CROP PREDICTION AND AGRICULTURAL TOOL RECOMMENDATION SYSTEM USING ML

A MINI PROJECT II REPORT

Submitted by

SUDHARSAN K (71812101239)
VIJAYANANDH R (71812101255)
SANTHOSH KUMAR M (71812101222)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING
in

COMPUTER SCIENCE AND ENGINEERING

SRI RAMAKRISHNA ENGINEERING COLLEGE
[Educational Service: SNR Sons Charitable Trust]
[Autonomous Institution, Reaccredited by NAAC with 'A+' Grade]
[Approved by AICTE and Permanently Affiliated to Anna University, Chennai]
[ISO 9001:2015 Certified and All Eligible Programmes Accredited by NBA]
Vattamalaipalayam, N.G.G.O. Colony Post,
COIMBATORE – 641 022

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2024

SRI RAMAKRISHNA ENGINEERING COLLEGE

BONAFIDE CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MINI PROJECT II – MAY 2024

This is to certify that the project entitled

CROP PREDICTION AND AGRICULTURAL TOOL RECOMMENDATION SYSTEM USING ML

is the bonafide record of Mini Project II done by

SUDHARSAN K (71812101239)
VIJAYANANDH R (71812101255)
SANTHOSH KUMAR M (71812101222)

of B.E. Computer Science and Engineering during the year 2023-2024.

who carried out the Mini Project II under my supervision, certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. R. Madhumathi, M.E, Ph.D.,	Dr. A. Grace Selvarani, M.E, Ph.D.,
PROJECT GUIDE	**HEAD OF THE DEPARTMENT**
Associate Professor,	Professor,
Computer Science and Engineering,	Computer Science and Engineering,
Sri Ramakrishna Engineering College,	Sri Ramakrishna Engineering College,
Coimbatore-641022.	Coimbatore-641022.

Submitted for the Project Viva-Voce Examination held on ______________

Internal Examiner **External Examiner**

DECLARATION

We affirm that the Mini Project II titled **“Crop Prediction and Agricultural Tool Recommendation System using ML”** being submitted in partial fulfillment for the award of Bachelor of Engineering is the original work carried out by us. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

_____________--___________--_____________

(Signature of the Candidates)

SUDHARSAN K	**(71812101239)**
VIJAYANANDH R	**(71812101255)**
SANTHOSH KUMAR M	**(71812101222)**

I certify that the declaration made above by the candidates is true.

____

(Signature of the guide)

Dr. R. MADHUMATHI
ASSOCIATE PROFESSOR,
DEPARTMENT OF CSE.

ACKNOWLEDGEMENT

We express our gratitude to **Sri. D. LAKSHMINARAYANASWAMY,** Managing Trustee, **Sri. R. SUNDAR,** Joint Managing Trustee, SNR Sons Charitable Trust, Coimbatore for providing excellent facilities to carry out our project.

We express our deepest gratitude to **Dr. N. R. ALAMELU,** Principal, for her valuable guidance and blessings.

We thank **Dr. A. GRACE SELVARANI,** Professor and Head, Department of Computer Science and Engineering who modeled us both technically and morally for achieving great success in life.

We sincerely thank our Project Coordinator**, Mr. R. S. VISHNU DURAI,** Assistant Professor (Sr. Grade), Department of Computer Science and Engineering for his great inspiration.

Words are inadequate to offer thanks to our respected guide. We wish to express our sincere thanks to **Dr. R. MADHUMATHI,** Associate Professor, Department of Computer Science and Engineering, who gives constant encouragement and support throughout this project work and who makes this project a successful one.

We also thank all the staff members and technicians of our department for their help in making this project a successful one

TABLE OF CONTENTS

ABSTRACT

Precision agriculture, marked by meticulous management of agricultural inputs to maximize yields while minimizing environmental impact, represents a promising frontier in contemporary farming. This project introduces an agricultural tool recommendation system aimed at supporting farmers in making informed decisions about crop selection and necessary agricultural equipment. By analyzing key factors like air temperature, soil pH, rainfall, and air humidity, this system harnesses machine learning algorithms, specifically the Decision Tree Classifier, achieving an impressive 93.8% accuracy in predicting suitable crop varieties based on environmental conditions. Furthermore, it goes beyond mere crop selection, integrating an advanced recommendation mechanism for agricultural tools tailored to specific farming requirements. Through thorough analysis of factors such as crop type, soil characteristics, and climatic conditions, the system offers personalized recommendations to optimize farming practices for increased efficiency and productivity. It also helps you find the right tools easily from trusted vendors, making it simpler to buy the suggested farming equipment. In the end, our system aims to improve farming productivity and efficiency by optimizing resource use and making farming operations smoother. This contributes to promoting sustainable practices in precision agriculture.

LIST OF FIGURES

LIST OF ABBREVIATION

ABBREVIATION	EXPANSION
CNN	CONVOLUTIONAL NEURAL NETWROK
RFC	RANDOM FOREST CLASSIFIER
DT	DECISION TREE
API	APPLICATION PROGRAMMING INTERFACE
GPU	GRAPHICAL PROCESSING UNIT
TPU	TENSOR PROCESSING UNIT
URL	UNIFORM RESOURCE LOCATOR
UI	USER INTERFACE

LIST OF TABLES

1. INTRODUCTION

This project “Crop prediction and tool recommendation system using ML”, analyses soil moisture, pH levels, and rainfall data, it accurately forecasts the ideal crop for cultivation. Additionally, it seamlessly integrates with a tool recommendation system, suggesting necessary equipment and trusted vendors for optimal agricultural productivity.

1.1 PRECISION FARMING

In recent years, agriculture has undergone a significant transformation with the advent of precision farming techniques. Precision agriculture, enabled by advancements in technology and data analytics, aims to optimize agricultural practices by precisely managing inputs such as water, fertilizers, and pesticides while maximizing yields and minimizing environmental impact which is depicted in figure 1.1.

Figure 1.1 – Precision Farming

1.2 PROJECT OVERVIEW

This project, titled "Crop prediction and tool recommendation system using ML," addresses the critical need for personalized guidance in agricultural decision-making processes. By leveraging key environmental parameters such as air temperature, air humidity, soil pH level, rainfall value, and soil moisture content, this system offers predictive insights into crop suitability and recommends the necessary tools for optimal farming practices. In doing so, it not only enhances productivity but also contributes to sustainability efforts by promoting efficient resource utilization.

1.3 KEY COMPONENTS OF THE SYSTEM

The foundation of this system lies in the integration of data-driven predictive modeling techniques with a sophisticated recommendation engine. Through the utilization of machine learning algorithms, we harness the power of historical data to predict the most suitable crop varieties for a given set of environmental conditions. By analyzing the interplay between climate factors, soil characteristics, and crop preferences, this system provides farmers with actionable insights into crop selection, tailored to their specific geographical location and farming practices.

1.4 TOOL RECOMMENDATION

Moreover, this system extends beyond crop prediction to address the critical aspect of tool selection in precision farming. Recognizing that the choice of agricultural tools significantly influences farm efficiency and productivity, we have developed a tool prediction module that suggests the necessary implements based on predicted crop types and farming requirements. By considering factors such as crop-specific cultivation techniques, pest management strategies, and soil preparation needs, our tool recommendation system assists farmers in making informed decisions regarding tool procurement.

1.5 ENHANCED ABILITY AND SUPPLY CHAIN INTEGRATION

A distinguishing feature of this project is the inclusion of dealer details for the recommended agricultural tools within the respective region. Recognizing the importance of accessibility and availability of agricultural inputs, we aim to bridge the gap between recommendation and implementation by providing farmers with information on local dealers who supply the recommended tools. This not only streamlines the procurement process but also fosters stronger connections within the agricultural supply chain, ultimately benefiting both farmers and tool dealers alike.

2. LITERATURE REVIEW

2.1 Crop Prediction Based on Characteristics of the Agricultural Environment Using Various Feature Selection Techniques and Classifiers [1]

This paper explored the vital task of crop prediction in agriculture, leveraging machine learning techniques to analyze environmental factors like rainfall and temperature. It emphasizes the importance of efficient feature selection methods to preprocess data, ensuring only relevant features contribute to the model. Through experimentation, the study demonstrates that ensemble techniques outperform existing classification methods, highlighting the significance of accurate prediction in agricultural practices.

2.2 Effective Crop Prediction Using Deep Learning [2]

This paper focuses on effective crop prediction in India, acknowledging the country's significant agricultural role in its economy and culture. Leveraging deep learning, the research presents a model utilizing a Feed forward neural network with Rectified Linear Activation Unit and backward/forward propagation techniques. By incorporating crucial environmental factors and chemical elements like nitrogen, phosphorus, and potassium, along with rainfall and temperature data, the model demonstrates satisfactory performance in predicting crop yield, showcasing the potential of deep learning in agricultural applications.

2.3 Agro App: An application for healthy living [3]

This paper introduces "Agro -App," a mobile application designed for both farmers and individuals interested in growing vegetables. The app aims to promote healthy living by providing comprehensive information on crops, pesticides, and financial aspects related to agriculture. It assists farmers in selecting suitable crops for their region and season, bridging the literacy gap by leveraging the widespread use of mobile devices. Agro-App aligns with governmental efforts to integrate technology into agriculture for enhanced productivity and sustainability.

2.4 Android App to Connect Farmers to Retailers and Food Processing Industry [4]

This paper introduces an Android app facilitating direct connections between farmers, retailers, and food processing industries. By leveraging mobile internet, it enables farmers to efficiently sell their produce and access market information. The app, designed with a user-friendly interface and native language support, serves as a platform for farmers to both buy and sell agricultural goods. Utilizing data.gov.in for market prices, the system ensures fair transactions while offering consumers a wide range of products with location-based filtering options, promoting transparency and efficiency in agricultural trade.

