Full Stack Development Documentation

Project Title: Clean Tech

Team ID: LTVIP2025TMID34101

Team Members: [Add team member names and roles here]

1. Introduction

Clean Tech is an AI-powered waste classification system designed to support smart city and industrial waste segregation efforts.

1. Project Overview

Purpose:

To automate waste classification using transfer learning and image processing, improving accuracy, speed, and safety in waste management.

Features:

* Smart waste classification from images
* Dashboard for tracking
* Camera-based real-time detection
* Factory/public bin integration

1. Architecture

Frontend:

Built using React.js to display real-time waste classification data, graphs, and bin status.

Backend:

Node.js with Express.js serves APIs for handling classification results, user auth, and reporting.

Database:

MongoDB stores classification results, timestamps, and user data.

1. Setup Instructions

Prerequisites:

* Node.js
* MongoDB
* Git
* Python (for ML model server)

Installation:

1. Clone the repository
2. Run `npm install` in client and server folders
3. Set up `.env` files for keys and DB URIs
4. Folder Structure

Client:

/client/src/components – UI Components

/client/src/pages – React Pages

/client/src/api – Axios calls

Server:

/server/routes – Express routes

/server/models – Mongoose schemas

/server/controllers – Business logic

1. Running the Application

Frontend: `npm start` in `/client`

Backend: `npm start` in `/server`

Model Server: Run Python script serving the model (Flask or FastAPI)

1. API Documentation

POST `/api/classify` – Receives image data, returns waste category

GET `/api/stats` – Returns classification summary

POST `/api/login` – Authenticates user

1. Authentication

JWT-based authentication. Login API returns a token stored in localStorage for protected routes.

1. User Interface

Includes dashboard view, live feed viewer, and result logs. Responsive design with basic theming.

1. Testing

Used Jest and Postman for unit and API testing. Accuracy validated using benchmark image datasets.

1. Screenshots or Demo

To be included: dashboard screenshot, live classification, output logs.

1. Known Issues

* Misclassification under poor lighting
* Edge device limitations on real-time speed

13. Future Enhancements

* Integration with physical waste sorting hardware
* Smart alerts for bin overflow
* More granular waste category support