

Smart Agriculture

"Smart agriculture uses technology to make farming more efficient, sustainable, and data-driven."

Presented By

Mina Samy

Fady Wadee

Fady Atia

Mahmoud Asla

Saif Samir

Mina Yasser

Supervisor

Dr. Elhossiny Ibrahim



Overview



C.V. model



Chatbot



Deployment



Monitoring



Local Data



Why Agriculture ?

Egypt's Cultivated Land

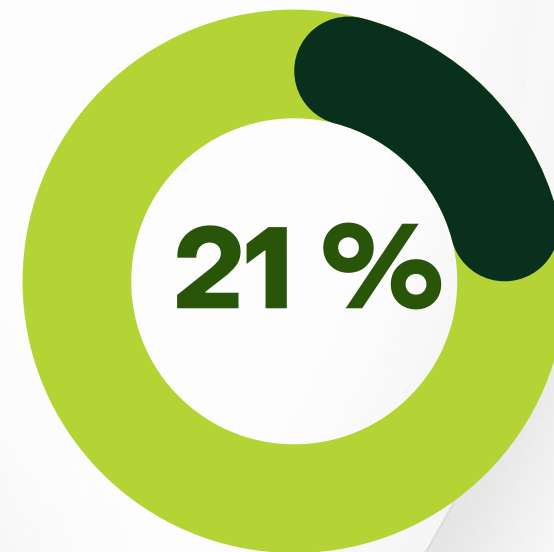
10.4 million feddans

Egypt's Workforce in Farming

5.4 million people



Egypt's Cultivated Land



Egypt's Workforce in Farming

Gross Domestic Product

14.7 % GDP



Project Overview

This mobile app uses AI to detect plant diseases and suggest treatments for better farm management.

- **Take Photo of the plant**
- **Detect Disease**
- **Ask the Chatbot**
- **Suggest Treatment**
- **Monitoring plant**
- **Get Information**



Demo

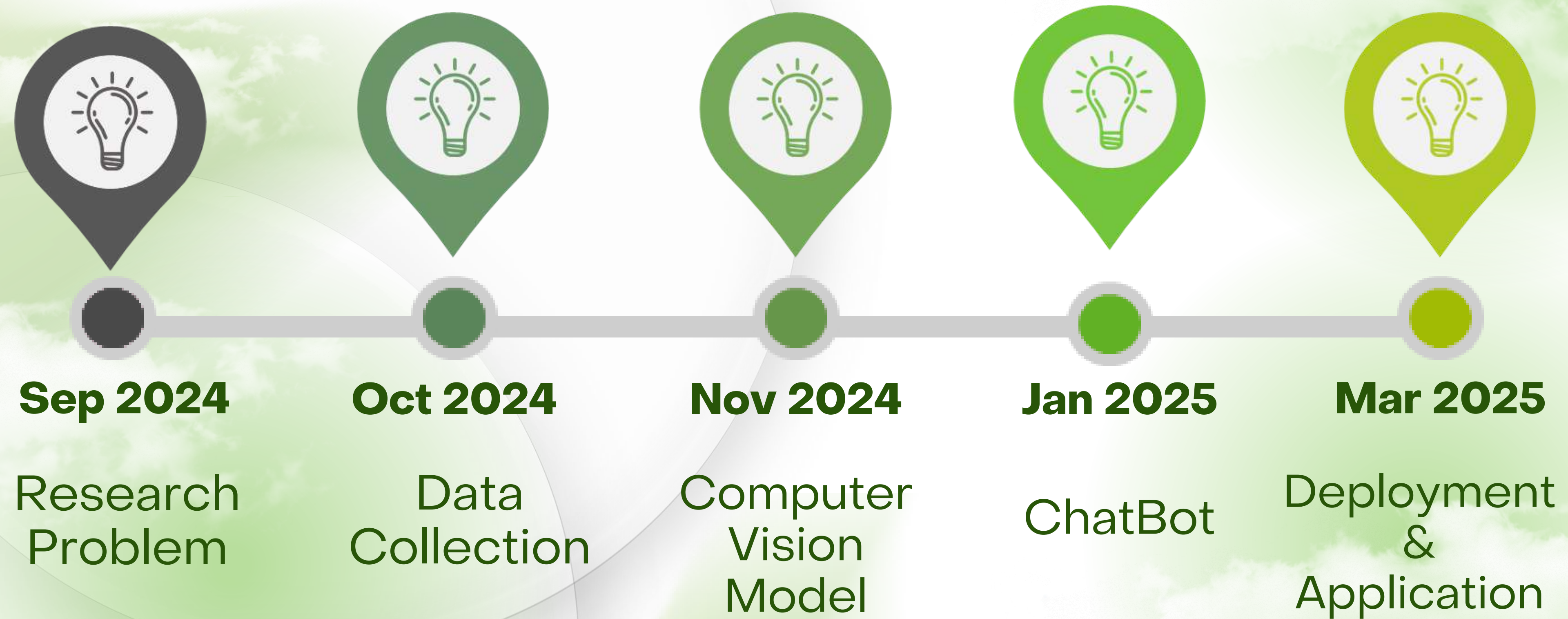


Your Smart Agriculture Companion

Discover smart solutions for your plants with personalized care, disease diagnosis, and weather insights – your ultimate agriculture companion!



Time Plan



Computer Vision Model

Building a CNN-Based
Plant Disease Classifier.



Dataset Description

● **Dataset: New Plant Diseases Dataset**

● **Total Images: ~87,000 RGB images**

● **Classes: 38 (Diseases + Healthy)**

● **Train/Validation/Test: 80% / 20%/33**



Data Explorer

Version 2 (1.43 GB)

- ▼  New Plant Diseases Dataset(Augmented)
 - ▼  New Plant Diseases Dataset(Augmented)
 - ▶  train
 - ▶  valid
 - ▼  test
 - ▶  test

Data Preprocessing



- **Image resizing to 160x160**
- **Label encoding (Text → Integer)**
- **Assign array `x_train ,x_test,x_pred`
`y_train ,y_test`**
- **Convert to NumPy arrays for
model input**



What is a CNN and Usage ?