2.5 Crop Prediction Using Machine Learning [5]

This paper addresses global agricultural challenges by proposing a methodology that utilizes decision tree and neuro-evolutionary algorithms to forecast crop prices. The aim is to improve farmers' livelihoods and enhance yield sustainability. It conducts a comprehensive exploration of machine learning techniques in agriculture and suggests future enhancements, such as an automated price recommendation system integrating genetic algorithms. This research endeavors to contribute significantly to the advancement of precision farming practices and agricultural sustainability on a global scale.

2.6 Crop Recommender System using Machine Learning Approach [6]

This paper introduces a mobile application-based yield prediction system tailored for Indian farmers. By utilizing GPS technology and user inputs regarding land area and soil type, the system employs machine learning algorithms, notably Random Forest, to achieve an impressive 95% accuracy in crop yield prediction. Furthermore, it provides recommendations for optimal fertilizer timing, aiming to enhance agricultural productivity. Through the integration of advanced technology and data-driven approaches, this system seeks to empower farmers with valuable insights for more efficient and successful farming practices.

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system for crop prediction employs machine learning algorithms to forecast suitable crops based on environmental factors such as soil composition, climate conditions, and historical data. Through data analysis and predictive modeling, the system evaluates various parameters to recommend crops that are likely to thrive in a particular region during specific seasons.

The existing project leverages advanced techniques like decision trees, support vector machines, or neural networks to analyze vast datasets for precise crop predictions. Currently, it focuses solely on suggesting crops, lacking tool recommendations. The project goal is to simplify and enhance the system by combining crop and tool suggestions into one user-friendly mobile app, offering farmers complete guidance for successful farming.

DRAWBACKS OF EXISTING SYSTEM

- **Limited Guidance**: Current system lacks crucial tool and technique recommendations, leaving farmers without comprehensive guidance.

- **Incomplete Decision Support:** Without tool and technique recommendations, the system may not fully support farmers, leading to inefficiencies and potential errors.

- **Missed Opportunities for Efficiency:** By not integrating tool recommendations, the existing system misses the opportunity to streamline farming operations and improve efficiency. Farmers may struggle to identify the most suitable tools for their specific crops and conditions, resulting in suboptimal resource utilization and productivity.

- **Increased Complexity for Users:** Separating crop prediction from tool recommendations may add complexity for users, requiring them to navigate multiple systems or sources of information. This fragmentation could hinder user adoption and usability, particularly for farmers with limited access to technology or technical expertise.

3.2 PROPOSED SYSTEM

This project, "Crop prediction and tool recommendation system using ML", aims to revolutionize farming practices by integrating crop and tool recommendations into a unified mobile application. This comprehensive tool harnesses the power of predictive analytics to offer farmers tailored suggestions for optimal crop selection and suitable agricultural tools and techniques.

By analyzing factors such as soil quality, ph value, air temperature and air humidity, the system provides accurate forecasts of crops that are best suited for cultivation. Additionally, it evaluates various agricultural tools and techniques, recommending those most aligned with the farmer's requirements and farming practices.

Through this innovative approach, our system empowers farmers to make informed decisions, enhancing precision and efficiency in agricultural operations. Ultimately, our goal is to support sustainable farming practices while maximizing productivity and profitability for farmers worldwide. The proposed methodology for the “Crop prediction and tool recommendation system using ML” involves several key steps to develop an effective recommendation system tailored to the needs of farmers and specific crop.

MERITS OF PROPOSED SYSTEM:

- **Integrated Recommendations:** Offers both crop and agricultural tool recommendations in a single platform, providing farmers with comprehensive guidance for optimized farming practices.
- **Improved Decision Making:** Empowers farmers to make informed decisions by providing accurate forecasts of suitable crops and recommending the most aligned agricultural tools and techniques, enhancing precision and efficiency in farming operations.
- **Increased Productivity:** Facilitates sustainable farming practices by recommending crops and tools that maximize productivity while minimizing environmental impact, ultimately leading to increased yields and profitability for farmers.
- **User-Friendly Interface:** Offers a user-friendly mobile application interface, making it easy for farmers to access and implement recommendations, even with limited technical expertise.

.

4. SYSTEM SPECIFICATION

4.1 SOFTWARE REQUIREMENTS

- FLUTTER – DART
- VS CODE
- ANDROID EMULATOR
- FAST API
- GOOGLE COLABORATORY
- URL LAUNCHER FLUTTER PACKAGE

4.2 HARDWARE REQUIREMENTS

- PERSONAL LAPTOP WITH
  - ➢PROCESSOR – INTEL I5
  - ➢RAM – 16GB
  - ➢WINDOWS – EDITION 11

5. SOFTWARE DESCRIPTION

5.1 FLUTTER - DART

For developing this smart attendance system, we have chosen the ‘Flutter’ platform. Flutter is an open-source UI software development kit created by Google. It allows developers to build high-performance, cross-platform mobile apps for Android and iOS, using a single codebase. Flutter offers a rich set of pre-built widgets and tools, making it easy and efficient to build beautiful and functional mobile apps**.**

5.2 VS CODE

The Text Editor that we had used to develop this crop and tool recommendation system is Visual Studio Code (VS Code). It is free, open-source, cross-platform code editor developed by Microsoft. It supports various programming languages and provides an array of features such as intellisense, debugging tools, Git integration, extensions, and an integrated terminal. With a customizable user interface and excellent performance, VS Code is a popular choice among developers for coding, debugging, and managing code efficiently.

5.3 ANDROID EMULATOR

The android emulator used for this project development is Google Pixel 5 with Android 11 OS and it is depicted in figure 5.1.

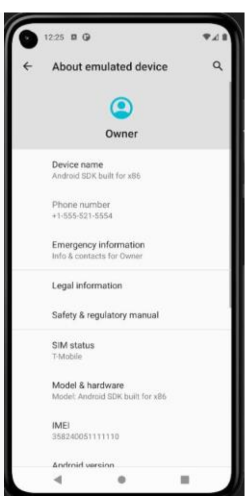

Figure 5.1 – Android Emulator

5.4 FAST API

FastAPI is a modern, high-performance web framework for building APIs with Python. It offers rapid development capabilities by leveraging type annotations for automatic data validation and documentation generation. With asynchronous support, it handles high loads efficiently, making it ideal for real-time applications. FastAPI boasts a simple, intuitive syntax that enables developers to quickly create robust APIs. Its built-in support for OpenAPI and JSON Schema ensures interoperability and easy integration with other tools. Overall, FastAPI accelerates API development while maintaining scalability and reliability.

5.5 GOOGLE COLABORATORY

Google Colab, short for Colaboratory, is a cloud-based platform provided by Google that enables users to write and execute Python code in a web browser. It offers free access to GPU and TPU resources, making it ideal for machine learning and data analysis tasks. With its collaborative features, users can share and edit notebooks in real-time, facilitating seamless collaboration among teams. Colab integrates with popular libraries like TensorFlow and PyTorch, simplifying the process of building and training machine learning models. Overall, Google Colab provides a convenient and powerful environment for Python programming and experimentation.

5.6 URL LAUNCHER FLUTTER PACKAGE

The URL Launcher Flutter package is a convenient tool for Flutter developers to integrate URL launching functionality into their mobile applications. With this package, developers can easily open web URLs, email addresses, phone numbers, and other external links directly from their Flutter apps. This package streamlines the process of directing users to external resources or apps installed on their devices, enhancing the overall user experience and enabling seamless navigation between the app and external content.

6. PROJECT DESCRIPTION

6.1 PROBLEM DEFINITION

The problem addressed by this project, revolves around the optimization of agricultural practices through the integration of predictive analytics. Specifically, the project aims to predict suitable crops for cultivation based on various environmental factors, including soil moisture, soil pH, air temperature, air humidity, and historical weather data. Additionally, the system endeavors to recommend necessary agricultural tools and techniques tailored to the predicted crop, thereby enhancing precision and efficiency in farming practices.

6.2 INTRODUCTION TO PROPOSED SYSTEM

This project aims to merge traditional farming methods with modern technology to assist farmers in making informed decisions. By utilizing predictive analytics and machine learning, it predicts suitable crops based on environmental parameters like soil moisture, pH, temperature, and humidity. Additionally, it recommends agricultural tools tailored to the predicted crop's needs, including irrigation systems, fertilizers, and pesticides. The system emphasizes user-friendliness, providing detailed information on recommended tools and facilitating access to suppliers. Ultimately, it seeks to empower farmers with the tools and insights necessary for precision, efficiency, and sustainability in agriculture, thereby revolutionizing the farming landscape.

In response to these challenges, this project, endeavors to bridge the gap between traditional farming practices and modern technological solutions. This innovative system harnesses the power of predictive analytics and machine learning to provide farmers with intelligent recommendations for both crop selection and necessary agricultural tools.

Furthermore, this system goes beyond crop prediction by seamlessly integrating a tool recommendation component. Once the system predicts the most suitable crop for cultivation, it automatically generates recommendations for the necessary agricultural tools required to support the successful growth and management of the predicted crop. The proposed system flow diagram has been depicted in figure 6.1.

