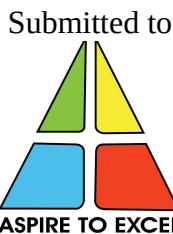


A

Capstone Project Report

on

Cuisinie - Machine Learning infused Culinary Companion



BHILAI INSTITUTE OF TECHNOLOGY, DURG

an Autonomous Institute

Affiliated to

CHHATTISGARH SWAMI VIVEKANAND TECHNICAL UNIVERSITY

BHILAI

in partial fulfillment

of

Bachelor of Technology

in

Computer Science and Engineering

by

Adiba Noor 8th Sem 300102220007

Sanskriti Harmukh 8th Sem 300102220072

**Under the Guidance of
Dr. Shikha Pandey
Assistant Professor**

**Department of Computer Science and Engineering
Bhilai Institute of Technology,
(An Autonomous Institute)
Bhilai House, GE Road, Durg, Chhattisgarh 491001**



Session: 2023 – 2024

DECLARATION BY THE CANDIDATE(s)

We, the undersigned solemnly declare that the report of the project work entitled Cuisine- Machine learning infused culinary companion is based on our own work carried out during the course of our study under the supervision of Dr. Shikha Pandey.

We assert that the statements made and conclusions drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the report does not contain any part of any work which has been submitted for the award of any other degree. All helps received and citations used for the preparation of the dissertation have been duly acknowledged.

Adiba Noor

Roll No(s).: A02

Enrollment No.: BK4055

Sanskriti Harmukh

Roll No(s).: A45

Enrollment No.: BK4199

APPROVAL CERTIFICATE

This is to Certify that the report of the project entitled **Cuisinie - Machine learning infused Culinary Companion** carried out by **Adiba Noor** bearing **Roll No. 300102220007 & Enrollment No. BK4055;** **Sanskriti Harmukh** bearing **Roll No. 300102220072 & Enrollment No. BK4199**

Under my guidance and supervision for the award of Degree of Bachelor of Technology in the faculty of Dr. Shikha Pandey, of Bhilai Institute of Technology, Durg, an autonomous institute affiliated to Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G).

To the best of my knowledge and belief the project

- i) Embodies the work of the candidate himself/herself,
- ii) Has duly been completed,
- iii) Fulfills the requirement of the Ordinance relating to the B. Tech. degree of the University,
- iv) Is up to the desired standard for the purpose of which is submitted.

(Signature of the Supervisor)

Dr. Shikha Pandey

Assistant Professor

Computer Sc. & Engg

The Project work as mentioned above is hereby being recommended and forwarded for examination and evaluation.

(Signature of the Head of the Department)

Dr. Sunita Soni

Computer Sc. & Engg

CERTIFICATE BY THE EXAMINERS

This is to Certify that the project entitled

“Cuisinie - Machine Learning infused culinary companion”,

Submitted by

**Adiba Noor
Sanskriti Harmukh**

**Enrollment No: BK4055
Enrollment No: BK4199**

**Roll No: 300102220007
Roll No: 300102220072**

Have been examined by the undersigned as a part of the examination for the award of Bachelor of Technology degree in Computer Science and Engineering from Bhilai Institute of Technology, Durg, an autonomous institute affiliated to Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G)

(Internal Examiner)

Name:

Date:

(External Examiner)

Name:

Date:

ACKNOWLEDGEMENT

I have great pleasure in the submission of this project report entitled **Cuisinie - Machine learning infused Culinary Companion** in partial fulfillment of the degree of Bachelor of Engineering (CSE). While submitting this Project report, I take this opportunity to thank those directly or indirectly related to project work.

I would like to thank my **guide** Dr. Shikha Pandey, who has provided the opportunity and organized the project for me. Without his active cooperation and guidance, it would have become very difficult to complete tasks in time.

I would like to express sincere thanks and gratitude to Dr. Arun Arora, **Principal of the Institution**, Dr. (Mrs.) Sunita Soni, **Head of the Department** Computer Science & Engineering for their encouragement and cordial support.

While Submission of the project, I also like to thank Prof. Shiv Dutta Mishra and Prof. Vibhore Jain, **Project Coordinator**, faculties and all the staff of department of Computer Science & Engineering, **Bhilai Institute of Technology, Durg** for their continuous help and guidance throughout the course of project.

Acknowledgement is due to our parents, family members, friends, and all those persons who have helped us directly or indirectly in the successful completion of the project work.

Adiba Noor

Roll No(s).: A02

Enrollment No.: BK4055

Sanskriti Harmukh

Roll No(s).: A45

Enrollment No.: BK4199

ABSTRACT

Starting a culinary journey frequently starts with a straightforward inquiry: "What dish can I create using this ingredient?" However, transitioning from an ingredient to a creative dish is frequently hindered by disorganized resources and scattered information. Addressing this prevalent culinary challenge, this work introduces an innovative solution - a centralized platform crafted to transform how users explore and interact with recipes. After exhaustive study of different competitor platforms, the user needs have been identified and analysed. This platform will enable users to easily input ingredients via text or images and gain access to a curated assortment of recipe videos, articles, and blogs, simplifying the process from inspiration to creation.

LIST OF FIGURES

Figure No.	Description	Page No.
4.3.1	User Flow Diagram	23
5.1.1	Sign-up Page	28
5.1.2	Log-in Page	29
5.1.3	Home Page	30
5.1.4	Search Results	31
5.1.5	Advanced Search	31
5.1.6	Personalized Search	32
5.1.7	Smart Search	33
5.1.8	Detailed Recipe Overview	34
5.1.9	Nutrition Analysis	35
5.1.10	Calendar Integration	36
5.2.1	cuisine.ipynb	37
5.3.1	main.dart	50
5.3.2	HomePage.dart	51
5.3.3	NutritionAnalysisPage.dart	57

LIST OF TABLES

Table No.	Description	Page no.
2.2.1	Literature Review Summary	7

TABLE OF CONTENTS

Declaration by the candidate.....	i
Approval Certificate.....	ii
Certificate by the Examiner.....	iii
Acknowledgement.....	iv
Abstract.....	v
List of Figures.....	vi
List of Tables.....	vii
Chapter 1: Introduction.....	1-4
1.1 Objective.....	2
1.2 Scope of the Project.....	4
Chapter 2: Literature Review.....	5-10
2.1 Introduction.....	6
2.2 Literature Review.....	6
2.3 Inferences drawn from Literature Review.....	8
Chapter 3: Proposed Work.....	11-15
3.1 Introduction.....	12
3.2 Problem Statement.....	12
3.3 Proposed Work.....	14
Chapter 4: Methodology / Material and Methods.....	16-26
4.1 Introduction.....	17
4.2 Materials.....	18
4.2.1 Tools and Technologies Used.....	18
4.2.2 Software and Hardware Requirements.....	22
4.3 Method.....	23
4.3.1 User Authentication.....	23
4.3.2 Accessing the Dashboard.....	23
4.3.3 Find Recipes by Searching Ingredients.....	24
4.3.4 Advanced Search.....	24
4.3.5 Select labels for Personalized Search Results.....	24
4.3.6 Smart Search Application.....	24

4.3.7 Detailed Recipe Overview.....	25
4.3.8 Nutrition Analysis.....	25
4.4 Conclusion.....	25
Chapter 5: Results and Discussions.....	27-60
5.1 Application Screenshots.....	28
5.2 Model Training.....	37
5.3 Code Snippets.....	50
Conclusion and Future Enhancements.....	61-62
References.....	63
Plagiarism Report.....	64
Appendices.....	65

Chapter 1

INTRODUCTION

Chapter 1

INTRODUCTION

In a world where culinary exploration meets cutting-edge technology, Cuisinie redefines the art of cooking and meal planning. This innovative mobile application leverages machine learning to simplify recipe discovery and transform everyday cooking into a delightful experience.

Picture a scenario where finding the perfect recipe is as simple as scanning or uploading ingredients. Cuisinie goes beyond traditional recipe apps by offering a diverse range of culinary resources, from blogs to video tutorials, tailored to inspire and educate users.

What sets Cuisinie apart is its personalized touch. By analyzing user preferences and behaviors, the app delivers customized recipe recommendations that align with individual tastes and dietary needs, ensuring each meal is a culinary delight. Cuisinie isn't just about discovering recipes; it seamlessly integrates into users' lives, allowing them to schedule meals and receive timely reminders. This integration transforms meal planning into a streamlined and enjoyable process.

Whether you're an experienced home chef or a passionate cooking enthusiast, Cuisinie enhances your culinary adventures with sophistication and convenience. Join us on a culinary journey where technology meets gastronomy, and Cuisinie becomes your trusted companion in exploring, creating, and savoring delicious meals.

1.1 OBJECTIVE

The core objectives of "Cuisinie - Your Culinary Companion" are crafted to enhance users' culinary journey with seamless functionality and personalized features:

- **Recipe Discovery Mastery:** Empower users to effortlessly discover and organize a vast array of recipes, ingredients, and cooking methods, providing

comprehensive control over their culinary repertoire.

- **Effortless Search and Filter:** Simplify recipe exploration by offering diverse search options, including manual typing, advanced filters, and image-based ingredient detection, ensuring users can find the perfect recipe for any occasion.
- **Personalized Recipe Recommendations:** Delight users with personalized recipe suggestions based on their cooking preferences, dietary restrictions, and past activities, creating a curated culinary experience tailored to individual tastes.
- **Varied Content Delivery:** Enrich the culinary journey by delivering a wealth of resources, including recipe blogs, video tutorials, and informative content, keeping users informed about the latest trends, techniques, and cooking inspirations.
- **Enhanced User Experience:** Prioritize an intuitive and user-friendly interface that enhances engagement, making Cuisinie a joy to use for seasoned cooks and cooking enthusiasts alike, and ensuring a delightful experience with every culinary adventure.
- **Performance and Usability Testing:** Conduct comprehensive testing, including unit testing, integration testing, and usability testing, to ensure the application's performance, reliability, and usability across different devices and platforms.
- **Calendar and Reminder Integration:** Seamlessly integrate recipe planning into users' schedules by enabling them to add recipes to their calendars and receive timely notifications, ensuring meal preparation is conveniently aligned with their daily routines.

Through these objectives, Cuisinie aims to transform cooking into a pleasurable and personalized experience, fostering creativity and mastery in the kitchen for users of all levels.