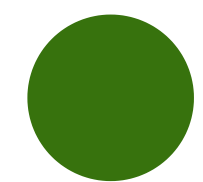
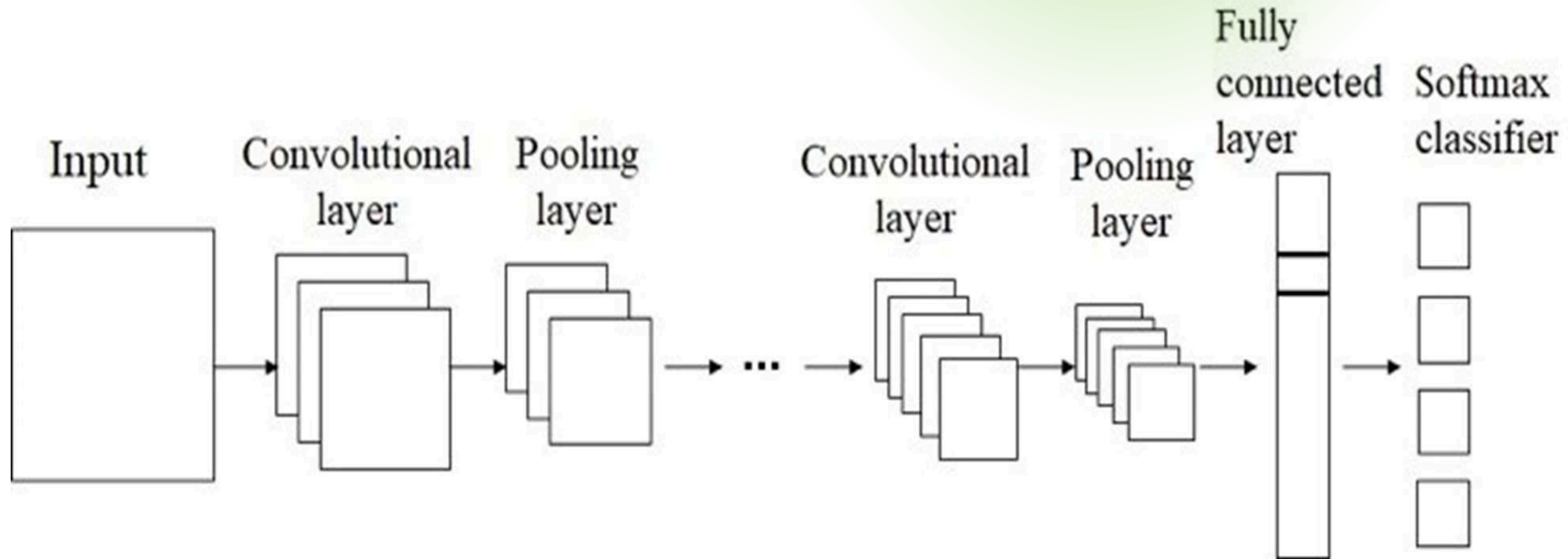
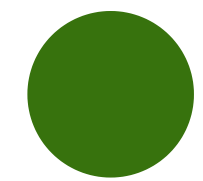
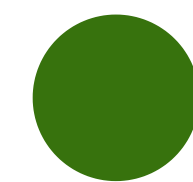