PROPOSED SYSTEM

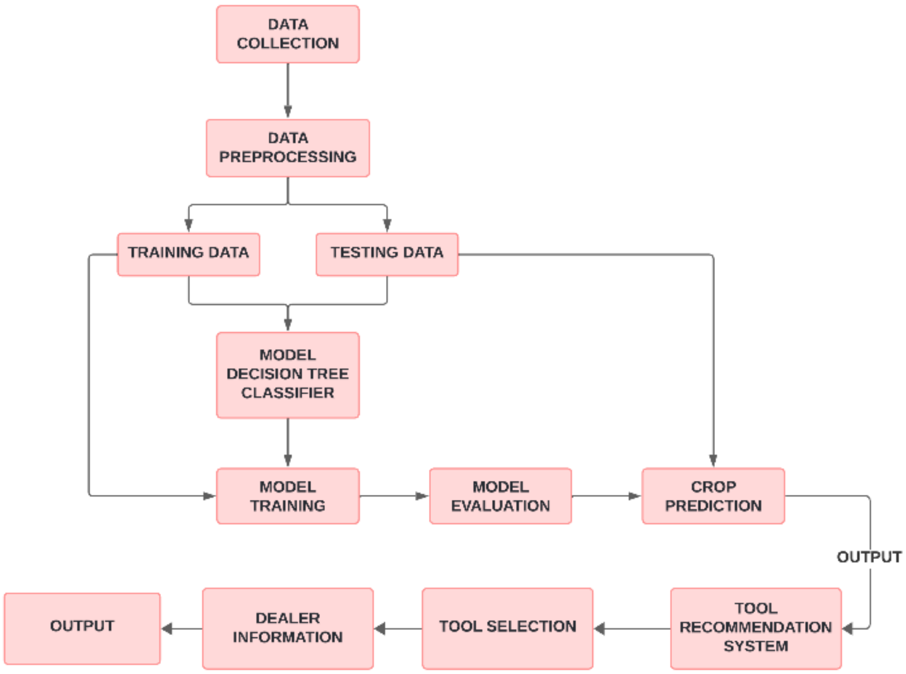

Figure 6.1. Proposed Flow Diagram

Algorithm - Tool Recommendation System

1. Data Collection
2. Data Preprocessing
3. Split( ) the data to evaluate its performance
4. Predict the crop
   4.1 CNN
   4.2 RFC
   4.3 DT
5. Tool Recommendation
6. Evaluate the preformance of the model
7. Display the output in Mobile App

DATA COLLECTION

- **Crop Prediction:** Crop prediction relies on meticulous data collection, encompassing crucial factors such as air temperature, humidity, soil moisture, pH value, and rainfall. These parameters offer insights into the environmental conditions conducive to crop growth and development. Gathering accurate and up-to-date information ensures the effectiveness of predictive models, enabling farmers to make informed decisions regarding crop selection, planting schedules, and resource management. Through systematic data collection, predictive accuracy can be optimized, enhancing agricultural productivity and sustainability. In order to train the model more accurately, a crop dataset from Kaggle has been used.

- **Tool Recommendation:** Gather detailed information about each merchant, including their name, location, contact details, website (if applicable), and types of products they offer. Collect information on the range of agricultural tools and equipment available, such as weeders, augers, and other available agriculture tools. The tool vendor details in and around Coimbatore region has been shown in table 1.

Table 1 – Agricultural tool vendor details

S.No	Dealer Name	Tools Available
1	Ganapathy Agro Industries	• Roller Weeder • Spiral Weeder • Auger
2	Banumathi Industries	• Hand Weeder • Kalai Vetti • Auger
3	Guru Murugan & Co	• Cycle Weeder • Kalapai
4	Tamilnadu Agro Agencies	• Hand Weeder • Kalapai
5	Green Kraft	• Roller Weeder • Cycle Weeder • Auger
6	Bhuvana Enterprises	• Groundnut Weeder • Auger • Spiral Weeder

7	Sri Balaji Industries	• Groundnut Breaker Machine • Corn Breaker Machine
8	Kisan Kraft Ltd	• Roller Weeder • Spiral Weeder • Hand Weeder
9	Star Bright Agency	• Spiral Weeder • Kalapai
10	Kave Enterprise	• Annchu Kalapai • Moonu Kalapai • Auger
11	Swathi Industries	• Fruit Plucker • Hand Weeder • Auger
12	Sakthi Agro Agencies	• Cycle Weeder • Kalapai • Spiral Weeder
13	Annai Engineering Enterprises	• Kalapai • Weeder • Auger
14	Sharp Garuda Farm Equipment	• Groundnut Weeder • Cycle Weeder • Spiral Weeder
15	Flow Tech	• Kalai Vetti • Fruit Plucker

6.3 MODULE DESCRIPTION

6.3.1 CROP RECOMMENDATION SYSTEM

This model utilizes decision tree regression to predict suitable crops based on environmental parameters such as air humidity, temperature, soil humidity, pH, and rainfall. It preprocesses data, trains the model, and predicts the crop. Additionally, it randomly recommends three agricultural tools. The trained model is saved for future use. This system aids farmers in making informed decisions regarding crop selection and necessary tools for precision farming.

6.3.2 TOOL RECOMMENDATION SYSTEM

The agricultural tool recommendation system utilizes predictive analytics to anticipate the tools required for specific crops, offering corresponding dealer information. By leveraging this data-driven approach, it ensures precision and efficiency in farming operations, ultimately enhancing productivity and optimizing resource utilization.

6.3.3 DEVELOPING MOBILE APPLICATION

The Flutter-based mobile app simplifies farming decisions by suggesting crops and tools. Using predictive analytics, it recommends crops based on environmental factors and prescribes necessary tools. Farmers can easily connect with dealers through the app for tool procurement, streamlining the farming process. With user-friendly features, it enhances decision-making and convenience for farmers, promoting efficient and productive farming practices.

6.3.4 INTEGRATING APP AND MOBILE APPLICATION

Integrating the mobile app with the recommendation system via FastAPI enhances farming efficiency. This module seamlessly connects the app's user interface with the backend recommendation system. Utilizing FastAPI, it ensures swift communication between the mobile app and the recommendation system, enabling real-time crop and tool suggestions for farmers. This integration streamlines decision-making processes, fostering more productive and sustainable agricultural practices

7. SYSTEM IMPLEMENTATION

7.1 WORKING OF TOOL RECOMMENDATION SYSTEM

Technology has brought about a new age of precise farming, where decisions are based on data, improving how we farm. This project, aims to streamline farming operations by integrating two essential components: a crop recommendation system and a tool recommendation system, within a unified mobile application. In the first phase, the crop recommendation system utilizes machine learning algorithms to analyze environmental parameters such as soil moisture, pH, air temperature, humidity, and rainfall to predict suitable crops for cultivation. This predictive model provides farmers with tailored recommendations, guiding them in selecting crops that are best suited to their specific geographical location and prevailing environmental conditions. The output of the crop recommendation system serves as input to the second phase of the project.

In the second phase, the tool recommendation system leverages the predicted crop information from the first phase to recommend necessary agricultural tools and provide dealer details. Based on the predicted crop, the system filters through a database of agricultural tools, selecting those essential for cultivating and managing the predicted crop. Additionally, the system retrieves dealer information associated with the recommended tools, including contact details, locations, and available inventory. By integrating these functionalities into a mobile application, farmers gain access to a comprehensive toolset tailored to their specific crop requirements, along with the means to procure these tools conveniently from trusted dealers.

This project aims to address the challenges faced by farmers in navigating the complexities of modern agriculture by providing a comprehensive solution that integrates crop and tool recommendations within a single mobile application. By leveraging machine learning algorithms and predictive analytics, our system empowers farmers with actionable insights, enabling them to make informed decisions at every stage of the farming process.

The first phase of this project focuses on developing a robust crop recommendation system capable of predicting suitable crops for cultivation based on environmental parameters. This system utilizes historical data on soil moisture, pH levels, air temperature, humidity, and rainfall to train machine learning models, which in turn provide farmers with tailored recommendations for crop selection. By analyzing these environmental factors, the system identifies crops that are most likely to thrive in a given geographical location and prevailing climatic conditions, thereby assisting farmers in optimizing their planting decisions for maximum yield and profitability.

In the second phase of the project, the output of the crop recommendation system serves as input to the tool recommendation system. This phase involves the development of a data containing information on various agricultural tools, including types, specifications, and recommended usage for specific crops. Based on the predicted crop from the first phase, the tool recommendation system filters through this data to select tools that are essential for cultivating and managing the predicted crop effectively. Additionally, the system retrieves dealer details associated with the recommended tools, providing farmers with access to trusted suppliers for procuring these tools conveniently. The underlying system architecture has been depicted in figure 7.1.

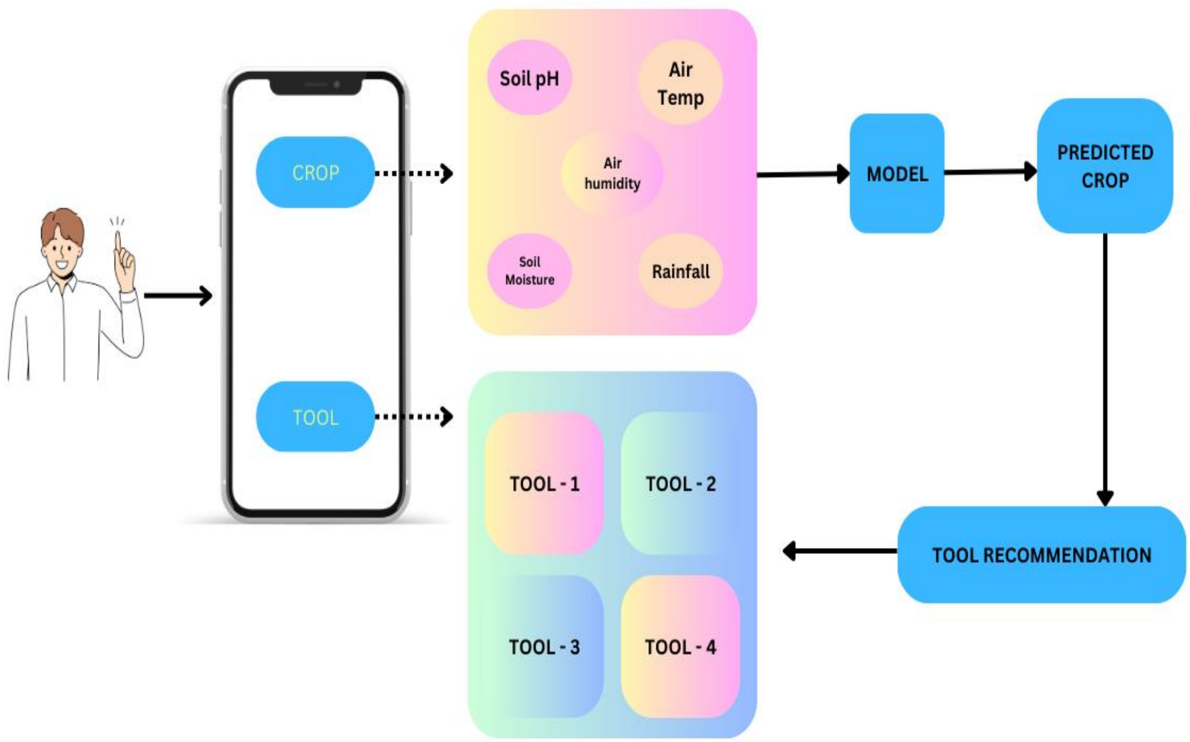

***Figure 7.1** – System Architecture*

8. RESULTS AND DISCUSSION

The CNN, Random Forest, and Decision Tree algorithms are utilized for crop prediction, with respective accuracies of 75%, 80%, and 93.6% as depicted in Figures 8.1, 8.2, and 8.3. The accuracy comparison of these algorithms is presented in Figure 8.4. Given its highest accuracy, the Decision Tree algorithm is selected for the crop prediction model.