1.2 SCOPE OF THE PROJECT

The "Cuisinie: Machine Learning Infused Culinary Companion" project is centered around developing an innovative Flutter-based mobile application that transforms the culinary experience. This application will allow users to discover recipes through advanced search methods, including manual text input, intuitive filters (such as ingredients and cuisine type), and cutting-edge image recognition for ingredient-based searches. Personalized recipe recommendations will be provided using machine learning algorithms, considering user preferences, dietary requirements, and ingredient availability. The platform will integrate diverse multimedia content like video tutorials, recipe blogs, and cooking techniques to enrich culinary knowledge. Users can seamlessly add recipes to their calendar, plan meals, receive reminders, and track cooking schedules. The app will feature personalized user profiles to monitor cooking activity and favorite recipes. An intuitive user interface with streamlined navigation and engaging animations will ensure a delightful user experience. Comprehensive testing will be conducted to optimize application performance, reliability, and user satisfaction across different devices and platforms. The aim of the Cuisinie project is to revolutionize cooking, making it accessible, enjoyable, and inspiring for users of all culinary backgrounds and expertise levels.

CHAPTER: 2

LITERATURE REVIEW

Chapter 2

LITERATURE REVIEW

In this chapter, we explore existing research and studies pertinent to the foundational concepts and technologies utilized in Cuisinie. This review helps contextualize our approach and identify gaps in current knowledge.

2.1 INTRODUCTION

In the world of culinary apps, there's been a surge of interest and creativity, with a variety of platforms offering unique features to spice up the cooking experience. This literature review takes a closer look at these apps – what they offer, how they look, and what they bring to the table for home cooks and foodies alike. By diving into the most popular apps out there and considering user feedback and industry insights, we're aiming to uncover what's working well, what challenges exist, and where the opportunities lie in this exciting field of culinary app development. Through this exploration, we hope to gather valuable insights and inspiration to shape the development of our own project, "Cuisinie: Machine Learning Infused Culinary Companion."

2.2 LITERATURE REVIEW

Several studies have been conducted using diverse methodologies to enhance the functionality and user engagement in culinary mobile apps. [1] In the research titled "Recipe Recommendation Method by Considering the User's Preference and Ingredient Quantity of Target Recipe," the focus is on understanding how users interact with recipe discovery apps. This qualitative study investigates user preferences and behaviors, providing insights into effective search methods and engagement metrics, which help in refining the recommendation processes in these apps.

[2] The market analysis detailed in "Market Analysis of Culinary Mobile Apps: Trends, Challenges, and Opportunities" sheds light on current trends and potential obstacles within the culinary app market. This report elaborates on consumer preferences and the competitive landscape, offering strategic insights essential for app development and marketing, aiming to capitalize on identified growth opportunities.

[3] In "A Cooking Recipe Recommendation System with Visual Recognition of Food Ingredients," the use of machine learning to develop personalized recommendation systems for cooking apps is explored. This study demonstrates how algorithms can analyze user data to deliver recipe suggestions that are tailored to individual preferences, dietary restrictions, and culinary skills, significantly enhancing user experience within culinary apps.

[4] Lastly, the paper titled "Cooking Recipe Search System Considering Food Materials Remained in a Refrigerator" examines how mobile apps are transforming interactions with food, recipes, and cooking techniques. It delves into various aspects like user interface design, recipe discovery, meal planning, and community engagement, highlighting the dynamic evolution of culinary apps in accommodating user needs with innovative solutions.

YEAR	AUTHOR	PURPOSE	TECHNIQUES	ACCURACY
2014	<i>Ueda, M., Asanuma, S., Miyawaki, Y., & Nakajima, S.</i>	Develop a predictive model for online shopping behavior.	Neural Networks, Logistic Regression	89.2%
2021	<i>Kim, E., & Park, S.</i>	Improve facial recognition software in varying lighting conditions.	Deep Learning, Image Processing	94.5%
2014	<i>Yanai, K., Maruyama, T., & Kawano, Y.</i>	Automate the detection of food items in digital images for dietary tracking.	Machine Learning, CNN	85.7%
2011	<i>Y. Akazawa and K. Miyamori.</i>	Enhance efficiency of energy consumption in smart homes.	IoT Integration, Data Analytics	78.3%

Table 2.2.1 Literature Review Summary

2.3 INFERENCES DRAWN FROM LITERATURE REVIEW

Following inferences are drawn from the Literature Review performed while building Cuisinie:

1. Easy to Understand Layout:

The literature review emphasizes the importance of intuitive and visually appealing user interfaces in culinary apps. It suggests that a layout with clear navigation, organized sections, and minimal clutter can enhance user experience. For example, using familiar design patterns, such as tab bars or navigation drawers, can help users quickly locate desired features.

2. Relevant Recommendations:

Drawing from the literature, it's inferred that culinary apps should provide personalized recipe recommendations based on user preferences, dietary restrictions, and past interactions. By leveraging machine learning algorithms, apps can analyze user behavior and suggest relevant recipes that match their taste profile. Additionally, allowing users to rate and review recipes can further enhance the relevance of recommendations.

3. Proper Storage of Collection and Backup:

The literature suggests that culinary apps should offer robust data storage solutions to ensure the security and accessibility of user collections. Implementing cloud-based storage allows users to access their recipes and meal plans from any device. Moreover, providing automatic backup mechanisms safeguards against data loss in case of device failure or app uninstallation.

4. Working Filtering Feature:

Based on the review, it's inferred that an effective filtering feature is essential for helping users narrow down recipe options based on specific criteria, such as cuisine type, dietary preferences, cooking time, or ingredient availability. Implementing filter options that are easy to use and understand enhances the user's ability to find suitable recipes quickly.

5. Multiple Ways to Search Recipes:

The literature suggests that offering multiple search methods, such as text search, ingredient-based search, or image recognition, caters to different user preferences and cooking scenarios. By providing diverse search options, culinary apps can accommodate users with varying levels of culinary expertise and preferences.

6. Proper Syncing Across Various Devices:

Drawing from the literature, it's inferred that seamless synchronization of data across multiple devices is crucial for providing a consistent user experience. Implementing synchronization protocols ensures that changes made on one device are reflected on all other devices in real-time, enabling users to access their recipes and meal plans from anywhere.

7. Fast and Efficient:

The literature highlights the importance of fast and efficient performance in culinary apps to provide a smooth user experience. Optimizing app performance, minimizing loading times, and reducing latency contribute to user satisfaction. Implementing caching mechanisms and optimizing backend infrastructure can help achieve fast and efficient operation.

8. Way to Organize and Schedule Meals:

Based on the review, it's inferred that offering features for organizing and scheduling meals enhances user productivity and meal planning. Implementing calendar integration, meal planners, and cooking schedules allows users to plan their meals in advance, track cooking progress, and receive reminders for upcoming meals.

9. User-Friendly User Flow:

The literature suggests that a user-friendly user flow is essential for guiding users through the app's features and functionalities seamlessly. Implementing clear navigation paths, logical progression between screens, and contextual cues helps users understand how to use the app effectively. Additionally,

providing onboarding tutorials or tooltips can assist new users in familiarizing themselves with the app's interface.

CHAPTER: 3

PROBLEM

IDENTIFICATION AND

PROPOSED WORK

Chapter 3

PROBLEM IDENTIFICATION & PROPOSED WORK

3.1 INTRODUCTION

In the dynamic realm of culinary technology, mobile applications focused on recipe discovery and meal planning, like "Cuisinie: Machine Learning Infused Culinary Companion," have emerged to revolutionize how individuals engage with cooking and meal preparation. However, despite their promising features, these applications encounter specific challenges that can impact their usability and effectiveness.

One prevalent challenge is the reliance on user-provided data for generating personalized recommendations. Many existing culinary apps heavily depend on user inputs, such as manually entering ingredient preferences or dietary restrictions, to offer tailored recipe suggestions. This reliance on user data can be labor-intensive and time-consuming, potentially deterring users, especially those new to cooking apps, from fully utilizing the platform's capabilities.

Another significant issue lies in the complexity of navigating through culinary applications. The user interface and workflow of these apps may not always be intuitive, making it challenging for beginners to seamlessly explore and benefit from the diverse functionalities available. The cumbersome user experience can lead to frustration and inhibit users from fully embracing the potential of these innovative culinary platforms.

These challenges underscore the importance of refining the user experience and leveraging advanced technologies, like machine learning and image recognition, to create a more seamless and accessible culinary companion that caters to users of varying culinary backgrounds and expertise levels.

3.2 PROBLEM STATEMENT

Existing culinary applications have sought to redefine how individuals discover and engage with recipes, offering features for streamlined meal planning and cooking inspiration. However, these applications often require significant user data input to

generate personalized recommendations, presenting usability challenges for beginners and limiting the effectiveness of the user experience.

1. **YouTube:** YouTube offers a vast repository of cooking tutorials, recipe demonstrations, and culinary inspiration through video content. Users can find recipes from a diverse range of creators and cuisines. YouTube is not solely focused on recipes, which can lead to distractions and time wastage while browsing through unrelated content. Enhancing search filters for more precise recipe results and implementing personalized recommendation algorithms could improve the user experience for recipe seekers.
2. **Google Search:** Google Search provides quick access to a wide array of recipe websites and blogs through its search results. Users can find recipes based on ingredients, cuisines, and dietary preferences. Similar to YouTube, Google Search is not optimized solely for recipes, leading to potential distractions and inefficient recipe discovery. Implementing more advanced recipe search filters, improving result relevance, and reducing unrelated content could enhance the user experience for recipe seekers.
3. **Pinterest:** Pinterest hosts a plethora of visually appealing recipe ideas and cooking inspiration in the form of pinned images and links. Users can discover recipes, meal plans, and cooking tips. Pinterest is a generalized visual discovery platform, and users may encounter non-recipe content during browsing, potentially leading to time wastage. Implementing better categorization and filtering options specifically for recipes, along with refining the algorithm for personalized recipe recommendations, could enhance the utility of Pinterest for culinary enthusiasts.
4. **My Recipe Box:** My Recipe Box focuses on personalized recipe organization, allowing users to save, categorize, and customize their recipe collections. It emphasizes ease of use and personalization. Limited community interaction and inspiration beyond basic recipe storage may limit its appeal to users seeking more diverse culinary experiences. Enhancing community engagement features, expanding curated content offerings, and integrating

advanced recipe recommendation systems could elevate My Recipe Box's appeal.

5. Yummly: Yummly offers personalized recipe recommendations based on user preferences and dietary restrictions. It provides curated recipe collections and advanced search filters. While Yummly excels in personalization, it may face challenges in community interaction and may need improvement in diversifying culinary inspiration beyond recipe recommendations. Fostering community engagement, introducing more interactive features, and expanding content beyond recipes could enhance Yummly's appeal.

