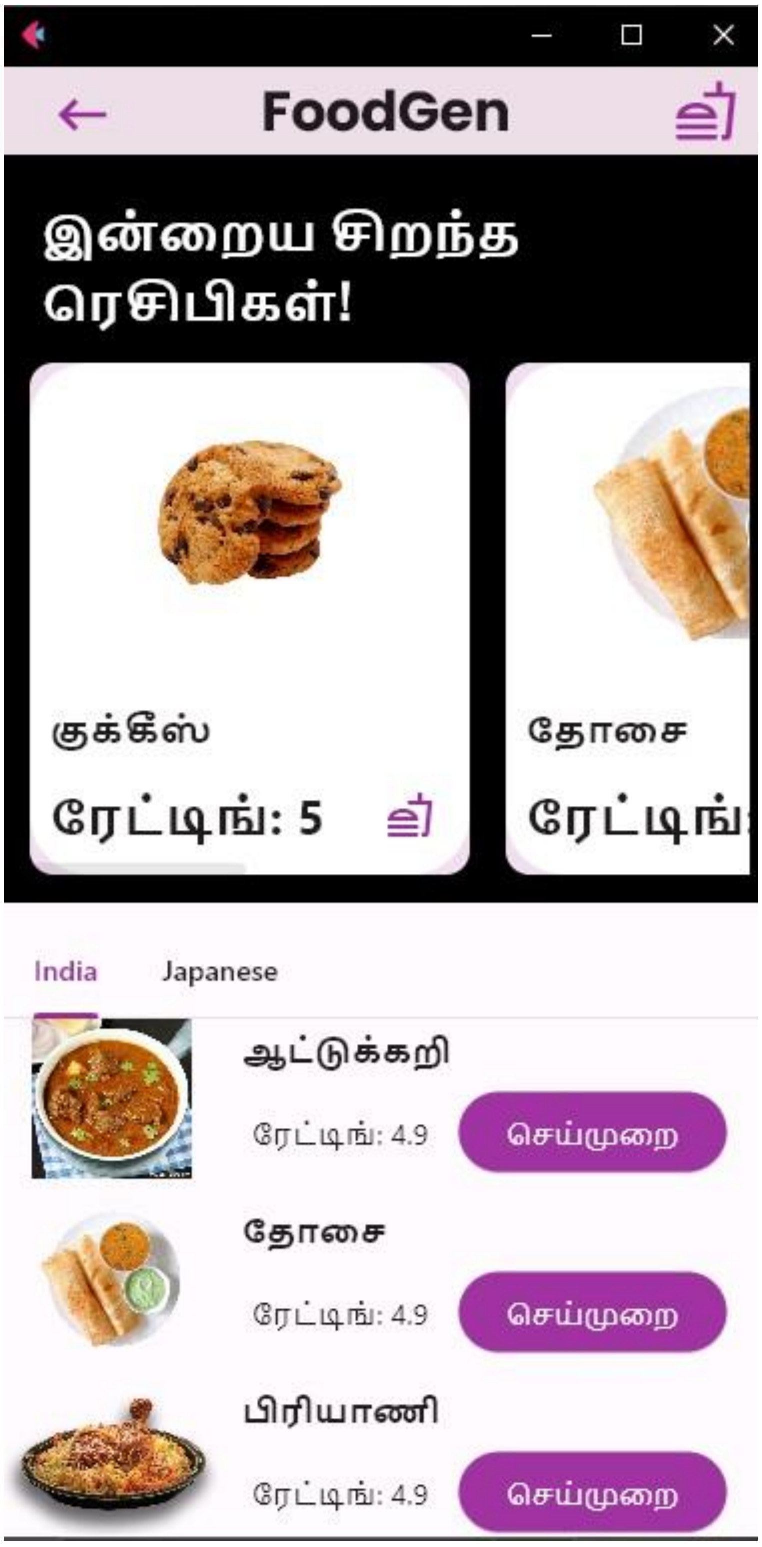
7.2.1 logo page:



7.2.3 Language Page:



7.2.3 Tamil Page:


FoodGen

இன்றைய சிறந்த ரெசிபிகள்!

