Full Stack Development Final Documentation

---

1. Introduction

Project Title: Waste Classification Using Transfer Learning\ **Team Members:** Praveen Kumar Reddy (Developer, ML Engineer)

---

2. Project Overview

Purpose: This project leverages transfer learning to assist users in waste classification. It categorizes uploaded images into Biodegradable, Recyclable, or Trash and offers recommendations for disposal.

Features:

- Drag-and-drop image upload
- Real-time waste prediction
- Confidence score display
- Category-specific recycling tips
- Responsive TailwindCSS UI

---

3. Architecture

Frontend: HTML5, TailwindCSS. Upload interface, responsive layout, integrated Flask routes.\ **Backend:** Flask app routes using Python; prediction logic powered by TensorFlow.\ **Database:** No traditional DB used. Model weights stored in `vgg16.h5`; user data not persisted.

---

4. Setup Instructions

Prerequisites:

- Python 3.10+
- Anaconda
- Flask, TensorFlow, Keras, Pillow, OpenCV

Installation:

```
conda create -n waste_classification python=3.10
conda activate waste_classification
pip install -r requirements.txt
```

Ensure dataset is extracted into the `data/raw/` folder.

5. Folder Structure

```
W_FLASK/
├── app.py
├── vgg16.h5
├── data/
│   ├── raw/
│   ├── output_dataset/
│   └── ...
├── static/
├── templates/
├── notebooks/
```

- **templates/**: Jinja HTML templates (index, predict, blog, etc.)
- **static/**: Image uploads, Tailwind assets
- **notebooks/**: Model training, data exploration

6. Running the Application

```
conda activate waste_classification
cd waste_classification_project/w_flask
python app.py
```

App runs locally at `http://127.0.0.1:2222`

7. API Documentation

- `POST /predict` → HTML form upload
- `POST /api/predict` → JSON output for API use

Each returns:

```
{
  "success": true,
  "prediction": { ... },
  "recycling_info": { ... }
}
```

8. Authentication

Not applicable — open access. Can be extended with JWT or OAuth.

---

9. User Interface

TailwindCSS-based layout with:

- Homepage (Get Started CTA)
- Upload section (drag-and-drop)
- Result display with recycling suggestions
- About & Contact pages

---

10. Testing

- Manual testing via image uploads (PNG, JPEG)
- Boundary testing for corrupted/unsupported files
- Verified model accuracy with validation split

---

11. Screenshots or Demo

Attached in submission folder. Includes:

- Homepage UI
- Upload and result page
- About/team layout

---

12. Known Issues

- Some image types (e.g., grayscale BMP) cause preprocessing errors
- Prediction accuracy dips for poor lighting/background

---

13. Future Enhancements

- Add more classes (e-waste, metal, glass)
- Deploy on AWS or GCP with authentication
- Real-time camera support
- Add user feedback loop to improve classification