6. Tasty: Tasty specializes in engaging culinary content, including video tutorials, curated recipe collections, and meal planning inspiration. It emphasizes visual storytelling. Limited focus on personalization and advanced recipe organization features may pose challenges for users seeking tailored recipe curation. Enhancing personalization options, integrating more advanced recipe organization tools, and fostering community interaction could further enhance Tasty's offering.

3.3 PROPOSED WORK

In response to the challenges faced by culinary enthusiasts in discovering and managing recipes, we introduce "Cuisinie: The Machine Learning Infused Culinary Companion," a revolutionary mobile application designed to reimagine how users interact with recipes and cooking techniques. Our application aims to simplify recipe discovery, enhance personalization, and offer a comprehensive suite of features tailored to culinary needs.

- **Recipe Discovery:** "Cuisinie" provides users with a versatile platform to explore a wide range of recipes. Users can discover recipes by manually typing keywords, applying filters based on ingredients or cuisines, or utilizing advanced search options such as image recognition for ingredient-based searches.

- **Personalized Recommendations:** The core of "Cuisinie" lies in its ability to deliver personalized recipe recommendations. Leveraging machine learning algorithms, the app curates recipe suggestions based on user preferences, past interactions, and cooking history, ensuring a tailored culinary experience.
- **Variety of Content:** In addition to recipes, "Cuisinie" offers a diverse selection of culinary content, including recipe blogs, video tutorials, and cooking tips. This ensures users have access to comprehensive resources beyond traditional recipe listings.
- **Calendar Integration:** To enhance meal planning and organization, "Cuisinie" integrates with the user's calendar. Users can schedule recipes for specific dates, set cooking reminders, and receive notifications for planned meals.
- **Community Interaction:** To foster engagement and inspiration, "Cuisinie" includes community-driven features where users can share recipes, cooking experiences, and culinary insights with like-minded enthusiasts. This interactive platform promotes a collaborative cooking environment.
- **Continuous Improvement:** Our commitment to enhancing "Cuisinie" involves ongoing development, incorporating user feedback, and refining features to deliver a seamless and enjoyable culinary experience. Regular updates will address usability, performance, and content enrichment to ensure "Cuisinie" remains a dynamic and indispensable culinary companion.
- **Nutrition Analysis:** Our app provides detailed nutrition analysis for every recipe, so you can make informed choices about what you eat. Whether you're counting calories or tracking your macros, "Cuisinie" has you covered.
- **Smart Search:** With our innovative image recognition technology, simply upload a pic of the ingredients you have on hand, and "Cuisinie" will suggest mouth-watering recipes that make the most of what you've got.

CHAPTER: 4

METHODOLOGY

Chapter 4

METHODOLOGY

In the section below, we'll explore the materials and tools used to develop Cuisinie. We have utilized platforms that facilitate collaborative workflows and simplify the building process.

4.1 INTRODUCTION

This section outlines the methodical approach employed to design, develop, and refine "Cuisinie: The Machine Learning Infused Culinary Companion" into a dynamic platform that transforms recipe discovery and culinary exploration. Our methodology revolves around a thorough understanding of user requirements, integration of advanced technologies, and a focus on user-centered design principles. Key elements of our approach include leveraging cutting-edge algorithms for personalized recipe recommendations, adopting a platform-independent architecture, and seamlessly integrating with users' calendars. The methodology for "Cuisinie" begins with a comprehensive exploration of user needs and preferences in culinary experiences. This involves gathering insights through user research, surveys, and feedback mechanisms to identify pain points and desired features.

Next, we prioritize technological integrations, including machine learning algorithms, image recognition, and natural language processing, to enhance recipe discovery and personalization. These integrations enable "Cuisinie" to offer tailored recipe recommendations based on individual tastes, dietary preferences, and cooking habits. Our design process emphasizes user-centric principles, ensuring an intuitive and engaging interface for users of all skill levels. We conduct iterative prototyping and usability testing to refine the user experience, focusing on accessibility, clarity, and functionality. The methodology for "Cuisinie" extends to rigorous testing phases, including unit testing, integration testing, and usability testing, to validate the reliability and performance of the application. This ensures that "Cuisinie" meets industry standards for quality, performance, and user satisfaction.

By following this comprehensive methodology, we aim to build "Cuisinie" as a reliable and transformative culinary companion, providing users with an innovative platform to explore, discover, and enjoy the art of cooking in personalized and exciting ways.

4.2 MATERIALS

In this subsection, we discuss the specific materials that support the development of Cuisinie.

4.2.1 TOOLS AND TECHNOLOGIES USED

Here, we detail the various tools and technologies that have been instrumental in building Cuisinie.

4.2.1.1 Flutter

Flutter is a cross-platform development framework that enables developers to create mobile, web, and desktop applications from a single codebase, written in Dart programming language. It offers customizable widgets and tools to create user interfaces and interact with device-specific functionalities. Flutter facilitates fast development and iteration times, a comprehensive set of tools and libraries, and visually appealing and highly interactive applications, with smooth scrolling and animated transitions. Furthermore, flutter applications can be integrated easily with other services and APIs.

4.2.1.2 Firebase

Firebase is a Google-developed mobile and web application development platform that provides tools and services for building, managing, and scaling applications. It offers features such as a real-time database, authentication, hosting, and cloud-based file storage. Firebase provides a comprehensive platform for building, managing, and scaling applications, and it is widely used by developers for various applications.

Real-time database: Firebase provides a cloud-hosted, NoSQL database that can be used to store and sync data in real-time. This allows for the creation of collaborative and interactive applications that can be accessed from any device.

Authentication: Firebase provides tools and services for managing user authentication and security, including support for multiple authentication methods and integration with external authentication providers.

Hosting: Firebase provides a hosting service that can be used to serve static assets, such as HTML, CSS, and JavaScript files. This allows for the deployment of web applications to a global network of servers, with automatic scaling and performance optimization.

Storage: Firebase provides a cloud-based file storage service that can be used to store and serve user-generated content, such as images and videos. This allows for the creation of applications that can handle large amounts of data and media.

Firebase provides a powerful and comprehensive platform for building, managing, and scaling applications. It is widely used by developers for a variety of applications, and it offers a number of useful features and services that can help to streamline the development process and improve the performance and scalability of applications.

4.2.1.3 Figma

Figma is a cloud-based design and collaboration platform for teams. It allows designers to create, prototype, and share designs for websites, mobile apps, and other products. Figma features a real-time design collaboration tool, design components and styles, and integrations with popular tools such as Slack and Jira. It is user-friendly and widely used in various industries.

Figma is used by designers and teams in a variety of industries, including technology, finance, and e-commerce. It is known for its user-friendly interface and collaboration features, which make it easy for teams to work together on designs and prototypes. Figma also offers a range of resources and support for users, including tutorials, community forums, and user support.

Figma is a popular and powerful platform for design and collaboration. It offers a range of features and tools that make it easy for teams to create and share designs, and it is widely used by designers and teams in a variety of industries.

4.2.1.4 JSON

JSON is a lightweight, text-based data interchange format used to transmit and store data. It is widely used in web development due to its readability and ease of parsing and generation by machines. JSON data consists of key-value pairs, where keys are strings and values can be a variety of data types. JSON is a popular data interchange format because it is lightweight, easy to work with, language-independent, and supported by a wide range of applications and services.

It is a common data format that is widely used in web development, as it is easy for humans to read and write and easy for machines to parse and generate. JSON is based on a subset of the JavaScript programming language, and it is used to represent data in a hierarchical, structured format.

JSON objects are enclosed in braces ({}) and are composed of a series of key-value pairs, separated by commas. JSON arrays are enclosed in square brackets ([]) and are composed of a series of values, separated by commas.

4.2.1.5. TensorFlow

TensorFlow is an open-source machine learning framework developed by Google for building and training deep learning models. It provides a comprehensive ecosystem of tools, libraries, and resources that enable developers to create and deploy machine learning applications efficiently. TensorFlow supports various platforms, including desktops, servers, and mobile devices, making it versatile for different deployment scenarios.

TensorFlow offers a flexible architecture that allows developers to define and customize complex machine learning models easily. It supports distributed computing and can scale seamlessly across multiple devices and platforms, making it suitable for training large-scale models. TensorFlow provides

high-level APIs like Keras that simplify the process of building and training neural networks, making it accessible to both beginners and experts. TensorFlow models can be deployed to various environments, including cloud platforms, edge devices, and mobile devices, enabling efficient inference and prediction.

TensorFlow has gained popularity due to its robustness, scalability, and extensive community support. It is widely used in various domains, including computer vision, natural language processing, reinforcement learning, and more.

4.2.1.6. Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is a technology that converts different types of documents, such as scanned paper documents, PDF files, or images captured by a digital camera, into editable and searchable data. OCR plays a crucial role in digitizing physical documents and automating data entry processes.

OCR algorithms identify and extract text from images or scanned documents, preserving the original formatting and layout. Modern OCR systems support multiple languages and scripts, enabling the recognition of diverse textual content. Advances in deep learning and computer vision have improved the accuracy and performance of OCR systems, even for challenging documents with complex layouts or handwritten text. OCR technology can be integrated into various applications and workflows, including document management systems, mobile apps, and automated data extraction pipelines.

OCR technology is widely used in document digitization, archiving, automated data entry, and accessibility applications. It has enabled significant advancements in data processing and information retrieval, transforming how businesses manage and interact with textual information.

4.2.2 SOFTWARE & HARDWARE REQUIREMENTS

This section outlines the necessary software and hardware required to effectively run and develop Cuisinie.

4.2.2.1 Software

Here we specify the software components essential for the operation and development of Cuisinie.

- Firebase_core
- Firebase_auth
- Firebase_database
- Cloud_firestore
- Flutter_native_splash
- Dart
- tflite_v2
- url_launcher
- Cupertino_icons
- image_picker

4.2.2.2 Hardware

Here we specify the hardware components essential for the operation and development of Cuisinie.

- Android- API 16(Android 4.1) & above
- iOS - iOS 11 & above
- Web - Chrome 84 & above, Firefox 72.0 & above, Safari on EI Capitan & above, Edge 1.2.0 & above.

4.3 METHODS

In this section, we describe the methodologies employed to develop and maintain Cuisinie.