குக்கீஸ்

ரேட்டிங்: 5

தோசை

ரேட்டிங்:

India Japanese

ஆட்டுக்கறி

ரேட்டிங்: 4.9 செய்முறை

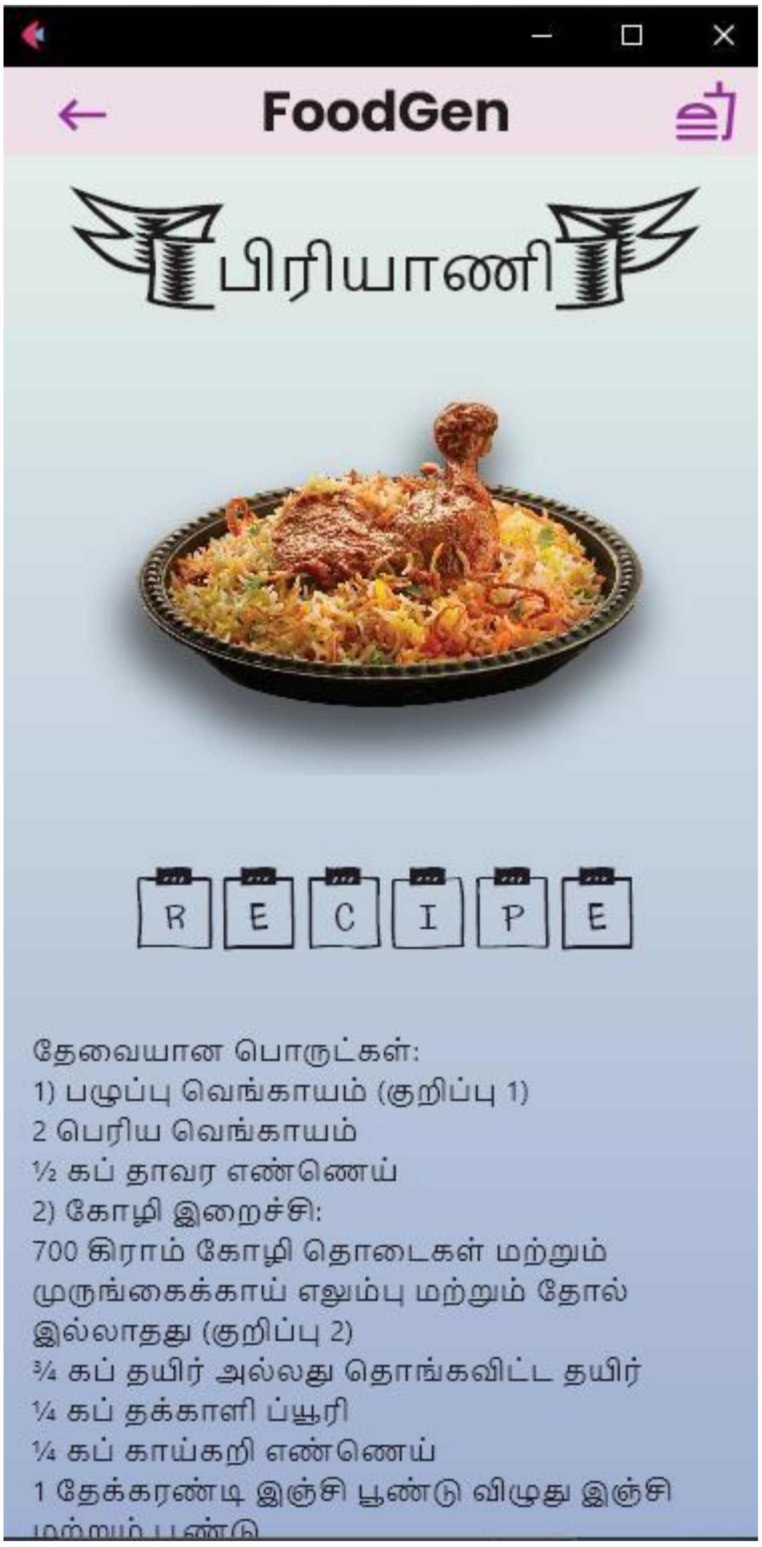
தோசை

ரேட்டிங்: 4.9 செய்முறை

பிரியாணி

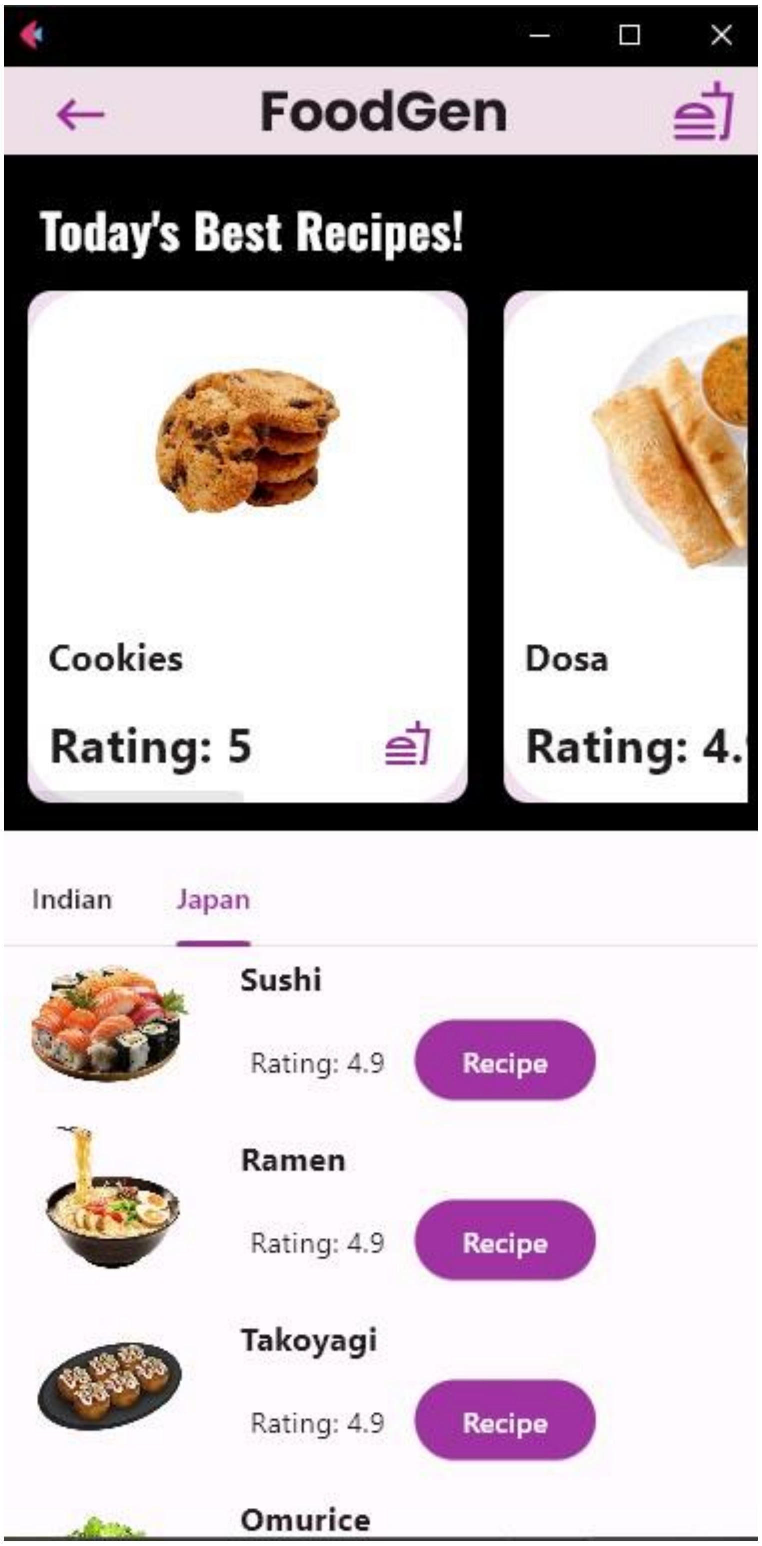
ரேட்டிங்: 4.9 செய்முறை

7.2.4 Tamil Recipe Page:



தேவையான பொருட்கள்:
1) பழுப்பு வெங்காயம் (குறிப்பு 1)
2 பெரிய வெங்காயம்
½ கப் தாவர எண்ணெய்
2) கோழி இறைச்சி:
700 கிராம் கோழி தொடைகள் மற்றும்
முருங்கைக்காய் எலும்பு மற்றும் தோல்
இல்லாதது (குறிப்பு 2)
¾ கப் தயிர் அல்லது தொங்கவிட்ட தயிர்
¼ கப் தக்காளி ப்யூரி
¼ கப் காய்கறி எண்ணெய்
1 தேக்கரண்டி இஞ்சி பூண்டு விழுது இஞ்சி
மற்றும் பூண்டு

7.2.5 English Page:



FoodGen

Today's Best Recipes!

Cookies

Rating: 5

Dosa

Rating: 4.