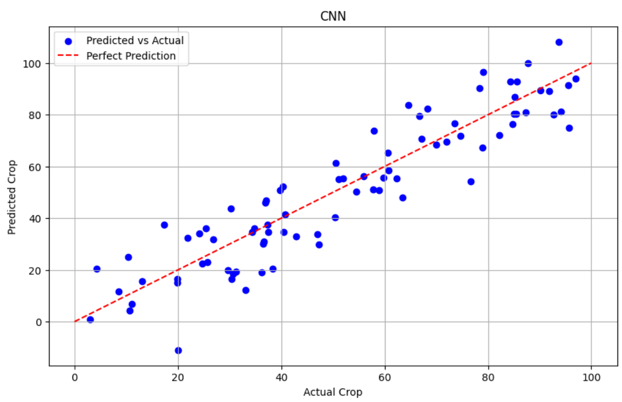

Figure 8.1 – CNN Graph

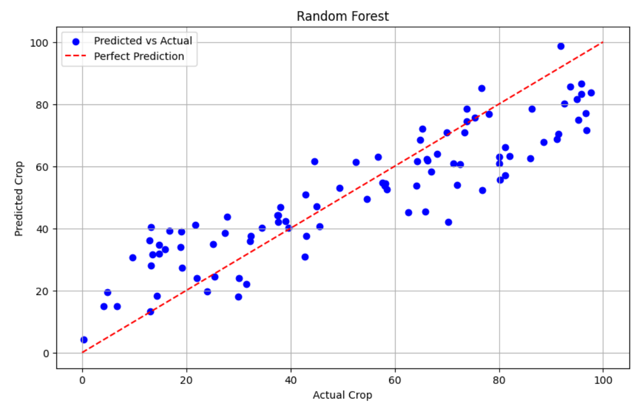

Figure 8.2 – Random Forest Graph

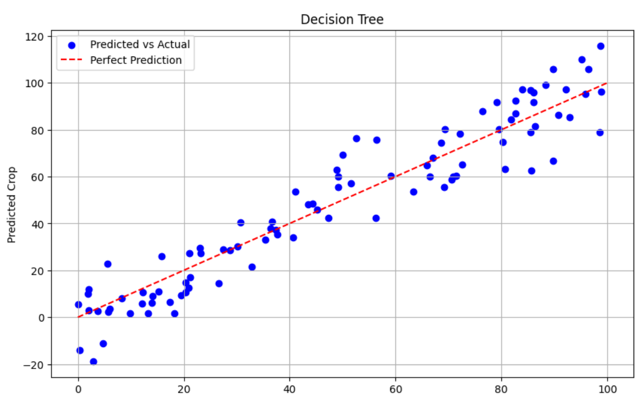

Figure 8.3 – Decision Tree Graph

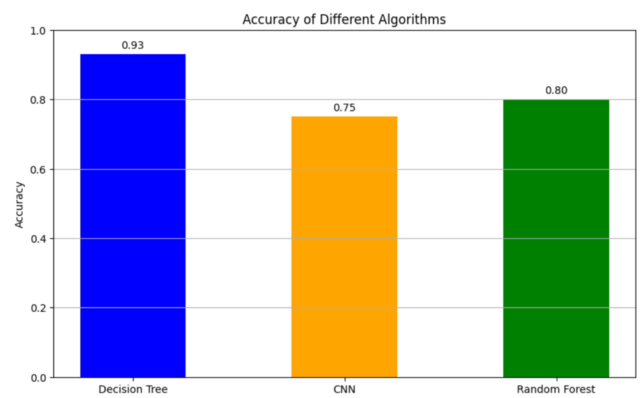

Figure 8.4 – Accuracy Comparison Graph

In this section, the visual representations of our mobile application, showcasing the various pages integral to our prediction system. These screenshots offer a comprehensive overview of the user interface and functionalities incorporated within the application, providing insight into the seamless navigation and user experience. The home page of the mobile application is depicted in figure 8.5 and the product description and the search bar are shown in figure 8.6 and 8.7 respectively.

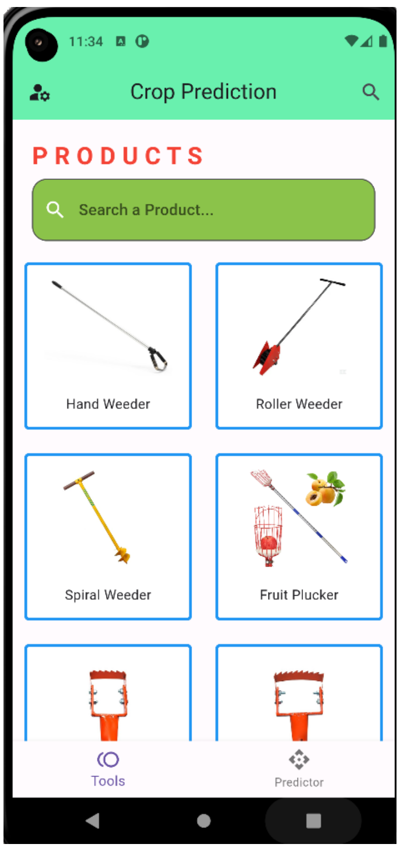

Figure 8.5 - *App Home Page*

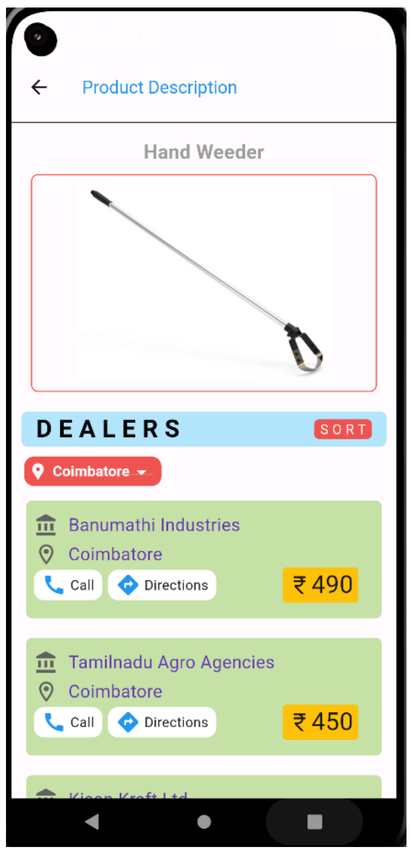

Figure 8.6 - *Product Description Page*

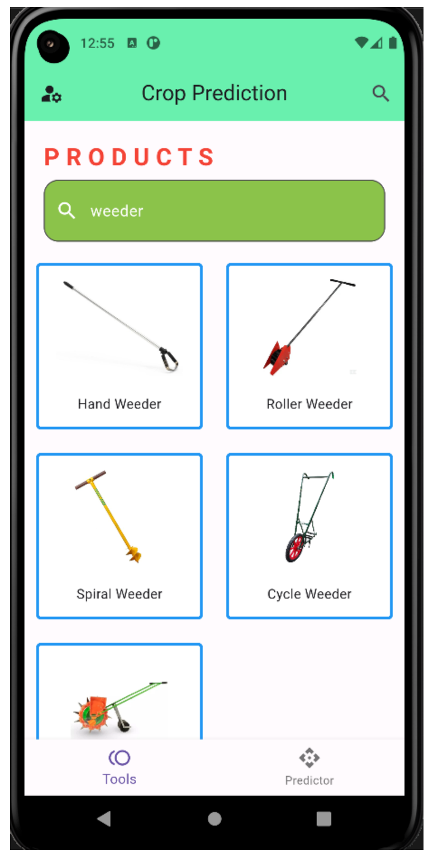

Figure 8.7 - *Search Bar*

The Crop predictor page is depicted in figure 8.8 and the predicted crop output screen and the suggested tools details are depicted in figure 8.9 and 8.10 respectively.