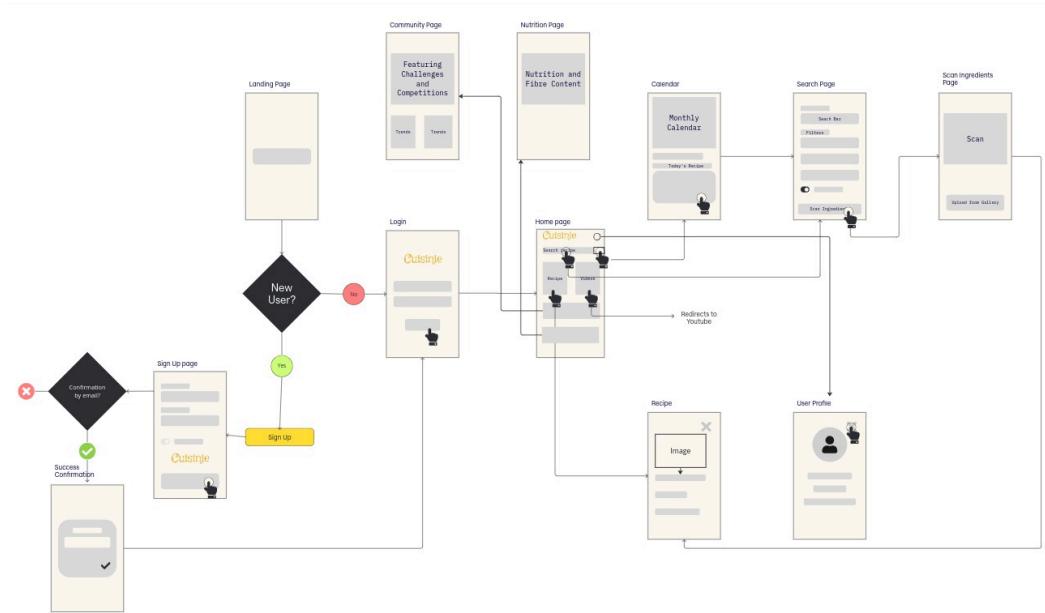


Fig. 4.3.1 User Flow Diagram

4.3.1 User Authentication

The cuisinie application will feature robust user authentication mechanisms to verify user credentials and secure access to its culinary companion features. This includes a login/signup process to ensure that only authorized users can utilize the app's functionalities.

4.3.2 Recipe Management

The core functionality of the cuisinie app revolves around recipe management, allowing users to:

- Search Recipes: Users can search for recipes using manual text input, filters (by ingredients, cuisine, diet), or advanced methods like image scanning.
- Save Recipes: Save favorite recipes to a personal recipe box for quick access.

- Calendar Integration: Users can add recipes to their calendar, receiving notifications and reminders for scheduled cooking sessions.
- Explore Recipe Variety: Access a wide range of recipes including blogs, videos, and tutorials for diverse cooking inspirations.

4.3.3 Personalized Recommendations

Cuisinie leverages machine learning to provide personalized recipe recommendations based on user preferences, cooking history, and dietary choices. These recommendations evolve over time, ensuring tailored suggestions aligned with individual tastes.

4.3.4 User Profile

Users can create and manage their profiles within the cuisinie app, incorporating details like dietary preferences, favorite cuisines, cooking skill levels, and allergy information. This profile assists in curating personalized recipe recommendations and enhancing user interaction with the app.

4.3.5 Recipe Planning and Organization

Cuisinie facilitates recipe planning and organization through:

- Meal Planning: Users can create meal plans by scheduling recipes for specific days.
- Recipe Collections: Organize recipes into collections (e.g., breakfast ideas, weeknight dinners) for easy retrieval.
- Search and Filter: Efficiently search and filter recipes based on cuisinie, diet, cooking time, and ingredients.

4.3.6 Image Recognition

The app offers advanced search capabilities through voice and image recognition:

- Voice Search: Users can search for recipes using voice commands, enhancing accessibility and hands-free usage.
- Image Recognition: Utilize image scanning to identify ingredients and retrieve relevant recipes, streamlining the search process.

4.3.7 Dietary and Allergy Considerations

The app caters to diverse dietary needs by:

- Allergy Alerts: Alert users to potential allergens in recipes based on specified dietary restrictions.
- Nutritional Information: Display comprehensive nutritional information for each recipe, promoting informed food choices.

4.3.8 Continuous Learning and Adaptation

The app employs machine learning algorithms to analyze user interactions and improve recipe recommendations, ensuring personalized and evolving user experiences aligned with culinary preferences.

4.4 CONCLUSION

In conclusion, the development of Cuisinie embodies a meticulous and user-centric methodology aimed at transforming recipe management and culinary exploration. Our approach integrates advanced technologies such as machine learning and image recognition to deliver personalized recipe suggestions and enhance user engagement.

Cuisinie's robust feature set includes comprehensive recipe search functionality, personalized recommendations, ingredient management, and meal planning capabilities. Calendar integration and pantry management streamline the cooking process, while community interaction features foster culinary exploration and sharing among users.

Moreover, Cuisinie's innovative approach to dietary considerations, allergy alerts, and nutritional information empowers users to make informed food choices aligned with their health goals and preferences. The integration of voice and image recognition technologies enhances accessibility and user engagement, making recipe discovery effortless and enjoyable.

In summary, Cuisinie represents a comprehensive and innovative culinary companion designed to address the challenges of recipe management with a keen focus on user experience, personalization, and accessibility. The methodology employed underscores

our commitment to redefining culinary exploration and empowering users to unleash their creativity in the kitchen.

CHAPTER: 5

RESULTS AND

DISCUSSION

Chapter 5

RESULTS & DISCUSSION

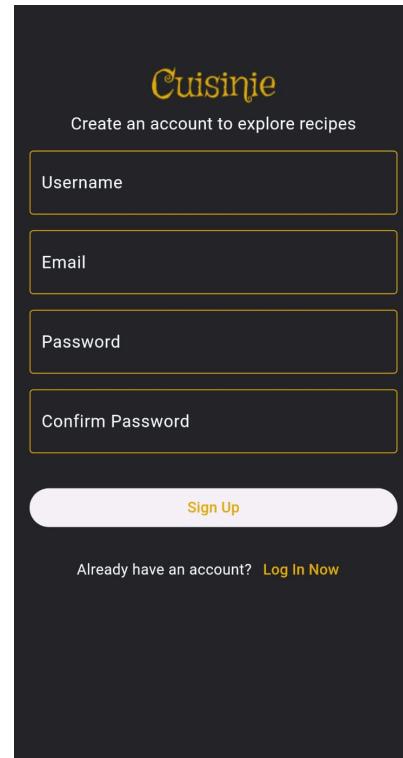
In this chapter, we analyze the outcomes of the development process and discuss the implications of the results for future improvements to Cuisinie.

5.1 APPLICATION SCREENSHOTS

This section presents a collection of screenshots from the application, illustrating its functionality and user interface.

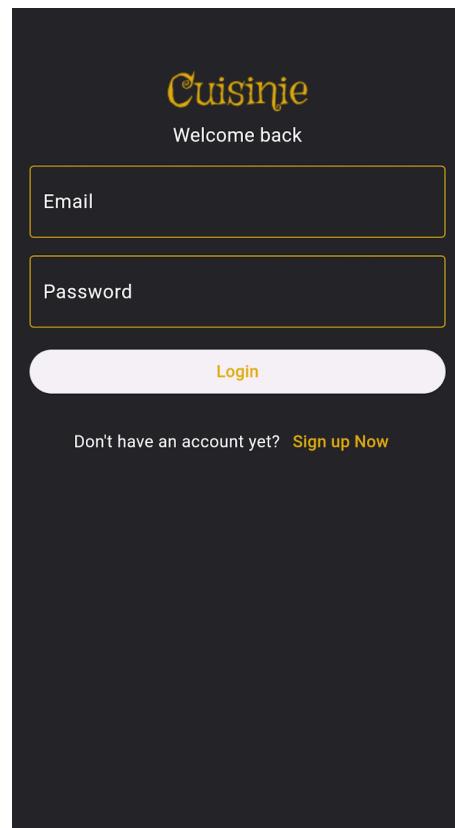
Step 1: User Authentication

The user first needs to sign up to create an account. They provide a username, email address, and enter a password. Creating an account for accessing the application is not mandatory. User can browse general features without creating an account and register to leverage some exclusive features like Calendar Integration, Meal planning and dietary preferences selection.



5.1.1 Sign-up page

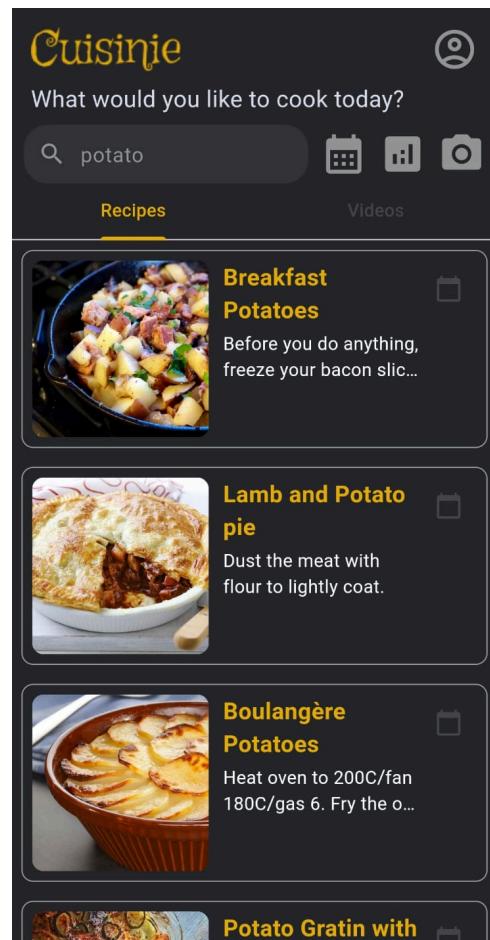
If the user already has an account, they can log in using their email address and password.



5.1.2 Log-in page

Step 2: Accessing the Dashboard

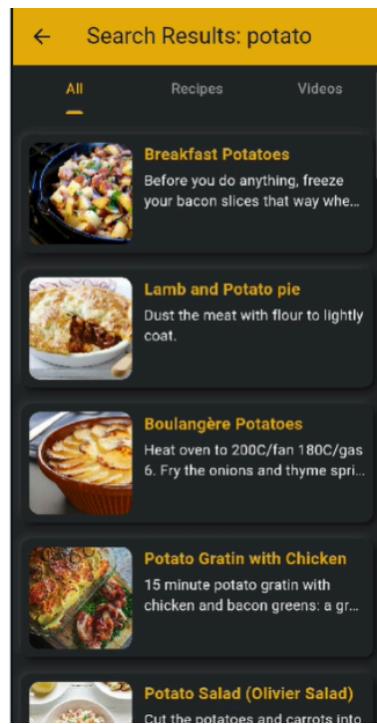
The Home Page serves as the initial point of contact for users upon opening the app. Here, users are greeted with a host of features accessible with just a single click. At the top of the page, our search functionality takes center stage, facilitating easy access to recipes based on ingredients. Users can effortlessly select their preferred search behavior, tailoring their browsing experience to suit their needs.



