**Project Pipeline Document**

**Title**: Plant Disease Detection System for Sustainable Agriculture

**1. Data Collection & Organization**

* A labeled image dataset of plant leaves is collected.
* The dataset is structured into:
  + Train
  + Test
  + Valid

Each folder contains subfolders for different disease categories (e.g., Category1, Category2).

**2. Dataset Preparation**

* The complete dataset is zipped and uploaded to **Google Drive**.
* Using **Google Colab**, the drive is mounted via Python code.
* The dataset is unzipped and made accessible for training and processing.

**3. Image Preprocessing**

* All images are resized to a uniform shape (e.g., **128×128** or **100×100**) regardless of their original resolution (like 400×400).
* Preprocessing steps include:
  + Normalization
  + Resizing
  + Noise reduction (optional)

**4. Image Augmentation**

* To enhance training robustness, augmentation techniques are applied:
  + Rotation
  + Flipping (horizontal/vertical)
  + Zoom
  + Brightness/contrast changes  
    This helps simulate real-world variability and prevents overfitting.

**5. Model Building (CNN)**

* A **Convolutional Neural Network (CNN)** model is built using Keras.
* Architecture includes:
  + Input layer (accepts 128×128×3 images)
  + Multiple Conv2D + Max Pooling layers
  + Dropout layers for regularization
  + Dense fully connected layers
  + Softmax output layer for multi-class classification

**6. Model Training**

* The model is trained using the training set.
* The validation set is used to monitor and tune the model performance.
* Parameters used:
  + Optimizer: Adam
  + Loss: Categorical Cross entropy
  + Batch Size: e.g., 32
  + Epochs: e.g., 20–50

**7. Model Evaluation**

* After training, the model is evaluated on the test set.
* Metrics include:
  + Accuracy
  + Precision, Recall, F1 Score
  + Confusion matrix for class-wise performance
* Visualization of results (e.g., accuracy/loss curves)

**8. Prediction & Recommendation**

* User uploads a new leaf image via interface (e.g., web or Colab UI).
* The image is pre-processed and passed to the trained CNN model.
* The model predicts the disease category.
* Based on the output, the system recommends:
  + Treatment options
  + Suitable fertilizer/crop rotation tips
* This is handled using **Python logic** in the backend.