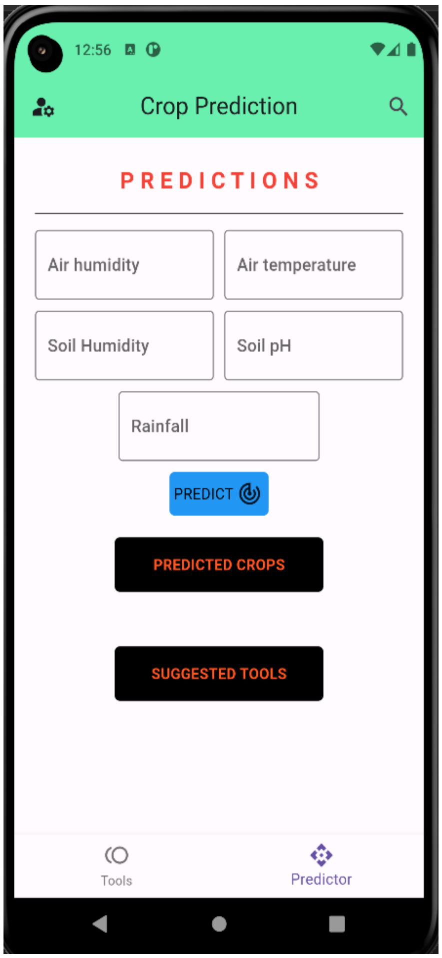

Figure 8.8 - *Crop Prediction Page*

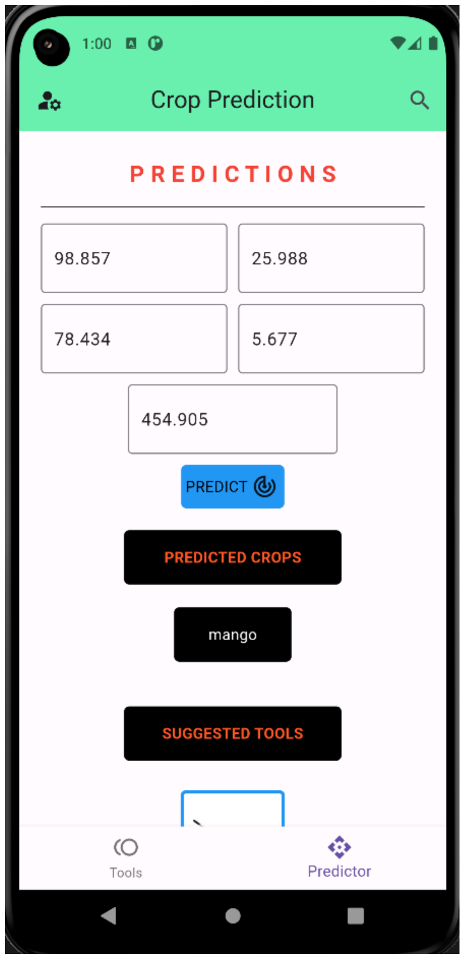

Figure 8.9 - *Predicted Crop Column*

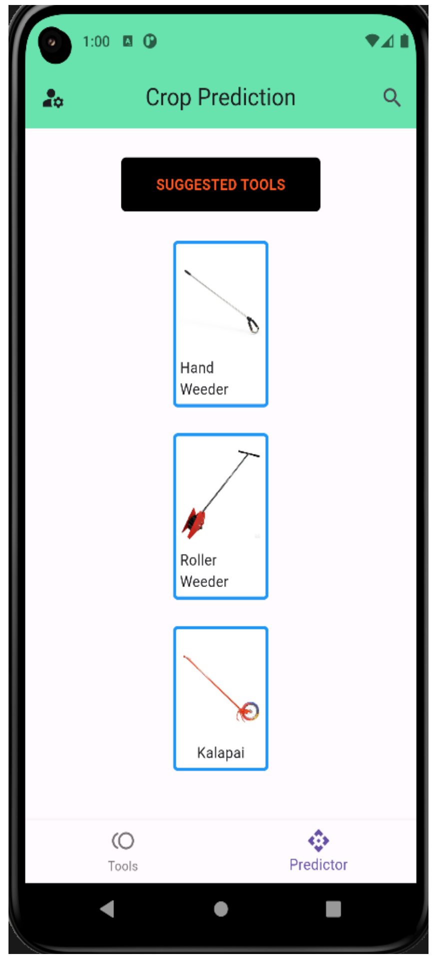

Figure 8.10 - *Suggested Tools*

9. CONCLUSION AND FUTURE ENHANCEMENT

9.1 CONCLUSION

In summary, this project titled "Crop prediction and agricultural tool recommendation system using ML" represents a significant advancement in farming methods. Our goal is to empower farmers with the necessary tools and knowledge essential for success in today's agriculture. By combining crop prediction and tool recommendations into a user-friendly mobile application, we aim to simplify decision-making for farmers by providing them with accurate and timely information based on comprehensive data analysis. This initiative is designed to support sustainable agricultural practices and promote the evolution of precision farming. By harnessing the power of machine learning, we can offer farmers insights into crop trends and optimal tools, enabling them to make informed decisions that maximize efficiency and productivity while minimizing environmental impact. Our holistic approach not only addresses the immediate needs of farmers but also contributes to the long-term resilience and prosperity of agriculture. We believe that by empowering farmers with the tools and knowledge they need to succeed, we can drive positive change in the industry and create a more sustainable future for generations to come. Together, we can revolutionize farming practices and ensure a prosperous and resilient future for agriculture.

9.2 FUTURE ENHANCEMENT

In the future, vendor details from all around Tamil Nadu will be collected and integrated into the mobile application. Additionally, essential parameters for crop prediction, such as soil pH, rainfall, soil moisture, and temperature, will be obtained in real-time using various IoT sensors. This data will then be transmitted to the server for processing.

APPENDIX

SOURCE CODE

MAIN.DART:

```
import 'package:agri_app/Pages/splash_screen.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: SplashScreen(),
    );
  }
}
```

SPLASH SCREEN.DART :

```
import "package:agri_app/Pages/home_page.dart";
import "package:flutter/material.dart";
import "package:flutter/services.dart";
class SplashScreen extends StatefulWidget {
  const SplashScreen({super.key});
  @override
  State<SplashScreen> createState() => _SplashScreenState();
}
class _SplashScreenState extends State<SplashScreen>
    with SingleTickerProviderStateMixin {
  @override
  void initState() {
    super.initState();
    SystemChrome.setEnabledSystemUIMode(SystemUiMode.immersive);
    Future.delayed(const Duration(seconds: 4), () {
      Navigator.of(context)
          .pushReplacement(MaterialPageRoute(builder: (_) => HomePage()));
    });
  }

  @override
```

```
  void dispose() {
    SystemChrome.setEnabledSystemUIMode(SystemUiMode.manual,
        overlays: SystemUiOverlay.values);
    super.dispose();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        width: double.infinity,
        decoration: BoxDecoration(
          gradient: LinearGradient(
              colors: [Colors.pink, Colors.blue],
              begin: Alignment.topCenter,
              end: Alignment.bottomCenter),
        ),
        child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [Text("Splash"), Icon(Icons.home)]),
      ),
    );
  }
}
```

HOME PAGE.DART:

```
import 'package:agri_app/Pages/predictor_page.dart';
import 'package:agri_app/Pages/search_dialog.dart';
import 'package:agri_app/Pages/tools_page.dart';
import 'package:flutter/material.dart';
class HomePage extends StatefulWidget {
  const HomePage({super.key});
  @override
  State<HomePage> createState() => _HomePageState();
}
class _HomePageState extends State<HomePage> {
  int _currentpage = 0;
  List Screens = [ToolsPage(), PredictorPage()];
  void changePage(int index) {
    setState(() {
      _currentpage = index;
    });
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Crop Prediction"),
        centerTitle: true,
        backgroundColor: Colors.greenAccent,
        leading: Icon(
```

```
          Icons.manage_accounts_sharp,
          size: 25,
        ),
        actions: [
          IconButton(
              onPressed: () {
                showDialog(
                    context: context,
                    builder: (BuildContext context) {
                      return SearchDialog();
                    });
              },
              icon: Icon(Icons.search_outlined))
        ],
      ),
      body: Screens[_currentpage],
      bottomNavigationBar: BottomNavigationBar(
          currentIndex: _currentpage,
          onTap: changePage,
          items: [
            BottomNavigationBarItem(
                icon: Icon(Icons.toll_outlined), label: "Tools"),
            BottomNavigationBarItem(icon: Icon(Icons.api), label: "Predictor")
          ]),
    );
  }
}
```