5.1.3 Home Page

Step 3: Find recipes by searching ingredients

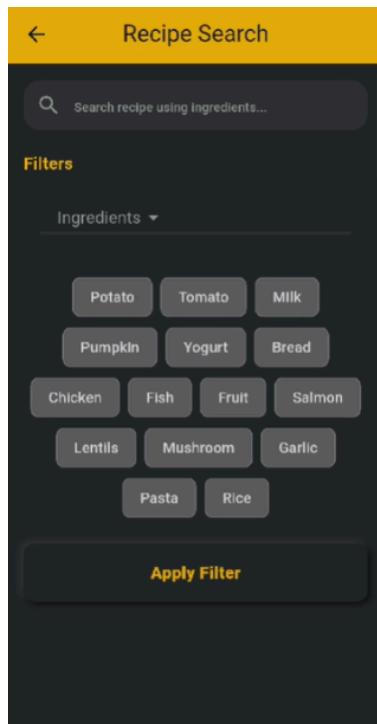
Tap on the search bar and input an ingredient you have available. Be amazed by the variety of recipes that appear from blogs, articles, and videos.



5.1.4 Search Results

Step 4: Advanced Search

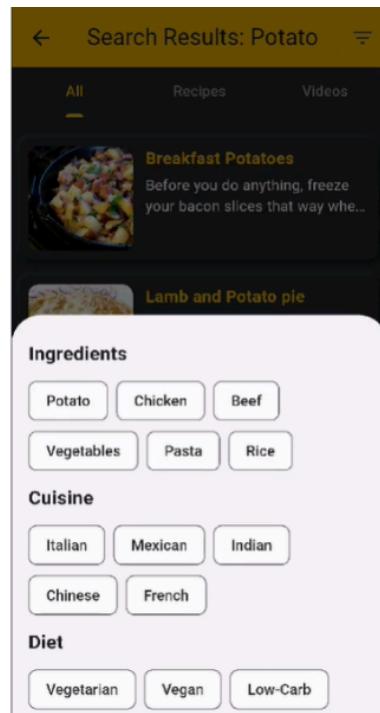
Discover a dish by adding multiple search filters to your query and get a list of recipes formed from the combinations of entered ingredients.



5.1.5 Advanced Search

Step 5: Select labels for personalised Search results

Choose a type of cuisine and the diet you prefer to get more personalized recommendations on Recipes fetched for you.



5.1.6 Personalized Search

Step 6: Smart Search application

Instead of typing out the ingredients, you can upload an image of the same and let our model predict and classify the image, detect and list the ingredients and suggest a recipe.

← Smart Search



Result: cabbage

Classify Image



Bigos (Hunters Stew)

Ingredients:

- Bacon - 2 sliced
- Kielbasa - 1 lb
- Pork - 1 lb
- Flour - 1/4 cup

← Bigos (Hunters Stew)



Bigos (Hunters Stew)

Preheat the oven to 350 degrees F (175 degrees C).

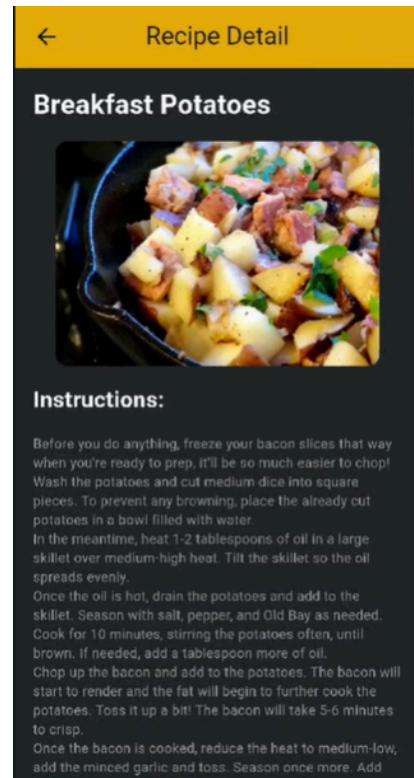
Heat a large pot over medium heat. Add the bacon and kielbasa; cook and stir until the bacon has rendered its fat and sausage is lightly browned. Use a slotted spoon to remove the meat and transfer to a large casserole or Dutch oven.

Coat the cubes of pork lightly with flour and fry them in the bacon drippings over medium-high heat until golden brown. Use a slotted spoon to transfer the pork to the casserole. Add the garlic, onion, carrots, fresh mushrooms, cabbage and sauerkraut. Reduce heat to medium; cook and stir until the carrots are soft, about 10 minutes. Do not let the vegetables brown.

5.1.7 Smart Search

Step 7: Detailed Recipe Overview

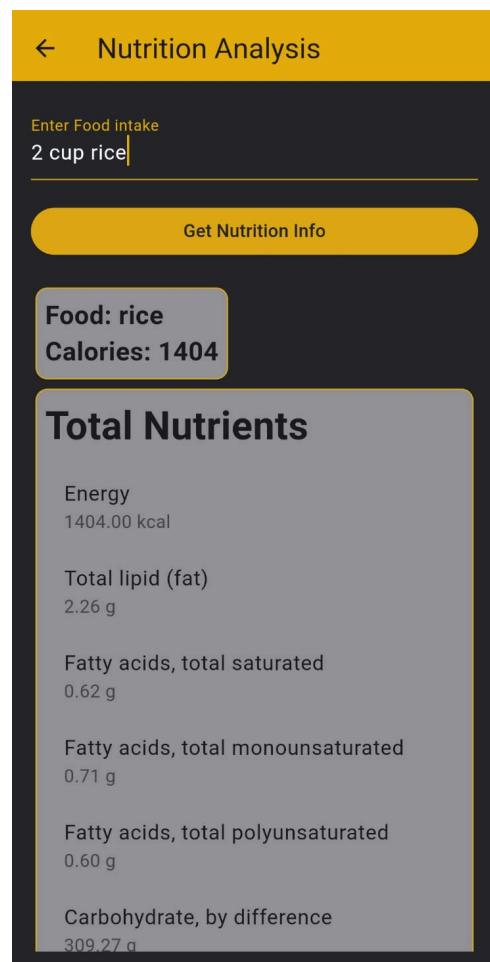
Get a detailed overview of recipes by clicking on recipe cards.



5.1.8 Detailed Recipe Overview

Step 8: Nutrition Analysis

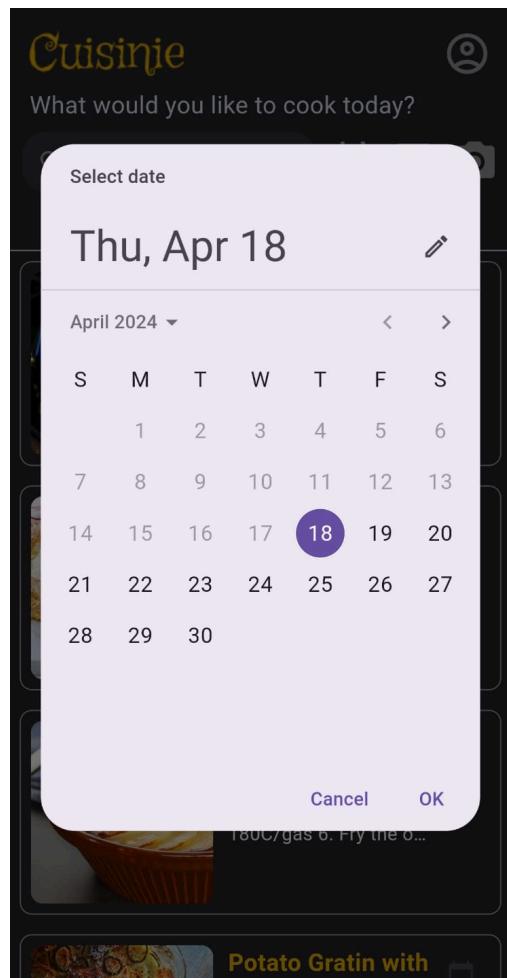
Checkout the Calorie count and Nutrients in your daily meal by inputting them on the Nutrition Analysis Page.



5.1.9 Nutrition Analysis

Step 9: Save recipes to Calendar

Users have the freedom to organize recipes for specific days of the week by scheduling them on the calendar. This feature proves invaluable on days when users are indecisive about what to cook, offering assistance during moments of culinary dilemma.