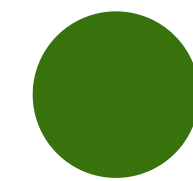
Image Classification



Object Detection



Medical imaging



Facial recognition

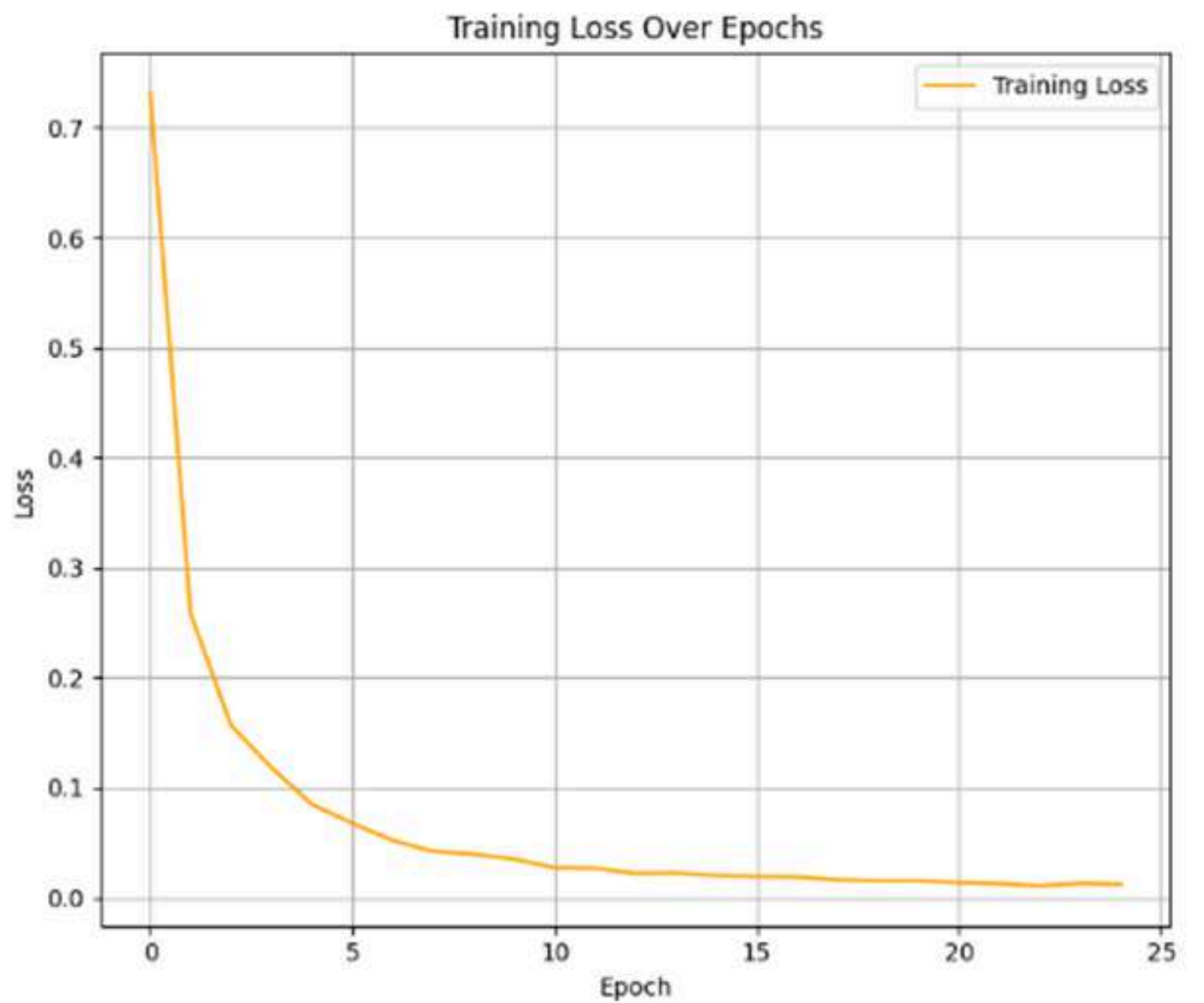
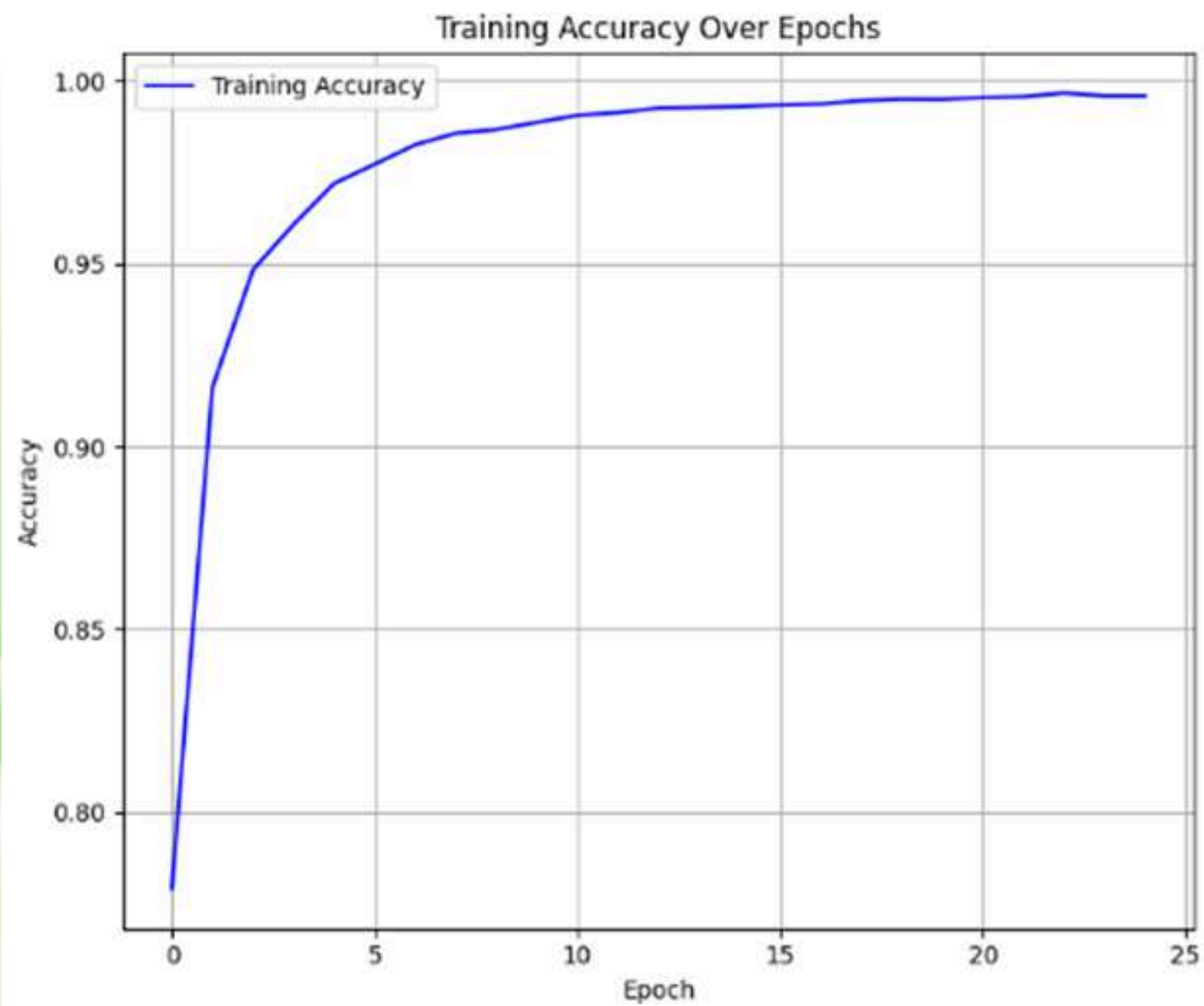
CNN Architecture Used

- **Input: 160*160*3 RGB**
- **4 convolutional blocks (64 → 128 → 256 → 512)**
- **Batch Normalization & MaxPooling & Dropout**
- **Dense(512) + Dense(128) → Softmax(38)**

Layer (type)	Output Shape
conv2d (Conv2D)	(None, 158, 158, 64)
batch_normalization (BatchNormalization)	(None, 158, 158, 64)
conv2d_1 (Conv2D)	(None, 156, 156, 64)
batch_normalization_1 (BatchNormalization)	(None, 156, 156, 64)
max_pooling2d (MaxPooling2D)	(None, 78, 78, 64)
conv2d_2 (Conv2D)	(None, 76, 76, 128)
batch_normalization_2 (BatchNormalization)	(None, 76, 76, 128)
conv2d_3 (Conv2D)	(None, 74, 74, 128)
batch_normalization_3 (BatchNormalization)	(None, 74, 74, 128)
max_pooling2d_1 (MaxPooling2D)	(None, 37, 37, 128)
conv2d_4 (Conv2D)	(None, 35, 35, 256)
batch_normalization_4 (BatchNormalization)	(None, 35, 35, 256)
conv2d_5 (Conv2D)	(None, 33, 33, 256)
batch_normalization_5 (BatchNormalization)	(None, 33, 33, 256)
max_pooling2d_2 (MaxPooling2D)	(None, 16, 16, 256)
conv2d_6 (Conv2D)	(None, 14, 14, 512)
batch_normalization_6 (BatchNormalization)	(None, 14, 14, 512)
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 512)
flatten (Flatten)	(None, 25088)
dense (Dense)	(None, 512)
batch_normalization_7 (BatchNormalization)	(None, 512)
dropout (Dropout)	(None, 512)
dense_1 (Dense)	(None, 128)
batch_normalization_8 (BatchNormalization)	(None, 128)
dense_2 (Dense)	(None, 38)

Model Training & Performance

Dataset	Accuracy	Loss
Training	99.68%	0.0106
Validation	98.25%	0.0732



Model Saving

Model saved in .h5 format





RAG – Arabic AI Chatbot

An intelligent chatbot system in Arabic that answers questions related to plant diseases based on RAG and LLM.



