**Full Stack Development with Flask**

# Introduction

**Project Title:** CleanTech: Transforming Waste Management with Transfer Learning **Team Members:** ROUTHU CHARAN SAI

Team ID : LTVIP2025TMID44727

# Project Overview

**Purpose:**

To automate waste classification using a deep learning model to improve urban waste management and recycling efficiency.

**Features:**

* Image classification (Biodegradable, Recyclable, Trash)
* Transfer learning using VGG16
* Flask-based web interface
* Real-time prediction with uploaded images

# Architecture

**Frontend:**

* HTML templates (index.html, result.html)
* Simple UI to upload image and display prediction **Backend:**
* Flask server (app.py)
* Handles file upload, model inference, and routing **Database:**
* Not applicable (no persistent database used)
* Images stored temporarily in static/uploads/

# Setup Instructions

**Prerequisites:**

* Python 3.x
* TensorFlow
* Flask
* Matplotlib, NumPy, Pillow, etc.

**Installation:**

pip install tensorflow flask numpy matplotlib

npm install

**To run the application:**

npm run dev

# Folder Structure

**Client:**

* templates/index.html – File upload form
* templates/result.html – Displays prediction and image **Server:**
* app.py – Flask backend script
* vgg16.h5 – Trained deep learning model
* static/uploads/ – Stores uploaded files temporarily

# Running the Application

**Commands:**

# From project root cd w\_flask python3 app.py

Then open your browser and go to http://localhost:5173/

# API Documentation

**Endpoint:** /predict

**Method:** POST

**Description:** Accepts uploaded image file and returns the prediction **Request Example:**

* Form field: file
* Content-Type: multipart/form-data **Response Example:**

Prediction: Recyclable

# Authentication

No authentication used in this prototype version. All users can upload images freely. In future versions, user authentication can be implemented using Flask-Login or OAuth.

# User Interface

Simple and responsive HTML UI:

* Upload field for image selection
* Submit button for prediction
* Output section for result and uploaded image display

# Testing

**Manual Testing:**

* Uploaded 10 images from each category
* Verified correct prediction and display **Tools:**
* Jupyter Notebook for model training
* Flask Debug console for runtime logs

# Screenshots or Demo

* Home page with upload option
* Result page showing predicted label and uploaded image

*(Screenshots should be embedded in actual report file)*

# Known Issues

* No persistent storage of results
* UI lacks styling or responsiveness on mobile
* Only three waste categories supported

# Future Enhancements

* Add user login and history of predictions
* Connect to a real database
* Extend classification to more waste types
* Integrate with IoT-enabled smart bins

**Appendix**

**Source Code:** Included in the w\_flask/ folder

**Dataset:**[Dataset](https://www.kaggle.com/datasets/elinachen717/municipal-solid-waste-dataset)

**GitHub Repo:** https://github.com/charansai2004/Waste-management-system.git

**Demo Video:** https://drive.google.com/file/d/1J2R11MnqxUWeHtB1OWZ0PeGD-cx99RrY/view?usp=drive\_link