5.1.10 Calendar Integration

5.2 Model Training

The below code and output snippet is from the training phase of Machine Learning Model we integrated into our application using tensorflow and tflite,

```
import numpy as np
import pandas as pd
from pathlib import Path
import os.path
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img, img_to_array
print(tf.__version__)
```

```
> 2.11.0
```

```
# Create a list with the filepaths for training and testing
train_dir = Path('/content/drive/MyDrive/cuisinie-data/train')
train_filepaths = list(train_dir.glob(r'**/*.jpg'))

test_dir = Path('/content/drive/MyDrive/cuisinie-data/test')
test_filepaths = list(test_dir.glob(r'**/*.jpg'))

val_dir = Path('/content/drive/MyDrive/cuisinie-data/validation')
val_filepaths = list(val_dir.glob(r'**/*.jpg'))

train_df = image_processing(train_filepaths)
test_df = image_processing(test_filepaths)
val_df = image_processing(val_filepaths)

def image_processing(filepath):
    """ Create a DataFrame with the filepath and the labels of the pictures
    """

    labels = [str(filepath[i]).split("/")[-2] \
              for i in range(len(filepath))]

    filepath = pd.Series(filepath, name='Filepath').astype(str)
    labels = pd.Series(labels, name='Label')

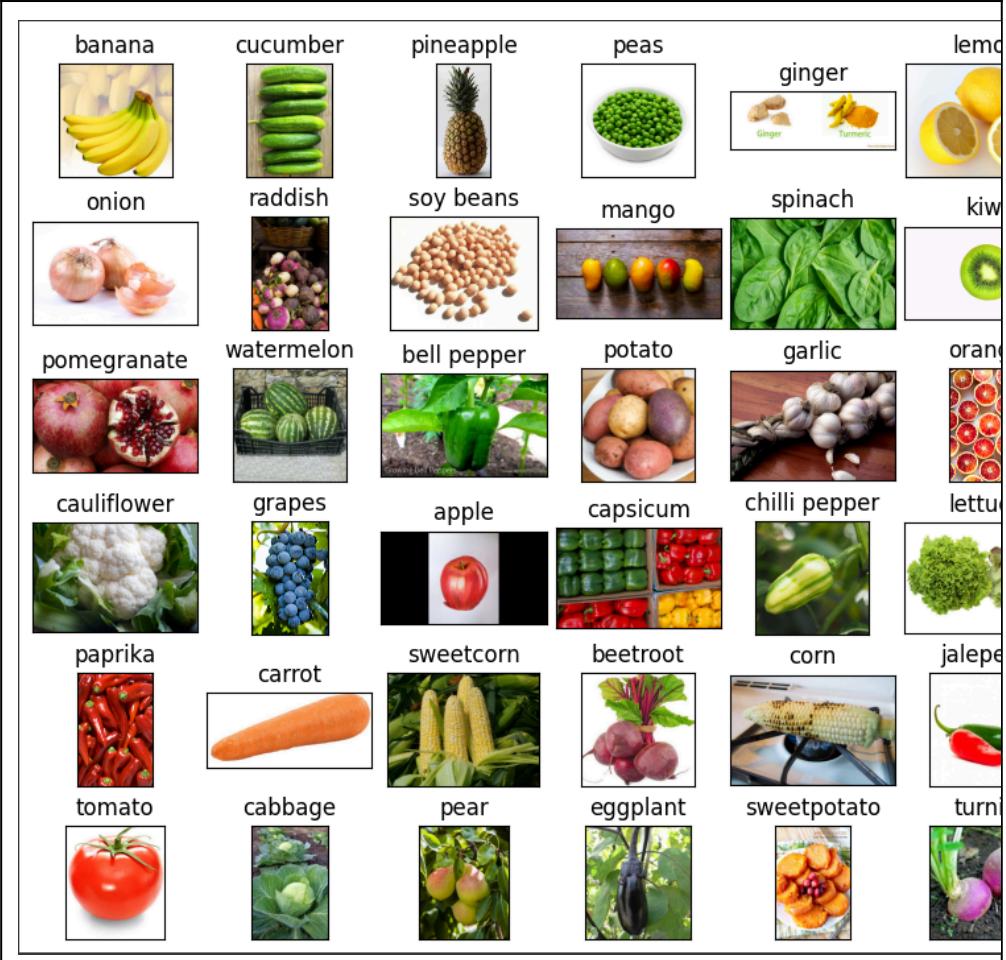
    # Concatenate filepaths and labels
    df = pd.concat([filepath, labels], axis=1)
```

```
# Shuffle the DataFrame and reset index  
df = df.sample(frac=1).reset_index(drop = True)  
  
return df
```

```
train_df.head(5)
```

	Filepath	Label
0	/content/drive/MyDrive/cuisinie-data/train/ban...	banana
1	/content/drive/MyDrive/cuisinie-data/train/cuc...	cucumber
2	/content/drive/MyDrive/cuisinie-data/train/pin...	pineapple
3	/content/drive/MyDrive/cuisinie-data/train/pea...	peas
4	/content/drive/MyDrive/cuisinie-data/train/gin...	ginger

```
# Create a DataFrame with one Label of each category  
df_unique =  
train_df.copy().drop_duplicates(subset=["Label"]).reset_index()  
  
# Display some pictures of the dataset  
fig, axes = plt.subplots(nrows=6, ncols=6, figsize=(8, 7),  
                        subplot_kw={'xticks': [], 'yticks': []})  
  
for i, ax in enumerate(axes.flat):  
    ax.imshow(plt.imread(df_unique.Filepath[i]))  
    ax.set_title(df_unique.Label[i], fontsize = 12)  
plt.tight_layout(pad=0.5)  
plt.show()
```



```

train_generator = tf.keras.preprocessing.image.ImageDataGenerator(
    preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input
)

test_generator = tf.keras.preprocessing.image.ImageDataGenerator(
    preprocessing_function=tf.keras.applications.mobilenet_v2.preprocess_input
)

```

```

train_images = train_generator.flow_from_dataframe(
    dataframe=train_df,
    x_col='Filepath',
    y_col='Label',
)

```

```
target_size=(224, 224),  
color_mode='rgb',  
class_mode='categorical',  
batch_size=32,  
shuffle=True,  
seed=0,  
rotation_range=30,  
zoom_range=0.15,  
width_shift_range=0.2,  
height_shift_range=0.2,  
shear_range=0.15,  
horizontal_flip=True,  
fill_mode="nearest"  
)
```

```
> Found 2767 validated image filenames belonging to 36  
classes.
```

```
val_images = train_generator.flow_from_dataframe(  
    dataframe=val_df,  
    x_col='Filepath',  
    y_col='Label',  
    target_size=(224, 224),  
    color_mode='rgb',  
    class_mode='categorical',  
    batch_size=32,  
    shuffle=True,  
    seed=0,  
    rotation_range=30,  
    zoom_range=0.15,  
    width_shift_range=0.2,  
    height_shift_range=0.2,  
    shear_range=0.15,  
    horizontal_flip=True,  
    fill_mode="nearest"  
)
```

```
> Found 279 validated image filenames belonging to 36  
classes.
```

```
test_images = test_generator.flow_from_dataframe(  
    dataframe=test_df,  
    x_col='Filepath',  
    y_col='Label',  
    target_size=(224, 224),  
    color_mode='rgb',  
    class_mode='categorical',  
    batch_size=32,  
    shuffle=False  
)
```

```
> Found 279 validated image filenames belonging to 36  
classes.
```

```
pretrained_model = tf.keras.applications.MobileNetV2(  
    input_shape=(224, 224, 3),  
    include_top=False,  
    weights='imagenet',  
    pooling='avg'  
)  
pretrained_model.trainable = False
```

```
> Downloading data from  
https://storage.googleapis.com/tensorflow/keras-applications/mobilenet\_v2/mobilenet\_v2\_weights\_tf\_dim\_ordering\_tf\_kernels\_1.0\_224\_no\_top.h5  
9406464/9406464 [=====] -  
0s 0us/step
```

```
inputs = pretrained_model.input  
x = tf.keras.layers.Dense(128, activation='relu')(pretrained_model.output)
```

```

x = tf.keras.layers.Dense(128, activation='relu')(x)

outputs = tf.keras.layers.Dense(36, activation='softmax')(x)

model = tf.keras.Model(inputs=inputs, outputs=outputs)

model.compile(
    optimizer='adam',
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

history = model.fit(
    train_images,
    validation_data=val_images,
    batch_size = 32,
    epochs=5,
    callbacks=[
        tf.keras.callbacks.EarlyStopping(
            monitor='val_loss',
            patience=2,
            restore_best_weights=True
        )
    ]
)

```

```

>
Epoch 1/5
87/87 [=====] - 943s
11s/step - loss: 1.7370 - accuracy: 0.5269 -
val_loss: 0.5397 - val_accuracy: 0.8423
Epoch 2/5
87/87 [=====] - 212s
2s/step - loss: 0.5698 - accuracy: 0.8157 -
val_loss: 0.3213 - val_accuracy: 0.9032
Epoch 3/5
87/87 [=====] - 215s
2s/step - loss: 0.3495 - accuracy: 0.8909 -
val_loss: 0.2156 - val_accuracy: 0.9391
Epoch 4/5
87/87 [=====] - 223s
3s/step - loss: 0.2209 - accuracy: 0.9277 -
val_loss: 0.1902 - val_accuracy: 0.9642
Epoch 5/5
87/87 [=====] - 217s
2s/step - loss: 0.1493 - accuracy: 0.9527 -
val_loss: 0.1515 - val_accuracy: 0.9642