Why This Project ?

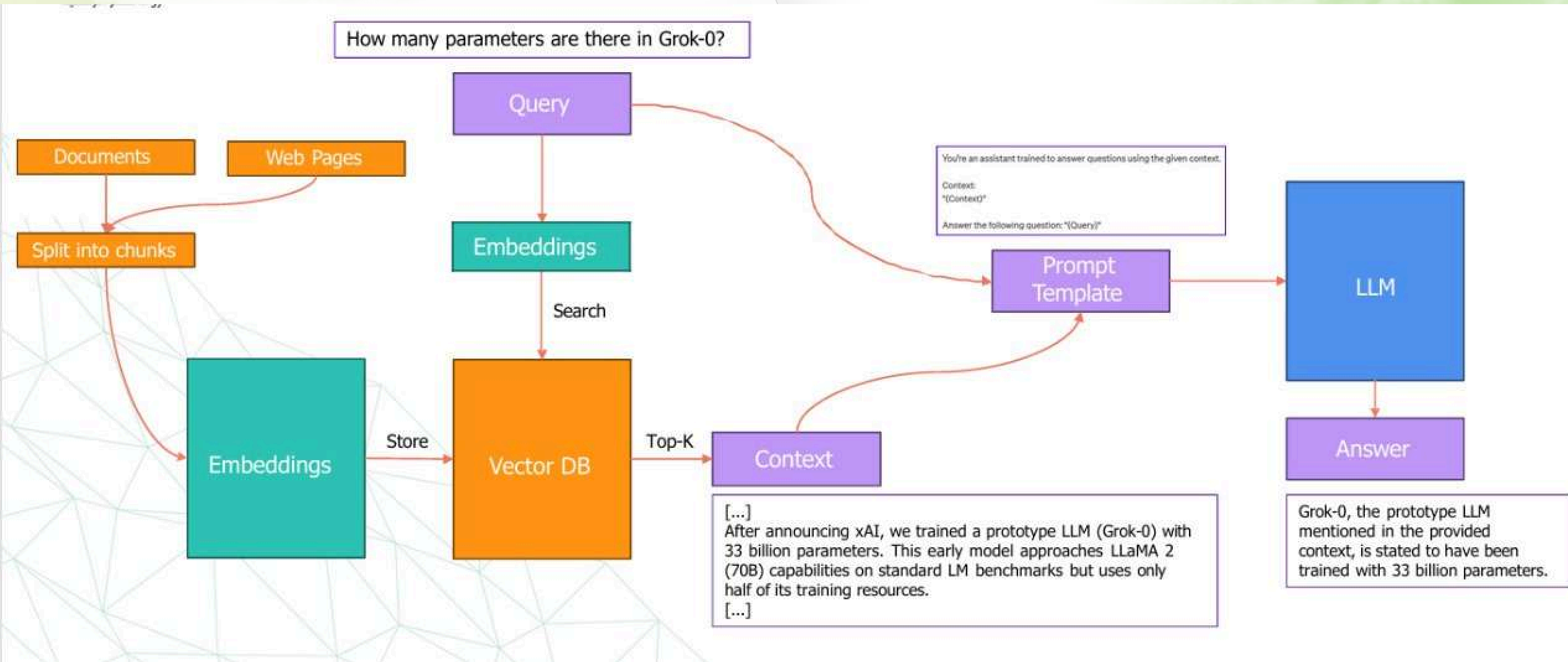
Most resources are in English → hard for Arabic-speaking farmers

No AI-based Arabic Q&A system exists for this domain.

Most online tools are in English, with no support for native Arabic agricultural content.



RAG - Overview



RAG - Implementation

Our application leverages the RAG architecture to provide precise answers. When a farmer asks a question in Arabic, the system:

1

Question Embedding

The user's query is converted into a numerical vector using a Sentence Transformer model.

2

Context Retrieval

FAISS, an efficient similarity search library, finds the most relevant plant disease information from our Arabic knowledge base.

3

Answer Generation

The Cohere command-r-plus model uses the retrieved context to generate a concise, accurate answer in Arabic.

Challenges

GPT-2

is trained on English-language data

AraGPT2

Fluent Arabic, but often inaccurate answers

AraBERT

Weak answer generation, hallucinated facts

Best Model: Cohere R7B

Excellent Arabic Support

Accurate and Grounded Answers

Fast and Lightweight



Command R7B

Chatbot Testing & Evaluation



Logic-Based Understanding



Arabic Dialect Recognition



Complex Question Handling

Deployment

Deployment makes a models usable in the real world by allowing external systems to get predictions.



What is Deployment?

- Deployment = Making the trained models available for real use.
- Allows external systems (like apps) to send data & get predictions.
- Bridges the gap between development and real- world usage.



Model Handover

Received the .h5 deep learning model from my teammate.

Chatbot is powered via Cohere LLM API and built using Python.

Prepare and deploy both models as APIs for the mobile team



Choosing a Deployment Tool

- Considered several frameworks (Flask, Django, etc.).

- Selected FastAPI – fast, lightweight, supports async.

