Plant Disease Detection System for Sustainable Agriculture

# Overview:

**This project involves building a machine learning-based system to identify plant diseases and recommend suitable crops and fertilizers. The system processes images of plant leaves to detect symptoms of diseases. It also takes into account environmental factors such as soil quality and weather conditions to optimize crop productivity.**

# Problem Statement:

**Farmers often struggle to detect plant diseases early, leading to crop loss and overuse of chemicals. There's a need for a quick and easy tool to identify diseases from leaf images to support healthy and sustainable farming.**

# Technologies:

**Python, OpenCV, TensorFlow/Keras, Machine Learning, Image Processing**

# Dataset:

We have a dataset named New Plant Diseases Data set

This data set in classified into 3 parts to create Plant Disease Detection System for Sustainable Agriculture model

1.Train dataset : this dataset is used to train the model.

2.Test dataset : this dataset is used to test our model.

3.Valid dataset: this dataset is used to evaluate the model.

# Pipeline:

A **pipeline** is a step-by-step process or workflow that describes how data flows through your project — from the beginning to the end.

## Data Collection & Data Loading:

In this step, we gather the data required to train and test the machine learning model. The dataset is categorized into train, test, and validation sets. Each set contains images belonging to different categories (e.g., category1, category2). These datasets are loaded and prepared for training.

## ZIP & Mounting:

The dataset is zipped and uploaded to Google Drive. It is then mounted in Google Colab for further use. Python code is used to unzip the dataset into the Colab environment.

# Image Processing & Augmentation

The images are processed to a consistent size (e.g., 128x128 pixels). Augmentation techniques may be applied to improve model generalization.

# CNN Model:

A CNN is used to train the model using the training dataset. The processed images are fed into the CNN, which learns to classify them

# Testing & Evaluation:

The model is tested using the test dataset. Evaluation metrics help determine the model’s performance.