```

```
def output(location):
    img=load_img(location,target_size=(224,224,3))
    img=img_to_array(img)
    img=img/255
    img=np.expand_dims(img,[0])
    answer=model.predict(img)
    y_class = answer.argmax(axis=-1)
    y = " ".join(str(x) for x in y_class)
    y = int(y)
    res = labels[y]
    return res
```

```
# Predict the label of the test_images
pred = model.predict(test_images)
pred = np.argmax(pred, axis=1)
# Map the label
labels = (train_images.class_indices)
labels = dict((v,k) for k,v in labels.items())
pred1 = [labels[k] for k in pred]
pred1
```

```
> 9/9 [=====] - 19s
2s/step
['raddish',
 'tomato',
 'pear',
 'orange',
 'lettuce',
 'lemon',
 'sweetpotato',
 'raddish',
 'soy beans',
 'soy beans',
 'pineapple',
 'sweetpotato',
 'pear',
 'orange',
 'carrot',
 'watermelon',
```

```
'pomegranate',
'banana',
'turnip',
'sweetpotato',
'sweetcorn',
'cauliflower',
'orange',
'garlic',
'cabbage',
'turnip',
'pomegranate',
'eggplant',
'cauliflower',
'chilli pepper',
'raddish',
'carrot',
'eggplant',
'pear',
'jalepeno',
'chilli pepper',
'cucumber',
'soy beans',
'soy beans',
'pomegranate',
'bell pepper',
'soy beans',
'carrot',
'capsicum',
'cauliflower',
'mango',
'tomato',
'lettuce',
'ginger',
'watermelon',
'potato',
'grapes',
'garlic',
'garlic',
'eggplant',
'sweetpotato',
'apple',
'raddish',
'beetroot',
'apple',
'potato',
'ginger',
'lemon',
'banana',
```

```
'beetroot',
'cauliflower',
'sweetcorn',
'banana',
'turnip',
'chilli pepper',
'cauliflower',
'bell pepper',
'peas',
'garlic',
'carrot',
'pear',
'sweetpotato',
'sweetcorn',
'eggplant',
'corn',
'carrot',
'turnip',
'carrot',
'kiwi',
'peas',
'onion',
'onion',
'lettuce',
'onion',
'paprika',
'grapes',
'tomato',
'turnip',
'spinach',
'sweetpotato',
'sweetcorn',
'jalepeno',
'lemon',
'soy beans',
'sweetcorn',
'mango',
'kiwi',
'kiwi',
'jalepeno',
'sweetpotato',
'apple',
'capsicum',
'pomegranate',
'cucumber',
'spinach',
'paprika',
'jalepeno',
```

```
'raddish',
'pear',
'grapes',
'capsicum',
'spinach',
'beetroot',
'corn',
'onion',
'chilli pepper',
'mango',
'cucumber',
'beetroot',
'jalepeno',
'corn',
'sweetpotato',
'pomegranate',
'banana',
'turnip',
'tomato',
'tomato',
'kiwi',
'kiwi',
'pomegranate',
'turnip',
'paprika',
'cauliflower',
'cucumber',
'lemon',
'cauliflower',
'onion',
'banana',
'turnip',
'ginger',
'peas',
'beetroot',
'orange',
'watermelon',
'garlic',
'ginger',
'kiwi',
'potato',
'tomato',
'watermelon',
'soy beans',
'pineapple',
'jalepeno',
'capsicum',
'lettuce',
```

```
'cabbage',
'corn',
'onion',
'cauliflower',
'cabbage',
'banana',
'mango',
'paprika',
'spinach',
'beetroot',
'corn',
'garlic',
'eggplant',
'cabbage',
'paprika',
'pineapple',
'spinach',
'tomato',
'pomegranate',
'lemon',
'paprika',
'corn',
'corn',
'cucumber',
'raddish',
'carrot',
'grapes',
'paprika',
'tomato',
'lemon',
'pineapple',
'corn',
'cucumber',
'chilli pepper',
'paprika',
'kiwi',
'mango',
'corn',
'pear',
'cauliflower',
'grapes',
'pear',
'turnip',
'watermelon',
'pineapple',
'corn',
'pineapple',
'potato',
```

```
'raddish',
'apple',
'ginger',
'apple',
'raddish',
'jalepeno',
'ginger',
'eggplant',
'onion',
'capsicum',
'garlic',
'grapes',
'soy beans',
'lettuce',
'bell pepper',
'mango',
'garlic',
'pear',
'chilli pepper',
'bell pepper',
'mango',
'kiwi',
'peas',
'ginger',
'pomegranate',
'corn',
'beetroot',
'onion',
'jalepeno',
'lettuce',
'apple',
'paprika',
'capsicum',
'kiwi',
'pomegranate',
'bell pepper',
'watermelon',
'cabbage',
'pineapple',
'cabbage',
'spinach',
'mango',
'soy beans',
'cabbage',
'cucumber',
'kiwi',
'eggplant',
'potato',
```

```
'mango',
'grapes',
'jalepeno',
'cucumber',
'turnip',
'potato',
'pear',
'potato',
'ginger',
'lemon',
'ginger',
'banana',
'lettuce',
'watermelon',
'orange',
'garlic',
'potato',
'watermelon',
'peas',
'onion',
'paprika',
'orange',
'grapes',]
```

```
img =
output('/content/drive/MyDrive/cuisinie-data/test/carrot/Image_1.jpg')
img
```

```
> 1/1 [=====] - 1s 941ms/step
carrot
```

```
model.save('cuisinie-final.h5')
```

```
import tensorflow as tf

# Load the saved model
model = tf.keras.models.load_model("/content/cuisinie-final.h5")
```

```

# Convert the model to TensorFlow Lite format
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

# Save the TensorFlow Lite model to a file
with open("model.tflite", "wb") as f:
    f.write(tflite_model)

from google.colab import drive
drive.mount('/content/drive')

```

5.2.1 cuisinie.ipynb

5.3 Code Snippets of Flutter Application

Below are few code snippets from Cusinie:

```

import 'package:cuisinie/HomePage.dart';
import 'package:cuisinie/LoginPage.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'LoginPage.dart';

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: FirebaseOptions(
      apiKey: "api key",
      appId: "1:808740408901:android:aadc3fce8bc84b367b9d2c",
      messagingSenderId: "808740408901",
      projectId: "cuisine-aff53",
    ),
  );
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Cuisinie',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      debugShowCheckedModeBanner: false,
      home: HomePage(),
    );
  }
}

```

5.3.1 main.dart

```

import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'dart:convert';
import 'package:url_launcher/url_launcher.dart';
import 'RecipeDetailsPage.dart';
import 'CalendarPage.dart';
import 'ProfilePage.dart';
import 'SmartSearchPage.dart';
import 'NutritionAnalysisPage.dart';

class HomePage extends StatefulWidget {
  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  TextEditingController _searchController = TextEditingController();
  List<dynamic> _recipeSearchResults = [];
  List<dynamic> _videoSearchResults = [];

  void _searchRecipes(String query, String apiKey) async {
    // Search for recipes using MealDB API
    final recipeResponse = await
    http.get(Uri.parse('https://www.themealdb.com/api/json/v1/1/search.php?s=$query'));
    if (recipeResponse.statusCode == 200) {
      setState(() {
        _recipeSearchResults = json.decode(recipeResponse.body)['meals'];
      });
    } else {
      throw Exception('Failed to load recipe search results');
    }
  }

  final refinedQuery = '$query recipe';

  final videoResponse = await
  http.get(Uri.parse('https://www.googleapis.com/youtube/v3/search?part=snippet&q=$refinedQuery&type=video&key=$apiKey'));
  if (videoResponse.statusCode == 200) {
    setState(() {
      _videoSearchResults = json.decode(videoResponse.body)['items'];
    });
  } else {
    throw Exception('Failed to load video search results');
  }
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: PreferredSize(
      preferredSize: Size.fromHeight(10),
      child: AppBar(
        backgroundColor: Colors.transparent,
        elevation: 0, // No shadow
        flexibleSpace: Container(
          decoration: BoxDecoration(
            color: Color(0xFFE4AD0C),
          ),
        ),
      ),
    ),
  );
}

```

```

),
),
body: Container(
  color: Color(0xFF25262A),
  child: Column(
    children: [
      Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Container(
            padding: EdgeInsets.only(top: 20, left: 16, right: 16),
            alignment: Alignment.centerLeft,
            child: Image.asset('assets/Cuisinie.png'),
          ),
          GestureDetector(
            onTap: () {
              Navigator.push(
                context,
                MaterialPageRoute(builder: (context) => ProfilePage()),
              );
            },
            child: Container(
              padding: EdgeInsets.only(top: 20, right: 16),
              child: Icon(Icons.account_circle_outlined, color: Color(0xFF959595), size: 40),
            ),
          ),
        ],
      ),
      SizedBox(height: 8),
      Padding(
        padding: const EdgeInsets.symmetric(horizontal: 16),
        child: Align(
          alignment: Alignment.centerLeft,
          child: Text(
            'What would you like to cook today?',
            style: TextStyle(
              fontSize: 18,
              color: Color(0xFFD9DEEE),
            ),
          ),
        ),
      ),
      SizedBox(height: 8),
      Container(
        padding: EdgeInsets.symmetric(horizontal: 10),
        child: Row(
          children: [
            Expanded(
              child: TextField(
                controller: _searchController,
                cursorColor: Color(0xFFE4AD0C),
                decoration: InputDecoration(
                  filled: true,
                  fillColor: Color(0xFF323337),
                  hintText: 'Search recipe using ingredients...',
                  hintStyle: TextStyle(color: Color(0xFF959595)),
                  contentPadding: EdgeInsets.only(left: 16, right: 16, top: 12, bottom: 12),
                  border: OutlineInputBorder(
                    borderRadius: BorderRadius.circular(20),

```

```

        borderSide: BorderSide.none,
    ),
    prefixIcon: Icon(Icons.search, color: Color(0xFF959595)),
),
style: TextStyle(color: Color(0xFF959595)),
onSubmitted: (value) {
    _searchRecipes(value, 'api key');
},
),
),
),
SizedBox(width: 10),
GestureDetector(
onTap: () {
    Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => CalendarPage()),
    );
},
),
child: Icon(Icons.calendar_month, color: Color(0xFF959595), size: 40,),
),
SizedBox(width: 10),
// Cooking events icon
GestureDetector(
onTap: () {
    Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => NutritionAnalysisPage()),
    );
},
),
child: Icon(Icons.analytics, color: Color(0xFF959595), size: 40,),
),
SizedBox(width: 10),
GestureDetector(
onTap: () {
    Navigator.push(
        context,
        MaterialPageRoute(builder: (context) => SmartSearchPage()),
    );
},
),
child: Icon(Icons.camera_alt, color: Color(0xFF959595), size: 40,),
),
],
),
),
),
),

Expanded(
child: SizedBox(
height: MediaQuery.of(context).size.height * 0.7,
child: DefaultTabController(
length: 2,
child: Column(
children: [
TabBar(
tabs: [
Tab(text: 'Recipes'),
Tab(text: 'Videos'),
],
indicatorColor: Color(0xFFE4AD0C),
labelColor: Color(0xFFE4AD0C),

```

```

),
Expanded(
  child: TabBarView(
    children: [
      _buildRecipeSearchResults(),
      _buildVideoSearchResults(),
    ],
  ),
),
],
),
),
),
),
),
),
),
),
),
),
),
),
),
);
}
}

Widget _buildRecipeSearchResults() {
  if (_recipeSearchResults.isEmpty) {
    return Center(
      child: Text('No recipes found'),
    );
  } else {
    return ListView.builder(
      itemCount: _recipeSearchResults.length,
      itemBuilder: (context, index) {
        Map<String, dynamic> recipe = _recipeSearchResults[index];

        return InkWell(
          onTap: () {
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => RecipeDetailsPage(recipe: recipe),
              ),
            );
          },
          child: Container(
            margin: EdgeInsets.all(8.0),
            padding: EdgeInsets.all(8.0),
            decoration: BoxDecoration(
              border: Border.all(color: Colors.grey),
              borderRadius: BorderRadius.circular(10.0),
            ),
            child: Row(
              mainAxisAlignment: MainAxisAlignment.start,
              children: [
                Container(
                  width: 150,
                  height: 150,
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(10.0),
                    image: DecorationImage(
                      image: NetworkImage(recipe['strMealThumb']),
                      fit: BoxFit.cover,
                    ),
                  ),
                ),
              ],
            ),
          ),
        );
      },
    );
  }
}

```

```

),
),
SizedBox(width: 12),
Expanded(
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
// Recipe title
Text(
recipe['strMeal'],
style: TextStyle(
fontSize: 18,
fontWeight: FontWeight.bold,
color: Color(0xFFE4AD0C),
),
),
SizedBox(height: 4),
Text(
recipe['strInstructions'],
maxLines: 2,
overflow: TextOverflow.ellipsis,
style: TextStyle(
color: Colors.white,
),
),
],
),
),
IconButton(
icon: Icon(Icons.calendar_today),
onPressed: () {
_addToCalendar(recipe);
},
),
],
),
),
);
},
);
}
}

void _addToCalendar(Map<String, dynamic> recipe) {
showDatePicker(
context: context,
initialDate: DateTime.now(),
firstDate: DateTime.now(),
lastDate: DateTime(DateTime.now().year + 1),
).then((selectedDate) {
if (selectedDate != null) {
_addRecipeToCalendar(recipe, selectedDate);
}
});
}

void _addRecipeToCalendar(Map<String, dynamic> recipe, DateTime selectedDate) {
// Implement adding recipe to calendar logic here
}