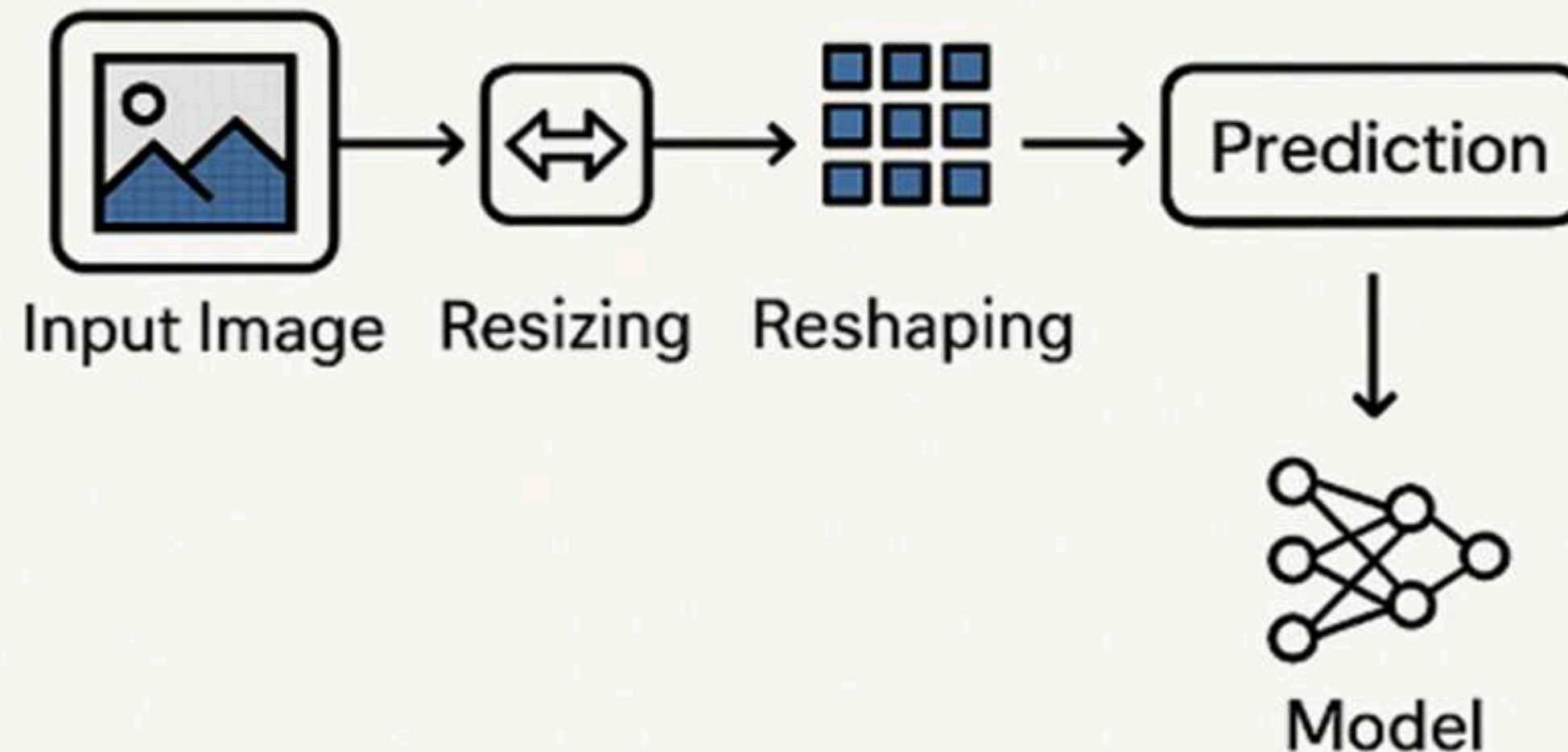
- Great for AI model inference.



Setting_Up environment

Choose FastAPI for deploying both models: CV model and AI chatbot.

Built endpoints: image classification `"/predict-image"`, chatbot interaction `"/ask"`



Testing Locally

- Needed to make it accessible to the Flutter team.



- Hosted the FastAPI app on localhost.

- Confirmed correct predictions.

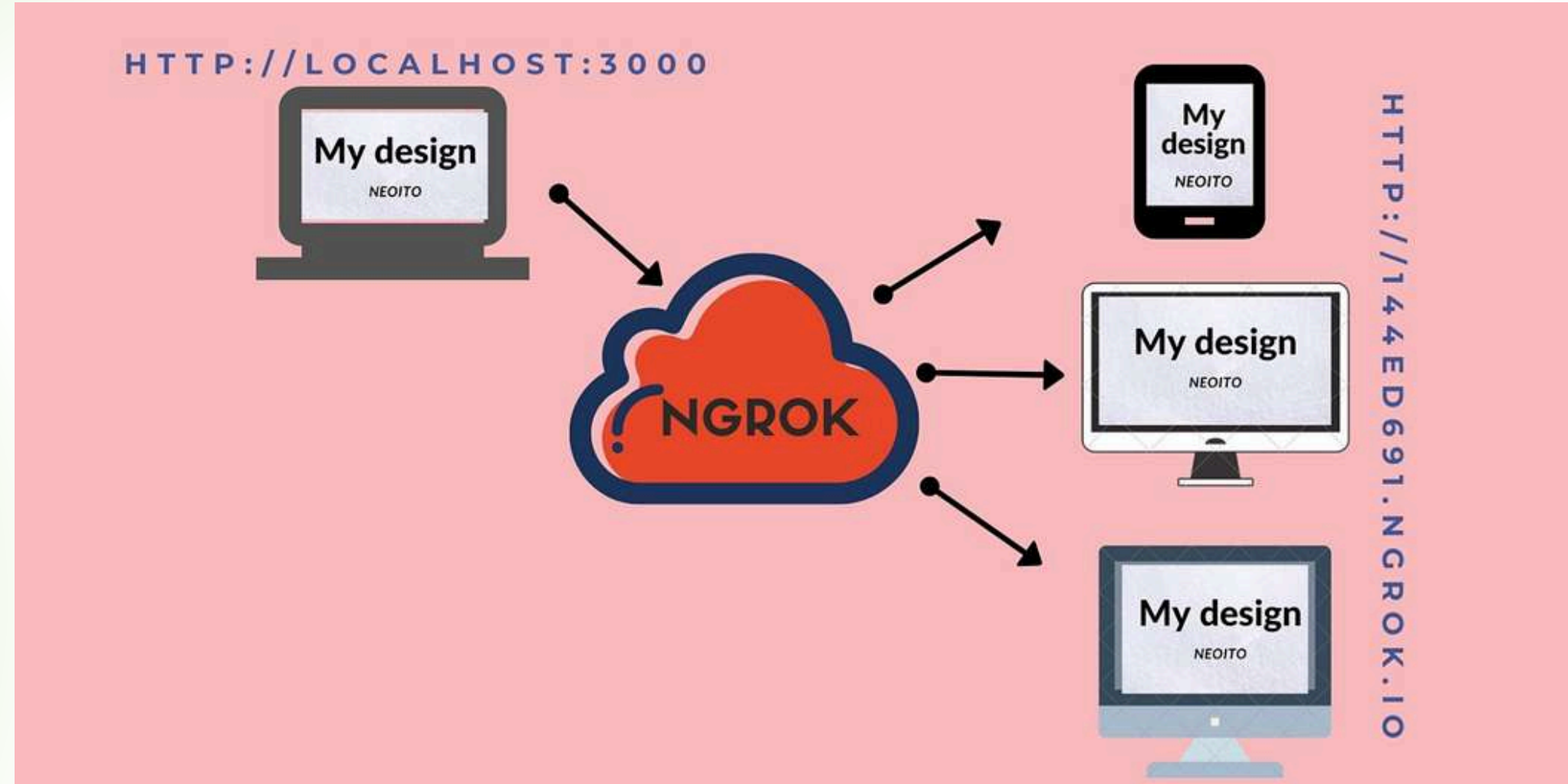


Using ngrok

- Used ngrok to generate a public URL for my local server.

