Leaf Vision: A Comprehensive Farmer AI Helper

Introduction

Leaf Vision is a lightweight web application that detects crop leaf diseases (corn, potato, rice, wheat) from photos using a fine-tuned Vision Transformer (ViT). After classification, the system provides farmer-friendly guidance (causes, prevention, treatment, risk factors) and integrates an AI chat assistant using API DeepSeek for contextual follow-up questions.

How It Works

- **Upload Image:** The farmers will upload image of a crop leaf diseases its either corn, potato, rice, and wheat.
- **AI Detection:** The image is analyzed using a fine-tuned Vision Transformer (ViT) model. The model predicts whether the leaf is healthy or diseased. For the detected disease, the app automatically provides Causes, Prevention, Treatment and Risk Factors.

Conversational AI Assistant:

After seeing the result, the farmer can ask follow-up questions in a chat-box and the AI will respond as the farm ask.

Development Challenges

At first, I struggled to develop this idea due to the limitations of my current hardware. I had difficulties training models, finding suitable datasets, and determining which models my laptop could handle for both image classification and the interactive AI chat. My second challenge was choosing an AI model for follow-up questions. I initially tried several APIs, but most were not working. I then experimented with downloaded models such as Google Flan and Phi-2. While both worked, they consumed a lot of RAM, responded slowly, and had limited knowledge.

How You Solve Them

I solved these challenges by finding a pre-trained model that did not require a dataset for training and testing which satisfied to my laptop specs that saved a lot of time. I also discovered a website called OpenRoute.ai, where I found a free AI service using an API key. Through this, I was able to use DeepSeek V3.1 as the solution for my web app now I can integrate these two models.

AI & Personal Contribution

I started building this project, I made sure to balance what the AI could do with my own hands-on work. The AI helped by suggesting patterns for setting up components, drafting some disease guidance, throwing in ideas for modern UI designs, giving quick hints when errors came up, and even suggesting outlines for documentation. But I didn’t just rely on it I took those drafts and refined them. I finalized the overall architecture, cleaned out unused code, and simplified the guidance so it made more sense for farmers.

For the design, I added my own responsive tweaks and custom class names, and for the code, I made it more defensive and tested it thoroughly. I also shaped the documentation to match the actual build process I followed. In the end, I managed to put together a working web app that combines image classification with an AI chat feature. It has a responsive UI that gives real-time, confidence-ranked predictions, a privacy-first design with no image storage, and clear, structured disease guides that farmers can actually use.