Indian

Japan

Sushi

Rating: 4.9

Recipe

Ramen

Rating: 4.9

Recipe

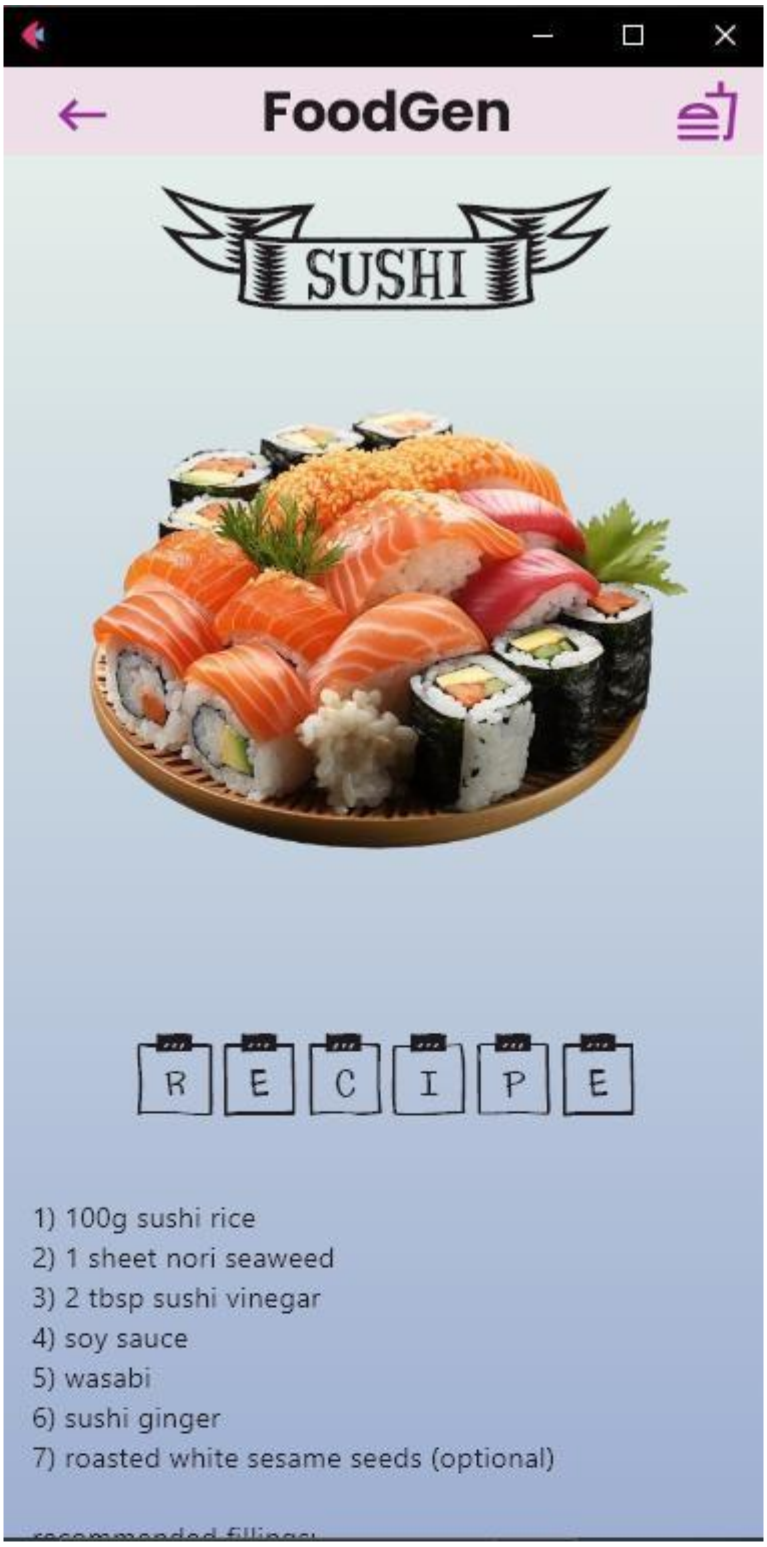
Takoyagi

Rating: 4.9

Recipe

Omurice

7.2.6 English Recipe Page:



1) 100g sushi rice
2) 1 sheet nori seaweed
3) 2 tbsp sushi vinegar
4) soy sauce
5) wasabi
6) sushi ginger
7) roasted white sesame seeds (optional)

recommended fillings:

8 CONCLUSION AND FUTURE ENHANCEMENT

8.1 Conclusion:

Food-Gen represents a significant advancement in the realm of mobile applications for culinary enthusiasts, offering a streamlined and user-friendly platform for recipe discovery. By harnessing the power of Python and the intuitive interface capabilities of Flet, Food-Gen simplifies the process of finding and preparing a wide array of dishes. The application's comprehensive database, personalized recommendations, and interactive features make it an essential tool for both novice cooks and seasoned chefs.

The development and implementation of Food-Gen address common limitations found in existing systems, such as time-consuming searches, lack of personalization, and poor user experiences. By providing a dynamic, responsive, and aesthetically pleasing interface, Food-Gen ensures that users can easily navigate through its features and find exactly what they need with minimal effort.

In summary, Food-Gen enhances the culinary experience by making recipe discovery quick, efficient, and enjoyable. It promotes culinary exploration and helps users expand their cooking repertoire, ultimately fostering a more engaging and rewarding cooking experience.

8.2 Future Enhancement:

Enhanced Personalization:

- **Machine Learning Integration:** Implement machine learning algorithms to provide even more accurate and personalized recipe recommendations based on user behavior, preferences, and dietary restrictions.
- **User Profiles:** Allow users to create detailed profiles, specifying their favorite cuisines, ingredient preferences, and cooking skill levels.

Voice-Activated Features:

- **Voice Search and Commands:** Integrate voice recognition technology to enable users to search for recipes and navigate the app using voice commands, providing a hands-free experience.

Social Features:

- **Community Sharing:** Enable users to share their own recipes, cooking tips, and experiences with the Food-Gen community, fostering a sense of community and shared learning.
- **Recipe Ratings and Reviews:** Allow users to rate and review recipes, helping others make informed choices and improving the overall quality of the recipe database.

Multilingual Support:

- **Language Options:** Add support for multiple languages to make Food-Gen accessible to a broader audience worldwide.