- Shared the link with Flutter team for testing.

- Issues: Temporary URL + server offline if laptop is off.



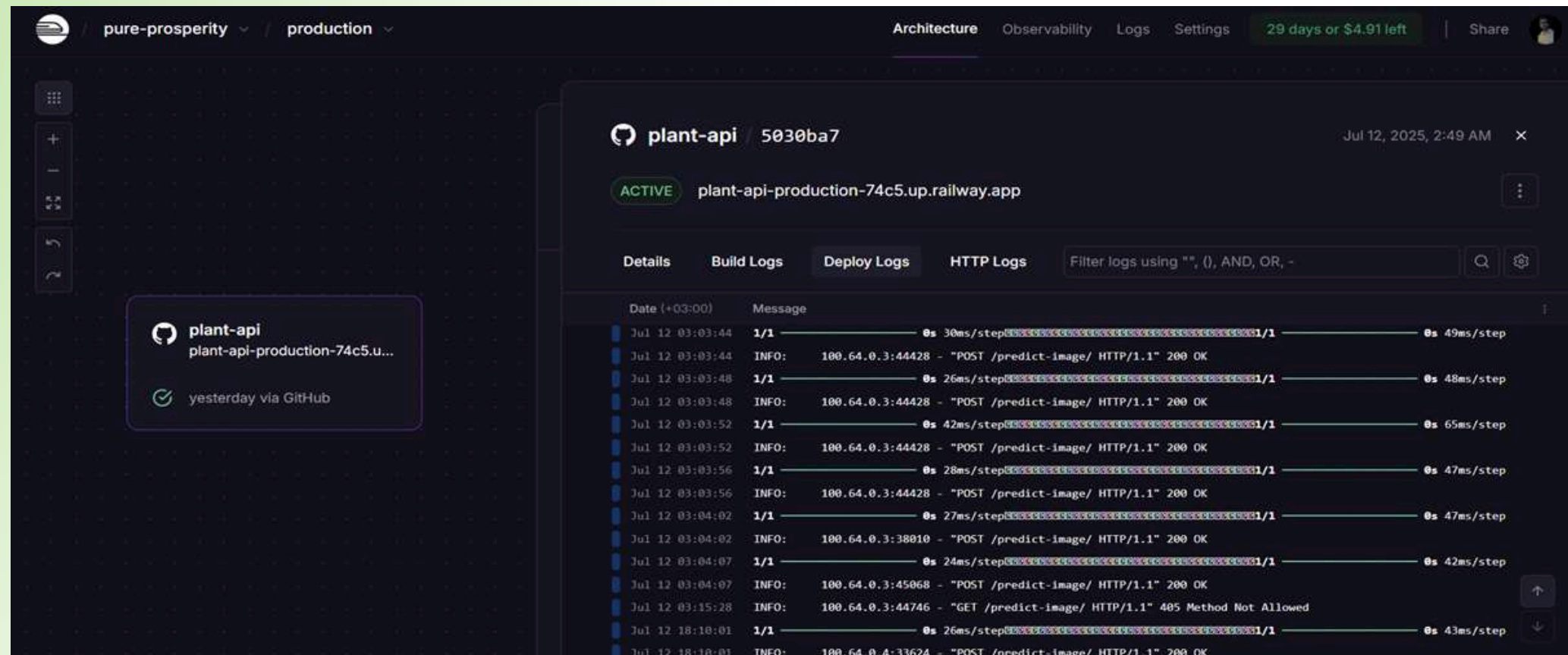
Permanent Hosting with Railway

- Explored hosting platforms – most were paid.

- Found Railway.app: free, easy deployment.

- Hosted the API and got a permanent public link.

- Limitations: RAM + limited free usage.





Smart Agricultural Guide

A mobile app built with Dart and Flutter for real-time plant monitoring, disease detection, and personalized advice.



Why Choose Our App?

⚡ High Performance : Fast and smooth experience using Dart & Flutter.

🎨 Modern Design : Simple, attractive, and easy to use.

👨🌾 User-Friendly : Designed for farmers & engineers—no tech skills needed.

🚜 Smart Farming Made Simple : Technology that makes farming easier and smarter.



Engaging OnBoarding Experience

1

Monitoring System



Your Smart Agriculture Companion

Discover smart solutions for your plants with personalized care, disease diagnosis, and weather insights – your ultimate agriculture companion!



2

Disease Detection



AI Plant Detection

Identify plants instantly using AI-powered detection – simply snap a photo or choose from your gallery to get accurate results!



3

Arabic AI Chatbot



AI Plant Chat Assistant

Connect with our AI Assistant for instant answers to all your plant care questions, anytime and anywhere!



Real-Time Plant Monitoring

12:08 PM

User Classification

12:09 PM

Select Your Plants

12:19 PM

Plant Care Test

Test notification sent successfully at 12:19 PM

Plant Care Advice

Expert tips for your plants

Keeps farmers informed with real-time alerts and tailored advice.