TOOLS PAGE.DART:

```
import 'package:agri_app/Pages/product_description.dart';
import 'package:agri_app/Pages/tool_card.dart';
import 'package:flutter/material.dart';
class ToolsPage extends StatefulWidget {
  const ToolsPage({super.key});
  @override
  State<ToolsPage> createState() => _ToolsPageState();
}
class _ToolsPageState extends State<ToolsPage> {
  List main_toolList = [
    {'name': 'Hand Weeder', 'img_path': 'lib/Assets/1.jpg'},
    {'name': 'Roller Weeder', 'img_path': 'lib/Assets/2.jpg'},
    {'name': 'Spiral Weeder', 'img_path': 'lib/Assets/3.jpg'},
    {'name': 'Fruit Plucker', 'img_path': 'lib/Assets/4.jpg'},
    {'name': 'kalai Vetti (3 inch)', 'img_path': 'lib/Assets/5.jpg'},
    {'name': 'kalai Vetti (4 inch)', 'img_path': 'lib/Assets/6.jpg'},
    {'name': 'kalai Vetti (6 inch)', 'img_path': 'lib/Assets/7.jpg'},
    {'name': 'kalai Vetti (8 inch)', 'img_path': 'lib/Assets/8.jpg'},
    {'name': 'kalai Vetti Handle', 'img_path': 'lib/Assets/9.jpg'},
    {'name': 'Cycle Weeder', 'img_path': 'lib/Assets/10.jpg'},
    {'name': 'GroundNut Weeder', 'img_path': 'lib/Assets/11.jpg'},
```

```
  {'name': 'Moonu Kalapai Blade', 'img_path': 'lib/Assets/12.jpg'},
  {'name': 'Moonu Kalapai', 'img_path': 'lib/Assets/13.jpg'},
  {'name': 'Annchu Kalapai Blade', 'img_path': 'lib/Assets/14.jpg'},
  {'name': 'Annchu Kalapai', 'img_path': 'lib/Assets/15.jpg'},
  {'name': 'Auger (2in)', 'img_path': 'lib/Assets/16.jpg'},
  {'name': 'Auger (3in)', 'img_path': 'lib/Assets/17.jpg'},
  {'name': 'Auger (4in)', 'img_path': 'lib/Assets/18.jpg'},
  {'name': 'Auger (5in)', 'img_path': 'lib/Assets/19.jpg'},
  {'name': 'Auger (6in)', 'img_path': 'lib/Assets/20.jpg'},
  {'name': 'Auger (8in)', 'img_path': 'lib/Assets/21.jpg'},
  {'name': 'Auger (12in)', 'img_path': 'lib/Assets/22.jpg'},
  {'name': 'Corn Breaker', 'img_path': 'lib/Assets/23.jpg'},
  {'name': 'GroundNut Breaker', 'img_path': 'lib/Assets/24.jpg'},
];
List _toolList = [];
@override
void initState() {
  super.initState();
  _toolList.addAll(main_toolList);
}
void updateList(String value) {
  setState(() {
    _toolList = main_toolList
        .where((element) =>
            element['name'].toLowerCase().contains(value.toLowerCase()))
        .toList();
  });
}
void routeTodescription(toolName, toolImg) {
  Navigator.push(
      context,
      MaterialPageRoute(
          builder: (context) => ProductPage(
                productName: toolName,
                productimg: toolImg,
              )));
}
@override
Widget build(BuildContext context) {
  return Scaffold(
      body: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Padding(
        padding: const EdgeInsets.only(top: 20, left: 20, right: 20),
        child: Text(
          "P R O D U C T S",
          style: TextStyle(
              color: Colors.red, fontSize: 25, fontWeight: FontWeight.bold),
        ),
```

```
        ),
        SizedBox(
          height: 5,
        ),
        Padding(
          padding:
              const EdgeInsets.only(top: 0, left: 20, right: 20, bottom: 10),
          child: TextField(
            onChanged: (value) => updateList(value),
            style: TextStyle(color: Colors.white),
            decoration: InputDecoration(
                filled: true,
                fillColor: Colors.lightGreen,
                border: OutlineInputBorder(
                  borderRadius: BorderRadius.circular(15),
                ),
                hintText: "Search a Product...",
                prefixIcon: Icon(Icons.search),
                prefixIconColor: Colors.white),
          ),
        ),
        Expanded(
          child: _toolList.length == 0
              ? Center(
                  child: Text(
                  "No Data Found",
                  style: TextStyle(
                      color: Colors.grey,
                      fontSize: 30,
                      fontWeight: FontWeight.bold),
                ))
              : GridView.builder(
                  itemCount: _toolList.length,
                  gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
                      crossAxisCount: 2),
                  itemBuilder: (context, ind) {
                    return Toolcard(
                      toolName: _toolList[ind]['name'],
                      imgPath: _toolList[ind]['img_path'],
                      onClick: () => {
                        routeTodescription(
                            _toolList[ind]['name'], _toolList[ind]['img_path'])
                      },
                    );
                  }),
        ),
      ],
    ));
  }
}
```

TOOL CARD.DART:

```
import "package:flutter/material.dart";
class Toolcard extends StatelessWidget {
  final toolName;
  final imgPath;
  final Function()? onClick;
  const Toolcard({super.key, this.toolName, this.imgPath, this.onClick});
  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      onTap: onClick,
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Container(
          padding: EdgeInsets.all(4),
          margin: EdgeInsets.all(4),
          width: double.infinity,
          decoration: BoxDecoration(
              color: Colors.white,
              border: Border.all(width: 3, color: Colors.blue),
              borderRadius: BorderRadius.circular(5)),
          child: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.spaceAround,
              children: [
                Image.asset(
                  imgPath,
                  height: 100,
                  width: 200,
                ),
                Text(toolName)
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

SEARCH DIALOG.DART:

```
import "package:flutter/material.dart";
class SearchDialog extends StatefulWidget {
  const SearchDialog({super.key});
  @override
  State<SearchDialog> createState() => _SearchDialogState();
}
class _SearchDialogState extends State<SearchDialog> {
```

```
  @override
  Widget build(BuildContext context) {
    return AlertDialog(
      title: Text("Search"),
      content: TextField(
        style: TextStyle(color: Colors.white),
        decoration: InputDecoration(
            filled: true,
            fillColor: Colors.black,
            border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(8),
                borderSide: BorderSide.none),
            hintText: "Search..",
            prefixIcon: Icon(Icons.search),
            prefixIconColor: Colors.white),
      ),
      actions: [
        TextButton(onPressed: () {}, child: Text("Search")),
        TextButton(
            onPressed: () {
              Navigator.pop(context);
            },
            child: Text("Close")),
      ],
    );
  }
}
```

PRODUCT DESCRIPTION PAGE :

```
import 'package:agri_app/Pages/dealerList.dart';
import 'package:agri_app/Pages/dealer_card.dart';
import 'package:flutter/material.dart';
class ProductPage extends StatefulWidget {
  final productName;
  final productimg;
  ProductPage({super.key, this.productName, this.productimg});
  @override
  State<ProductPage> createState() => _ProductPageState();
}
class _ProductPageState extends State<ProductPage> {
  var dropDownValue = 'Coimbatore';
  List<Map<String, dynamic>> sortedDealerList = [];
  bool identify = false;
  void _sortDealerlist(List<Map<String, dynamic>> dealerList) {
    setState(() {
      sortedDealerList = List<Map<String, dynamic>>.from(dealerList)
        ..sort((a, b) => a['price'].compareTo(b['price']));
      identify = !identify;
    });
  }
```

```
@override
Widget build(BuildContext context) {
  Dealer obj = Dealer();
  List<Map<String, dynamic>> dealerList = obj.DealerList[widget.productName];
  return Scaffold(
      appBar: AppBar(
        title: Text(
          "Product Description",
          style: TextStyle(color: Colors.blue, fontSize: 18),
        ),
        leading: IconButton(
            onPressed: () => Navigator.pop(context),
            icon: Icon(Icons.arrow_back)),
        backgroundColor: Colors.transparent,
      ),
      body: Column(
        // mainAxisAlignment: MainAxisAlignment.start,
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Divider(
            color: Colors.grey.shade900,
          ),
          Padding(
            padding: const EdgeInsets.all(8.0),
            child: Container(
              width: double.infinity,
              child: Center(
                  child: Text(
                widget.productName,
                style: TextStyle(
                    color: Colors.grey,
                    fontSize: 20,
                    fontWeight: FontWeight.bold),
              )),
            ),
          ),
          Container(
            margin: EdgeInsets.symmetric(horizontal: 20),
            width: double.infinity,
            decoration: BoxDecoration(
                border: Border.all(color: Colors.red),
                borderRadius: BorderRadius.circular(7)),
            child: Padding(
              padding: EdgeInsets.all(10),
              child: Image.asset(
                widget.productimg,
                height: 200,
              ),
            ),
          ),
```

```
SizedBox(
  height: 20,
),
Container(
  margin: EdgeInsets.symmetric(horizontal: 10),
  decoration: BoxDecoration(
    color: Colors.lightBlue[100],
    borderRadius: BorderRadius.circular(7),
  ),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      Padding(
        padding: const EdgeInsets.only(left: 15),
        child: Text(
          "D E A L E R S",
          style: TextStyle(
            color: Colors.black,
            fontSize: 25,
            fontWeight: FontWeight.bold,
          ),
        ),
      ),
      Padding(
        padding: const EdgeInsets.only(right: 15),
        child: GestureDetector(
          onTap: () {
            _sortDealerlist(dealerList);
          },
          child: Container(
            decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(5),
                color: Colors.red[400]),
            child: Padding(
              padding: const EdgeInsets.symmetric(horizontal: 6),
              child: Text(
                "S O R T",
                style: TextStyle(color: Colors.white),
              ),
            ),
          ),
        ),
      )
    ],
  ),
),
SizedBox(
  height: 10,
),
Padding(
```

```
  padding: const EdgeInsets.only(left: 13),
  child: Container(
    padding: EdgeInsets.all(4),
    height: 30,
    width: 140,
    decoration: BoxDecoration(
        borderRadius: BorderRadius.circular(10),
        color: Colors.red[400]),
    child: Row(
      children: [
        Icon(
          Icons.location_pin,
          size: 20,
          color: Colors.white,
        ),
        SizedBox(
          width: 5,
        ),
        DropdownButton(
          iconEnabledColor: Colors.white,
          style: TextStyle(
              color: Colors.white,
              fontSize: 15,
              fontWeight: FontWeight.bold),
          value: dropDownValue,
          underline: null,
          dropdownColor: Colors.greenAccent,
          onChanged: (String? newValue) {
            setState() {
              dropDownValue = newValue!;
            }
          },
          items: const [
            DropdownMenuItem(
                value: "Coimbatore", child: Text("Coimbatore"))
          ],
        )
      ],
    ),
  ),
),

SizedBox(
  height: 5,
),
Expanded(
  child: identify == false
      ? ListView.builder(
          shrinkWrap: true,
          itemCount: obj.DealerList[widget.productName].length,
```

```
              itemBuilder: (context, ind) => DealerCard(
                  dealerName: obj.DealerList[widget.productName][ind]
                    ['name'],
                  dealerAddress: obj.DealerList[widget.productName]
                    [ind]['address'],
                  productPrice: obj.DealerList[widget.productName]
                    [ind]['price'],
                  mobNumber: obj.DealerList[widget.productName][ind]
                    ['contact'],
                ))
            : ListView.builder(
              shrinkWrap: true,
              itemCount: sortedDealerList.length,
              itemBuilder: (context, ind) => DealerCard(
                  dealerName: sortedDealerList[ind]['name'],
                  dealerAddress: sortedDealerList[ind]['address'],
                  productPrice: sortedDealerList[ind]['price'],
                  mobNumber: sortedDealerList[ind]['contact'],
                )),
        )
      ],
    ));
  }
}
```

DEALER CARD.DART:

```
import 'package:agri_app/Pages/google_map.dart';
import 'package:flutter/material.dart';
import 'package:url_launcher/url_launcher.dart';
class DealerCard extends StatelessWidget {
  final String dealerName;
  final String dealerAddress;
  final String productPrice;
  final String mobNumber;
  const DealerCard(
      {super.key,
      required this.dealerName,
      required this.dealerAddress,
      required this.productPrice,
      required this.mobNumber});
  @override
  Widget build(BuildContext context) {
    return Container(
      margin: EdgeInsets.symmetric(vertical: 10, horizontal: 15),
      padding: EdgeInsets.symmetric(vertical: 4),
      height: 120,
      width: double.infinity,
      decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(6),
          color: Colors.lightGreen[200]),
```

```
child: Padding(
  padding: const EdgeInsets.symmetric(vertical: 8, horizontal: 8),
  child: Column(
    children: [
      Row(
        children: [
          Icon(
            Icons.account_balance_sharp,
            size: 25,
            color: Colors.grey[700],
          ),
          SizedBox(
            width: 10,
          ),
          Text(
            dealerName,
            style: TextStyle(fontSize: 18, color: Colors.deepPurple),
          )
        ],
      ),
      SizedBox(
        height: 4,
      ),
      Row(
        children: [
          Icon(
            Icons.location_on_outlined,
            size: 25,
            color: Colors.grey[700],
          ),
          SizedBox(
            width: 10,
          ),
          Text(
            dealerAddress,
            style: TextStyle(fontSize: 18, color: Colors.deepPurple),
          )
        ],
      ),
      Row(
        children: [
          GestureDetector(
            onTap: () async {
              final Uri url = Uri(
                scheme: 'tel',
                path: mobNumber,
              );
              if (await canLaunchUrl(url)) {
                await launchUrl(url);
              } else {
```

```
        print("cannot launch this url");
      }
    },
    child: Container(
      padding: EdgeInsets.symmetric(vertical: 3, horizontal: 8),
      decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(10),
          color: Colors.white),
      child: Row(
        children: [
          Icon(
            Icons.call,
            size: 25,
            color: Colors.blue,
          ),
          SizedBox(
            width: 4,
          ),
          Text("Call")
        ],
      ),
    ),
  ),
  SizedBox(
    width: 6,
  ),
  GestureDetector(
    onTap: () {
      MapUtils.openMap(30.625168, 131.751500);
    },
    child: Container(
      padding: EdgeInsets.symmetric(vertical: 3, horizontal: 8),
      decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(10),
          color: Colors.white),
      child: Row(
        children: [
          Icon(
            Icons.directions,
            size: 25,
            color: Colors.blue,
          ),
          SizedBox(
            width: 4,
          ),
          Text("Directions")
        ],
      ),
    ),
  ),
```

```
                SizedBox(
                  width: 68,
                ),
                Container(
                  padding: EdgeInsets.symmetric(
                    horizontal: 5,
                  ),
                  decoration: BoxDecoration(
                      color: Colors.amber,
                      borderRadius: BorderRadius.circular(4)),
                  child: Row(
                    children: [
                      Icon(Icons.currency_rupee_sharp),
                      Text(
                        productPrice,
                        style: TextStyle(fontSize: 25),
                      )
                    ],
                  ),
                )
              ],
            )
          ],
        ),
      ),
    );
  }
}
```

PREDICTOR PAGE.DART :

```
import 'dart:convert';
import 'package:agri_app/Pages/predicted_product_description_page.dart';
import 'package:agri_app/Pages/predicted_tools_list.dart';
import 'package:agri_app/Pages/predicted_tool_card.dart';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
class PredictorPage extends StatefulWidget {
  const PredictorPage({super.key});
  @override
  State<PredictorPage> createState() => _PredictorPageState();
}

class _PredictorPageState extends State<PredictorPage> {
  String? urlResponse;
  Map? mapResponse;
  Map? dataResponse;
  List? listResponseTools;
  String? listResponseCrops;
  final _airHumidityController = TextEditingController();
```

```
final _airTempController = TextEditingController();
final _soilHumidityController = TextEditingController();
final _pHController = TextEditingController();
final _rainfallController = TextEditingController();
String airHumidity = '';
String airTemp = '';
String soilHumidity = '';
String pHValue = '';
String rainfall = '';
Future<void> apicall() async {
  try {
    http.Response response = await http.get(Uri.parse(
        "http://192.168.31.100:8000/predict/$airHumidity/$airTemp/$soilHumidity/$pHValue/$rainfall"));
    if (response.statusCode == 200) {
      setState(() {
        // urlResponse = response.body;
        mapResponse = json.decode(response.body);
        listResponseTools = mapResponse?['Predicted_Tools'];
        listResponseCrops = mapResponse?['Recommended_Crops'];
      });
      // print(listResponse?[0].toString());
      // print(listResponse?[1].toString());
      // print(listResponse?[2].toString());
    } else {
      // Handle non-200 status code
      print("Failed to fetch data: ${response.statusCode}");
    }
  } catch (e) {
    // Handle connection errors
    print("Failed to connect to the server: $e");
  }
}

void routeTodescription(toolName, toolImg) {
  Navigator.push(
      context,
      MaterialPageRoute(
          builder: (context) => PredictedProductPage(
                productName: toolName,
                productimg: toolImg,
              )));
}
@override
Widget build(BuildContext context) {
  PredictedList obj = PredictedList();
  return Scaffold(
      // backgroundColor: Colors.blue[300],
      body: SingleChildScrollView(
    child: Padding(
      padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 25),
```

```
child: Column(
  mainAxisAlignment: MainAxisAlignment.center,
  crossAxisAlignment: CrossAxisAlignment.center,
  children: [
    Padding(
      padding: const EdgeInsets.symmetric(horizontal: 80),
      child: Text(
        "P R E D I C T I O N S",
        style: TextStyle(
            color: Colors.red,
            fontWeight: FontWeight.bold,
            fontSize: 21),
      ),
    ),
    SizedBox(
      width: double.infinity,
      height: 30,
      child: Divider(
        color: Colors.grey.shade900,
      ),
    ),
    Container(
      child: Column(
        children: [
          Row(
            children: [
              Expanded(
                child: TextField(
                  controller: _airHumidityController,
                  decoration: InputDecoration(
                    hintText: "Air humidity",
                    border: OutlineInputBorder(),
                  ),
                ),
              ),
              SizedBox(
                width: 10,
              ),
              Expanded(
                child: TextField(
                  controller: _airTempController,
                  decoration: InputDecoration(
                    hintText: "Air temperature",
                    border: OutlineInputBorder(),
                  ),
                ),
              )
            ],
          ),
          SizedBox(
```

```
  height: 10,
),
Row(
  children: [
    Expanded(
      child: TextField(
        controller: _soilHumidityController,
        decoration: InputDecoration(
          hintText: "Soil Humidity",
          border: OutlineInputBorder(),
        ),
      ),
    ),
    SizedBox(
      width: 10,
    ),
    Expanded(
      child: TextField(
        controller: _pHController,
        decoration: InputDecoration(
          hintText: "Soil pH",
          border: OutlineInputBorder(),
        ),
      ),
    )
  ],
),
SizedBox(
  height: 10,
),
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    SizedBox(
      width: 80,
    ),
    Expanded(
      child: TextField(
        controller: _rainfallController,
        decoration: InputDecoration(
          hintText: "Rainfall",
          border: OutlineInputBorder(),
        ),
      ),
    ),
    SizedBox(
      width: 80,
    )
  ],
)
```

```
        ],
      ),
    ),
    SizedBox(
      height: 10,
    ),
    GestureDetector(
      onTap: () {
        setState(() {
          airHumidity = _airHumidityController.text;
          airTemp = _airTempController.text;
          soilHumidity = _soilHumidityController.text;
          pHValue = _pHController.text;
          rainfall = _rainfallController.text;
        });
        apicall();
      },
      child: Container(
        padding: EdgeInsets.all(4),
        height: 40,
        width: 95,
        decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(5), color: Colors.blue),
        child: Row(
          children: [
            Text(
              "PREDICT ",
              style: TextStyle(color: Colors.black),
            ),
            Icon(Icons.track_changes_sharp)
          ],
        ),
      ),
    ),
    SizedBox(
      height: 20,
    ),
    Container(
      padding: EdgeInsets.symmetric(horizontal: 35),
      height: 50,
      width: 200,
      decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(5), color: Colors.black),
      child: Center(
        child: Text(
          "PREDICTED CROPS",
          style: TextStyle(
              color: Colors.deepOrange, fontWeight: FontWeight.bold),
        ),
      ),
```

```
),
Container(
  width: 150,
  // color: Colors.blue,
  padding: EdgeInsets.all(15),
  child: ListView.builder(
      shrinkWrap: true,
      itemCount: listResponseCrops != null ? 1 : 0,
      // listResponseCrops == null ? 0 : listResponseCrops?.length,
      itemBuilder: (context, index) {
        return Container(
          margin: EdgeInsets.all(6),
          padding: EdgeInsets.all(5),
          height: 50,
          // width: 30,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(5),
            color: Colors.black,
          ),
          child: Center(
            child: Text(
              listResponseCrops ?? '',
              style: TextStyle(color: Colors.white),
            ),
          ),
        );
      }),
),
SizedBox(
  height: 20,
),

Container(
  height: 50,
  width: 200,
  decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(5), color: Colors.black),
  child: Center(
    child: Text(
      "SUGGESTED TOOLS",
      style: TextStyle(
          color: Colors.deepOrange, fontWeight: FontWeight.bold),
    ),
  ),
),
Container(
  width: 150,
  // color: Colors.blue,
  padding: EdgeInsets.all(15),
  child: ListView.builder(
```

```
                shrinkWrap: true,
                itemCount:
                    listResponseTools == null ? 0 : listResponseTools?.length,
                itemBuilder: (context, index) {
                  return PredictedToolCard(
                    predicted_tool_name: listResponseTools?[index],
                    predicted_img_path:
                        obj.PredictedToolList[listResponseTools?[index]]?[0]
                            ['img_path'],
                    onClick: () {
                      routeTodescription(
                          listResponseTools?[index],
                          obj.PredictedToolList[listResponseTools?[index]]?[0]
                              ['img_path']);
                    },
                  );
                }),
          )
        ],
      ),
    ),
  ));
 }
}
```

PREDICTED TOOL CARD.DART:

```
import "package:flutter/material.dart";
class PredictedToolCard extends StatelessWidget {
  final predicted_tool_name;
  final predicted_img_path;
  final Function()? onClick;
  const PredictedToolCard(
      {super.key,
      this.predicted_tool_name,
      this.predicted_img_path,
      this.onClick});
  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      onTap: onClick,
      child: Padding(
        padding: const EdgeInsets.all(8.0),
        child: Container(
          padding: EdgeInsets.all(4),
          margin: EdgeInsets.all(4),
          width: double.infinity,
          decoration: BoxDecoration(
              color: Colors.white,
              border: Border.all(width: 3, color: Colors.blue),
              borderRadius: BorderRadius.circular(5)),
```

```
          child: Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.spaceAround,
              children: [
                Image.asset(
                  predicted_img_path,
                  height: 100,
                  width: 200,
                ),
                Text(predicted_tool_name)
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```

PREDICTED PRODUCT DESCRIPTION PAGE.DART:

```
import 'package:agri_app/Pages/dealer_card.dart';
import 'package:agri_app/Pages/predicted_tools_list.dart';
import 'package:flutter/material.dart';
class PredictedProductPage extends StatefulWidget {
  final productName;
  final productimg;
  PredictedProductPage({super.key, this.productName, this.productimg});
  @override
  State<PredictedProductPage> createState() => _PredictedProductPageState();
}
class _PredictedProductPageState extends State<PredictedProductPage> {
  var dropDownValue = 'Coimbatore';
  List<Map<String, dynamic>> sortedDealerList = [];
  bool identify = false;

  void sortDealerList(List<Map<String, dynamic>> dealerList) {
    setState(() {
      sortedDealerList =
          List<Map<String, dynamic>>.from(dealerList[1]['dealers'])
            ..sort((a, b) => a['price'].compareTo(b['price']));
      identify = !identify;
    });
  }
  @override
  Widget build(BuildContext context) {
    PredictedList obj = PredictedList();
    List<Map<String, dynamic>> dealerList =
        obj.PredictedToolList[widget.productName];
    return Scaffold(
        appBar: AppBar(
```

```
      title: Text(
        "Product Description",
        style: TextStyle(color: Colors.blue, fontSize: 18),
      ),
      leading: IconButton(
          onPressed: () => Navigator.pop(context),
          icon: Icon(Icons.arrow_back)),
      backgroundColor: Colors.transparent,
    ),
    body: Column(
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
        Divider(
          color: Colors.grey.shade900,
        ),
        Padding(
          padding: const EdgeInsets.all(8.0),
          child: Container(
            width: double.infinity,
            child: Center(
                child: Text(
              widget.productName,
              style: TextStyle(
                  color: Colors.grey,
                  fontSize: 20,
                  fontWeight: FontWeight.bold),
            )),
          ),
        ),
        Container(
          margin: EdgeInsets.symmetric(horizontal: 20),
          width: double.infinity,
          decoration: BoxDecoration(
              border: Border.all(color: Colors.red),
              borderRadius: BorderRadius.circular(7)),
          child: Padding(
            padding: EdgeInsets.all(10),
            child: Image.asset(
              widget.productimg,
              height: 200,
            ),
          ),
        ),
        SizedBox(
          height: 20,
        ),
        Container(
          margin: EdgeInsets.symmetric(horizontal: 10),
          decoration: BoxDecoration(
            color: Colors.lightBlue[100],
```

```
        borderRadius: BorderRadius.circular(7),
      ),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceBetween,
        children: [
          Padding(
            padding: const EdgeInsets.only(left: 15),
            child: Text(
              "D E A L E R S",
              style: TextStyle(
                color: Colors.black,
                fontSize: 25,
                fontWeight: FontWeight.bold,
              ),
            ),
          ),
          Padding(
            padding: const EdgeInsets.only(right: 15),
            child: GestureDetector(
              onTap: () {
                sortDealerList(dealerList);
              },
              child: Container(
                decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(5),
                    color: Colors.red[400]),
                child: Padding(
                  padding: const EdgeInsets.symmetric(horizontal: 6),
                  child: Text(
                    "S O R T",
                    style: TextStyle(color: Colors.white),
                  ),
                ),
              ),
            ),
          )
        ],
      ),
    ),

    //start here
    SizedBox(
      height: 10,
    ),
    Padding(
      padding: const EdgeInsets.only(left: 13),
      child: Container(
        padding: EdgeInsets.all(4),
        height: 30,
        width: 140,
```

```
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      color: Colors.red[400]),
    child: Row(
     children: [
      Icon(
       Icons.location_pin,
       size: 20,
       color: Colors.white,
      ),
      SizedBox(
       width: 5,
      ),
      DropdownButton(
       iconEnabledColor: Colors.white,
       style: TextStyle(
          color: Colors.white,
          fontSize: 15,
          fontWeight: FontWeight.bold),
       value: dropDownValue,
       underline: null,
       dropdownColor: Colors.greenAccent,
       onChanged: (String? newValue) {
        setState() {
         dropDownValue = newValue!;
        }
       },
       items: const [
        DropdownMenuItem(
          value: "Coimbatore", child: Text("Coimbatore"))
       ],
      )
     ],
    ),
   ),
  ),
  SizedBox(
   height: 5,
  ),
  Expanded(
   child: identify == false
     ? ListView.builder(
       shrinkWrap: true,
       itemCount: obj
          .PredictedToolList[widget.productName][1]['dealers']
          .length,
       itemBuilder: (context, ind) => DealerCard(
           dealerName:
             obj.PredictedToolList[widget.productName][1]
               ['dealers'][ind]['name'],
```

```
                    dealerAddress:
                        obj.PredictedToolList[widget.productName][1]
                            ['dealers'][ind]['address'],
                    productPrice:
                        obj.PredictedToolList[widget.productName][1]
                            ['dealers'][ind]['price'],
                    mobNumber: obj.PredictedToolList[widget.productName]
                        [1]['dealers'][ind]['contact'],
                  ))
              : ListView.builder(
                  shrinkWrap: true,
                  itemCount: sortedDealerList.length,
                  itemBuilder: (context, ind) => DealerCard(
                    dealerName: sortedDealerList[ind]['name'],
                    dealerAddress: sortedDealerList[ind]['address'],
                    productPrice: sortedDealerList[ind]['price'],
                    mobNumber: sortedDealerList[ind]['contact'],
                  )),
          )
        ],
      ));
  }
}
```

10. REFERENCES

[1] S. P. Raja, B. Sawicka, Z. Stamenkovic and G. Mariammal, "Crop Prediction Based on Characteristics of the Agricultural Environment Using Various Feature Selection Techniques and Classifiers," *IEEE Access, vol. 10, pp. 23625-23641, 2022.*

[2] A. Mondal and S. Banerjee, "Effective Crop Prediction Using Deep Learning," *International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON), Pune, India, pp. 1-6, 2022.*

[3] M. Aggarwal, A. Kaushik, A. Sengar, A. Gangwar, A. Singh and V. Raj, "Agro App: An application for healthy living," *International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, pp. 30-32, 2021.*

[4] P. Shriram and S. Mhamane, "Android App to Connect Farmers to Retailers and Food Processing Industry," *3rd International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, pp. 284-287, 2022.*

[5] M. Kalimuthu, P. Vaishnavi and M. Kishore, "Crop Prediction using Machine Learning," Third International Conference on Smart Systems and Inventive Technology (ICSSIT), *Tirunelveli, India, pp. 926-932, 2022.*

[6] S. M. Pande, P. K. Ramesh, A. Anmol, B. R. Aishwarya, K. Rohilla and K. Shaurya, "Crop Recommender System Using Machine Learning Approach," *5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 1066-1071, 2022.*

[7] M. Kalimuthu, P. Vaishnavi and M. Kishore, "Crop Prediction using Machine Learning," *Third International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, pp. 926-932, 2022.*

[8] D. VAN GREUNEN, "Mobile Applications in Support of Small-scale Crop Farming," *IST-Africa Conference (IST-Africa), Kampala, Uganda, pp. 1-7, 2022.*

[9] H. S. Negi, S. C. Dimri, B. Kumar and A. Singh, "Crop Prediction Based on Soil Properties using Machine Learning for Smart Farming," *International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, pp. 366-370, 2022.*

[10] C. Rakesh D, V. Vardhan, B. B. Vasantha and G. Sai Krishna, "Crop Recommendation and Prediction System," *9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, pp. 1244-1248, 2023.*

[11] P. Sarma, B. Basumatary and K. Jain, "Smart Farming: A Web-based Approach using ML," *International Conference on Innovative Data Communication Technologies and Application (ICIDCA), Uttarakhand, India, pp. 1-9, 2023.*

[12] J. Sreemathy and N. Prasath, "Crop Recommendation with BiLSTM-MERNN Algorithm for Precision Agriculture," *International Conference on Networking and Communications (ICNWC), Chennai, India, pp. 1-5, 2023.*

[13] M. Aruna Devi, D. Suresh, D. Jeyakumar, D. Swamydoss and M. Lilly Florence, "Agriculture Crop Selection and Yield Prediction using Machine Learning Algorithms," *Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, pp. 510-517, 2022.*

[14] P. Sharma, P. Dadheech, N. Aneja and S. Aneja, "Predicting Agriculture Yields Based on Machine Learning Using Regression and Deep Learning," i*n IEEE Access, vol. 11, pp. 111255-111264, 2023.*

[15] V. Krishna, T. Reddy, S. Harsha, K. Ramar, S. Hariharan and B. A, "Analysis of Crop Yield Prediction using Machine Learning algorithms," *2nd International Conference on Innovative Sustainable Computational Technologies (CISCT), Dehradun, India, pp. 1-4, 2022.*