```

```

Widget _buildVideoSearchResults() {
  if (_videoSearchResults.isEmpty) {
    return Center(
      child: Text('No videos found'),
    );
  } else {
    return ListView.builder(
      itemCount: _videoSearchResults.length,
      itemBuilder: (context, index) {
        return InkWell(
          onTap: () {
            String videoId = _videoSearchResults[index]['id']['videoId'];
            String videoUrl = 'https://www.youtube.com/watch?v=$videoId';
            _launchURL(videoUrl);
          },
          child: Container(
            margin: EdgeInsets.all(8.0),
            padding: EdgeInsets.all(8.0),
            decoration: BoxDecoration(
              border: Border.all(color: Colors.grey),
              borderRadius: BorderRadius.circular(10.0),
            ),
            child: Row(
              mainAxisAlignment: MainAxisAlignment.start,
              children: [
                Container(
                  width: 100,
                  height: 100,
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(10.0),
                    image: DecorationImage(
                      image:
                        NetworkImage(_videoSearchResults[index]['snippet']['thumbnails']['medium']['url']),
                      fit: BoxFit.cover,
                    ),
                ),
                SizedBox(width: 12),
                Expanded(
                  child: Column(
                    mainAxisAlignment: MainAxisAlignment.start,
                    children: [
                      // Video title
                      Text(
                        _videoSearchResults[index]['snippet']['title'],
                        style: TextStyle(
                          fontSize: 18,
                          fontWeight: FontWeight.bold,
                          color: Color(0xFFE4AD0C),
                        ),
                      ),
                      SizedBox(height: 4),
                      Text(
                        _videoSearchResults[index]['snippet']['description'],
                        maxLines: 2,
                        overflow: TextOverflow.ellipsis,
                        style: TextStyle(
                          color: Colors.white,
                        ),
                      ),
                    ],
                  ),
                ),
              ],
            ),
          ),
        );
      }
    );
  }
}

```

```

        ),
        ],
        ],
        ],
        ],
        ],
        );
        },
        );
        }
    }
}

void _launchURL(String url) async {
    if (await canLaunch(url)) {
        await launch(url, forceSafariVC: false);
    } else {
        throw 'Could not launch $url';
    }
}

```

5.3.2 HomePage.dart

```

import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

class NutritionAnalysisPage extends StatefulWidget {
    @override
    _NutritionAnalysisPageState createState() => _NutritionAnalysisPageState();
}

class _NutritionAnalysisPageState extends State<NutritionAnalysisPage> {
    TextEditingController _foodController = TextEditingController();
    Map<String, dynamic>? _nutritionInfo;

    Future<void> fetchNutritionInfo(String food) async {
        final appId = 'api';
        final appKey = 'app id';
        final response = await http.get(Uri.parse(
            'https://api.edamam.com/api/nutrition-data?app_id=$appId&app_key=$appKey&ingr=$food'
        ));

        if (response.statusCode == 200) {
            setState(() {
                _nutritionInfo = jsonDecode(response.body);
            });
        } else {
            print('Failed to fetch nutrition information: ${response.statusCode}');
        }
    }
}

```

```

        }

Widget buildNutritionInfo() {
List<Widget> nutrientWidgets = [];

_nutritionInfo!['totalNutrients'].forEach((key, value) {
String label = value['label'];
double quantity = value['quantity'];
String unit = value['unit'];

nutrientWidgets.add(
 ListTile(
 title: Text(label),
 subtitle: Text('${quantity.toStringAsFixed(2)} $unit'),
 ),
);
});

return Column(
 children: nutrientWidgets,
);
}

@Override
Widget build(BuildContext context) {
return Scaffold(
 appBar: AppBar(
 title: Text('Nutrition Analysis'),
 backgroundColor: Color(0xFFE4AD0C),
 ),
 backgroundColor: Color(0xFF25262A),
 body: Padding(
 padding: EdgeInsets.all(16.0),
 child: Column(
 crossAxisAlignment: CrossAxisAlignment.stretch,
 children: [
 TextField(
 controller: _foodController,
 decoration: InputDecoration(labelText: 'Enter Food intake', labelStyle: TextStyle(color: Color(0xFFE4AD0C)), hintStyle: TextStyle(color: Color(0xFFE4AD0C)),
 focusedBorder: UnderlineInputBorder(
 borderSide: BorderSide(color: Color(0xFFE4AD0C)), // Change the color of the line when focused
 ),
 style: TextStyle(color: Colors.white),
 cursorColor: Color(0xFFE4AD0C),
 ),
 SizedBox(height: 20),
 ElevatedButton(
 onPressed: () {
 fetchNutritionInfo(_foodController.text);
 },
 style: ElevatedButton.styleFrom(

```

```
primary: Color(0xFFE4AD0C),
),
child: Text('Get Nutrition Info',
style: TextStyle(color: Color(0xFF25262A)),
),
),
SizedBox(height: 20),
if (_nutritionInfo != null)
Expanded(
child: SingleChildScrollView(
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Card(
color: Color(0xFF959595),
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(10.0),
side: BorderSide(color: Color(0xFFE4AD0C)),
),
child: Padding(
padding: EdgeInsets.all(8.0),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text(
'Food: ${_nutritionInfo!['ingredients'][0]['parsed'][0]['food']}',
style: TextStyle(fontWeight: FontWeight.bold, fontSize: 20),
),
Text('Calories: ${_nutritionInfo!['calories']}',
style: TextStyle(fontWeight: FontWeight.bold, fontSize: 20)),
],
),
),
),
),
),
Card(
color: Color(0xFF959595),
shape: RoundedRectangleBorder(
borderRadius: BorderRadius.circular(10.0), // Set border radius
side: BorderSide(color: Color(0xFFE4AD0C)), // Set border side
),
child: Padding(
padding: EdgeInsets.all(8.0),
child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
children: [
Text('Total Nutrients', style: TextStyle(fontWeight: FontWeight.bold, fontSize: 30)),
SizedBox(height: 10),
buildNutritionInfo(),
],
),
),
)
```

```
        ),  
        ),  
        ],  
        ),  
        ),  
        )  
    ],  
    ),  
    );  
}  
}
```

5.3.3 NutritionAnalysisPage.dart

CHAPTER: 6

CONCLUSIONS &

FUTURE SCOPE OF

WORK

Chapter 6

CONCLUSIONS AND FUTURE SCOPE OF WORK

In conclusion, our application represents a transformative solution for individuals seeking a seamless and personalized culinary experience. Developed with Flutter, this app simplifies recipe discovery, meal planning, and personalized recommendations, enhancing users' engagement with cooking and meal preparation.

Looking forward, we have identified several exciting enhancements and future scope opportunities to further elevate this innovative platform:

- Pantry Management: Introduce pantry management features to help users organize ingredients and plan meals based on available items in their kitchen.
- Integration with Grocery Shopping Sites: Enable integration with grocery shopping platforms to facilitate seamless ingredient procurement based on selected recipes.
- Community Interaction: Enhance community interaction by allowing users to share recipes, cooking tips, and meal ideas with each other, fostering a vibrant culinary community.
- Voice Search and Integration: Implement voice search capabilities to enable hands-free recipe discovery and navigation within the application, making cooking more convenient and accessible.

By incorporating these enhancements and future scope elements, we aim to further enrich the user experience, making our cuisinie application a comprehensive and indispensable tool for cooking enthusiasts and home chefs alike. The journey ahead promises continuous innovation and refinement, ensuring that our platform remains at the forefront of culinary technology.

REFERENCES

- [01] K. J. Hammond, “Chef: A model of case-based planning.” in AAAI, vol. 86, 1986, pp. 267–271.
- [02] L. R. Varshney, F. Pinel, K. R. Varshney, D. Bhattacharjya, A. Schorgendorfer, and Y.-M. Chee, “A big data approach to computational creativity: The curious case of chef watson,” IBM Journal of Research and Development, vol. 63, no. 1, pp. 7–1, 2019.
- [03] Yanai, K., Maruyama, T., & Kawano, Y. (2014). A Cooking Recipe Recommendation System with Visual Recognition of Food Ingredients. *International Journal of Interactive Mobile Technologies (iJIM)*, 8(2), 28.
- [04] C. Trattner and D. Elsweiler, “Food recommender systems: important contributions, challenges and future research directions,” arXiv preprint arXiv:1711.02760, 2017.
- [05] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, “A survey on food computing,” ACM Computing Surveys (CSUR), vol. 52, no. 5, pp. 136, 2019.
- [06] A. Hashimoto, N. Mori, T. Funatomi, Y. Yamakata, K. Kakusho, and M. Michihiko. Smart kitchen: A user centric cooking support system. In Proc. of International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pages 848-854, 2008.
- [07] M. Ueda, M. Takahata, and S. Nakajima. User's food preference extraction for cooking recipe recommendation. In Proc. of the 2nd Workshop on Semantic Personalized Information Management: Retrieval and Recommendation, 2011.
- [08]] Desai, Ms. Soundarya, Ms. Pooja Patil, Mr. Pratik Shinde, Mr. Azhar Sayyed, and Rohini Bhosale. "Recipe recommendation based on ingredients using machine learning. 8 (3)." (2019).

PLAGIARISM REPORT

PAPER NAME

Cusinie - Machine Learning infused Culinary Companion Report.pdf

WORD COUNT

7507 Words

CHARACTER COUNT

49950 Characters

PAGE COUNT

55 Pages

FILE SIZE

6.7MB

SUBMISSION DATE

Apr 30, 2024 12:57 PM GMT+5:30

REPORT DATE

Apr 30, 2024 12:58 PM GMT+5:30

● 17% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 15% Internet database
- Crossref database
- 17% Submitted Works database
- 6% Publications database
- Crossref Posted Content database

● Excluded from Similarity Report

- Bibliographic material
- Cited material
- Quoted material
- Small Matches (Less than 10 words)

APPENDICES

- Appendix I: Cover page
- Appendix II: Declaration by the candidate
- Appendix III: Certificate by the Supervisor
- Appendix IV: Certificate by the Examiner
- Appendix V: Acknowledgement from student
- Appendix VI: Table of contents
- Appendix VII: References
- Appendix VIII: Plagiarism Report