Select Your Role




Farm Owner



Agricultural Engineer


Next

Choose Your Plants




Radish

✓




Clover

✓




Okra

☐




Cotton

✓




Sugarcane



Fava Beans

Save Selected Plants



Tip

Radish

Get expert care advice for your plant

View Tips

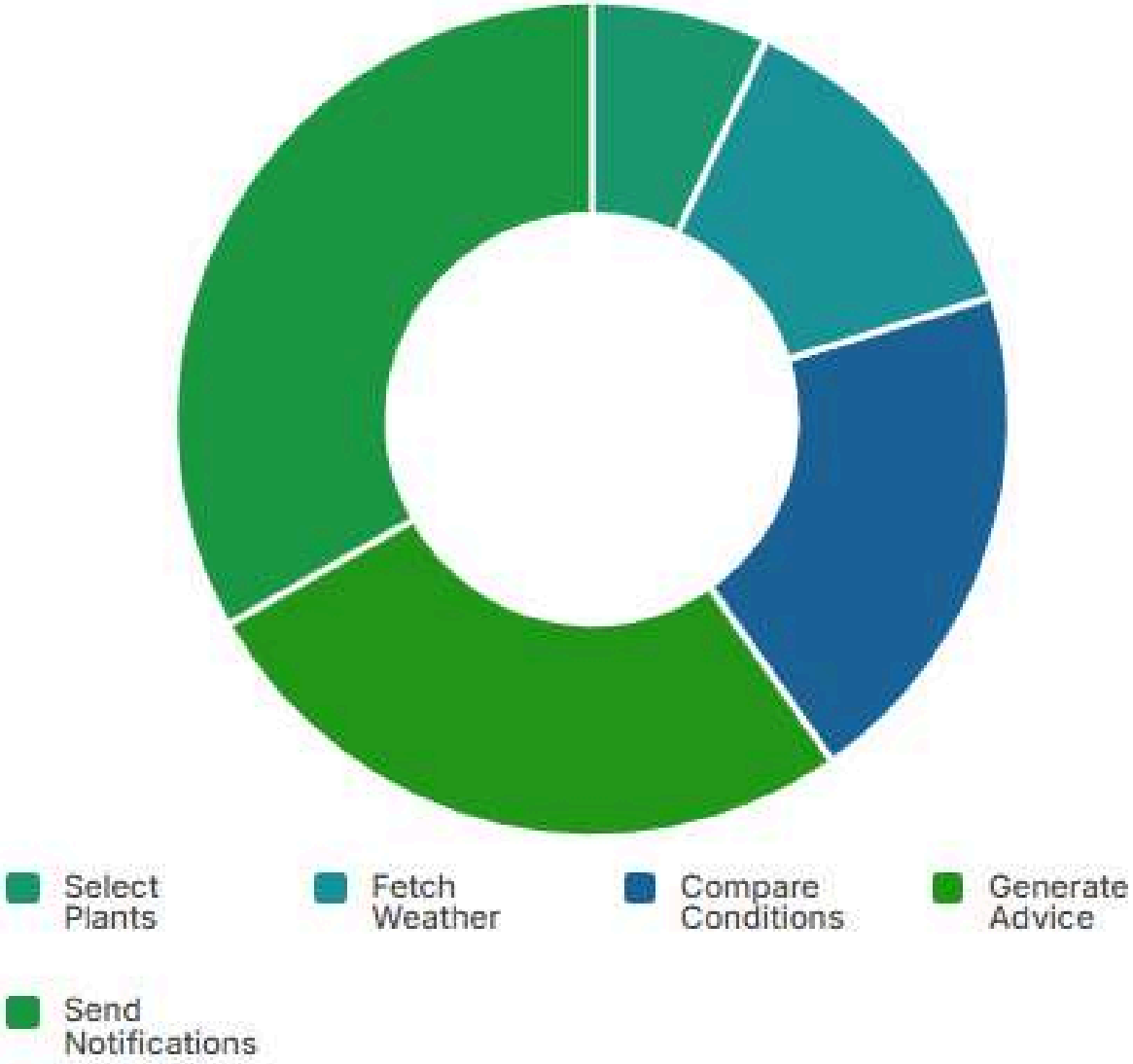
Test notification sent successfully!

