**✅ Phase-5: Integration & Development**

**Objective:**Code the project and integrate components**.**

**Key Points:**

1. **Technology Stack Used:**
   * Frontend: HTML
   * Backend: Python
   * AI Model: pre-trained CNN, Keras/TensorFlow
   * Libraries: NumPy, OpenCV, Pillow, TensorFlow
2. **Development Process:**
   * Dataset downloaded from Kaggle (Recyclable/Non-Recyclable waste).
   * Data preprocessing: resizing, normalization, label encoding.
   * Model trained on Google Colab using transfer learning (VGG16).
   * Flask backend created for inference API.
   * Frontend developed with upload form, prediction result display, and single-page app navigation.
   * AJAX used to connect frontend to /predict\_api.
3. **Challenges & Fixes:**
   * Challenge: Class imbalance in dataset  
     Fix: Used data augmentation and class weighting during training.
   * Challenge: Real-time image preview was not updating  
     Fix: Implemented FileReader preview in JS with error fallback.
   * Challenge: Inconsistent predictions due to poor lighting in test images  
     Fix: Added normalization and brightness-contrast adjustment during preprocessing.