Home

Messages

Plants

Advice



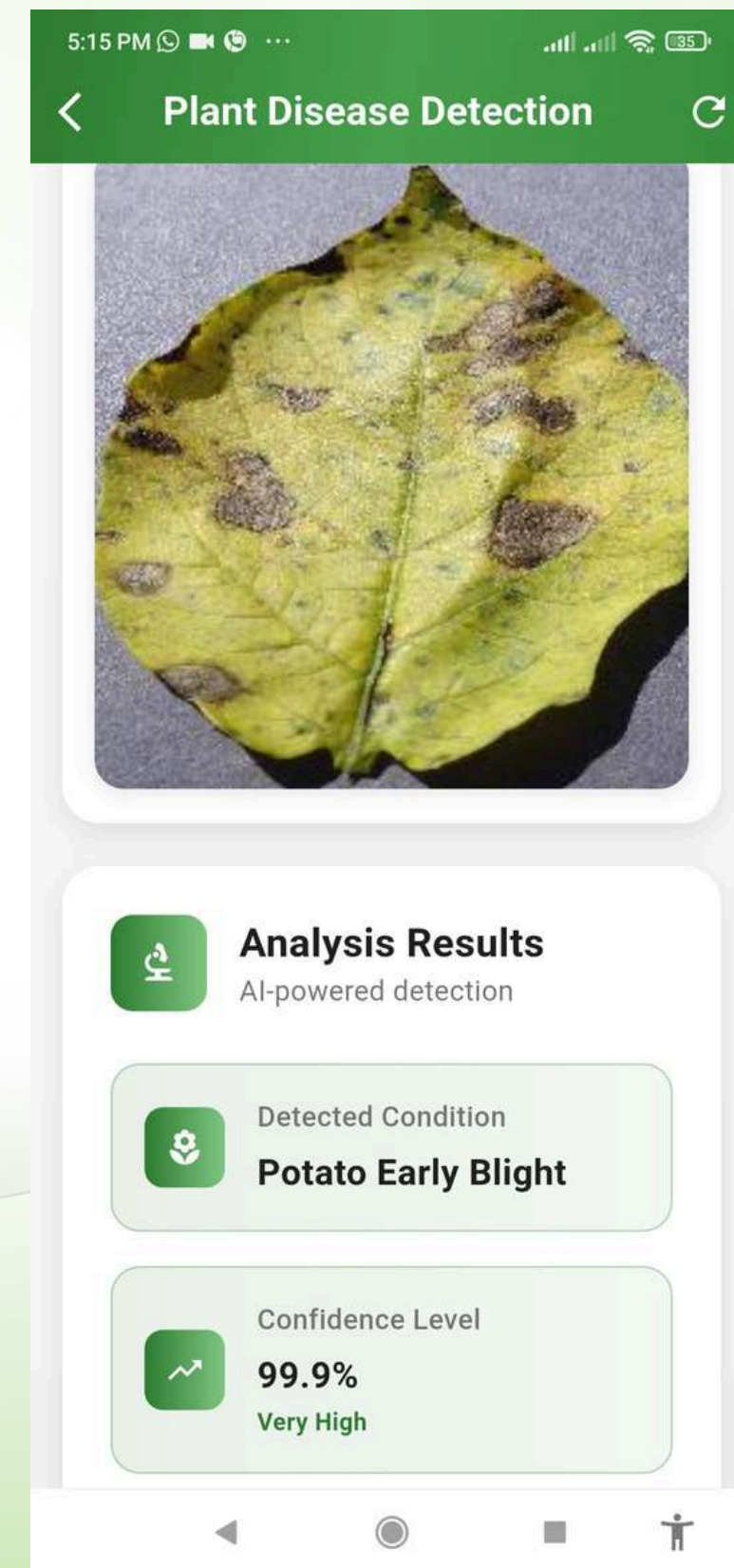
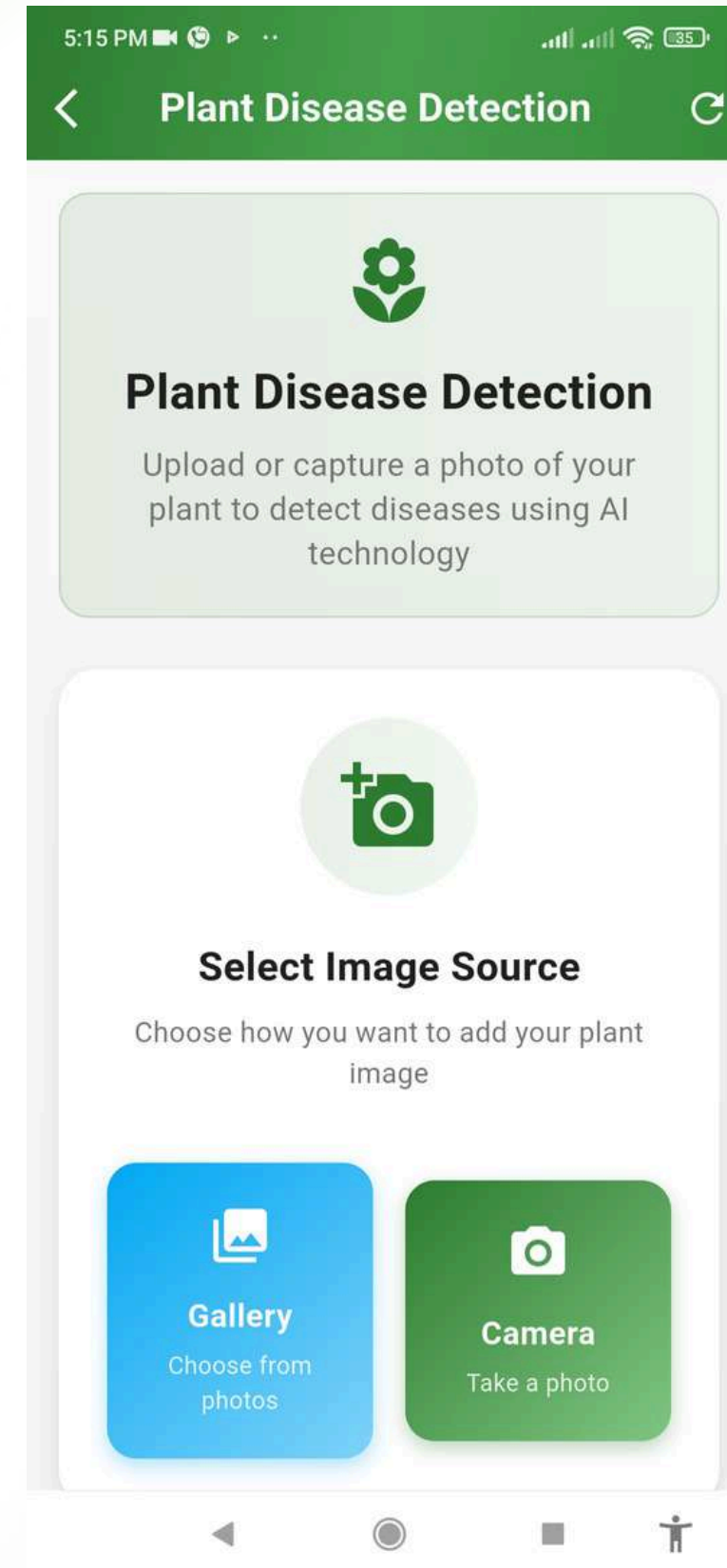
AI-Powered Disease Detection

Upload Image : Capture or upload a plant image.

Send to AI API : Image sent to AI model for fast diagnosis.

Receive Results : Get plant type, disease name, and treatment tips.

Display Recommendations : Save results locally for easy offline use.



Local Data & AI Chat Agricultural

Helping farmers with plant info they
can use offline and a smart AI chat.



Why Local Data?

Providing detailed plant data supports accurate agricultural decisions and better yields.

Farmers need access to plant info anytime even when offline.



Data Collection & Structure

Extensive plant data gathered from reliable sources.



Basic

quick facts (name, image, short info)



Details

in-depth info (watering, temperature,
soil, growth cycle)

Categories



Firestore Storage

Uploaded all plant data to Firestore Realtime Database.

Two data nodes

Basic

Details



Firestore
REALTIME DATABASE

Offline Caching with Hive

App fetches data every 24 hours from Firebase.

Data stored locally in Hive, enabling offline usage.

Ensures fast and reliable access for farmers in remote areas.



Search Feature

Separate search pages for each category.

Enables efficient and fast plant lookup.



Agricultural Chat – Text

Farmers and agronomists can ask plant-related questions.

Queries sent to AI-powered API, returning helpful text answers.



Adding Voice Support

Users can record voice questions.

Voice converted to text using speech-to-text, then sent to API.

Responses returned as text.



Future Development

- Scale the server to handle more queries and users.
- Launch the project as a real-world business targeting the agricultural sector.
- Enhance communication between farmers and agricultural engineers via the app.
- Integrate the system with IoT devices for real-time monitoring and intelligent responses

Thank You

