FOOD WASTE MANAGEMENT

NAME – ANSHIKA SINGH

(20231CSE0190)

EMAIL – anshika.tech.outlook.com

DATE – 17 JULY 2025

TABLE OF CONTENTS:

1. PROBLEM STATEMENT

2.PROJECT OVERVIEW

3. DATASET DESCRIPTION

4. FEATURED IMPLEMENTATION

5. TECHNICAL ARCHITECTURE

6.CODE WALKTHROUGH

7.SCREENSHOT OF OUTPUT

1. PROBLEM STATEMENT:

Develop a Flask-based web dashboard to analyze food waste management (from a CSV dataset) and provide actionable insights, including:

* Identifying best/worst months for food wastage trends in restaurants,
* Detecting inactive high-rise waste of food trends and measures to prevent and donate,
* Visualizing trends via monthly reports and time-slot pie charts.
* Data analysis of monthly ,weekly and daily trends of wastage.

1. PROJECT OVERVIEW:

Food wastage is a growing concern in events, restaurants, and catering industries. This project aims to build a **web-based analytical dashboard** to study food wastage trends using historical data. By identifying patterns, common issues, and peak wastage times, organizations can reduce food loss, improve efficiency, and promote sustainability.

## **Objectives**

* Build a responsive **Flask web dashboard**
* Load and analyze food wastage data from a **CSV file**
* Visualize wastage trends **monthly and weekly**
* Identify:
  + ✅ **Best Month** (least food wastage)
  + ❌ **Worst Month** (most food wastage)
* Detect common causes of food waste (e.g. storage conditions)
* Provide **actionable insights and preventive suggestions.**

3. DATASET DESCRIPTION

This dataset captures various **factors affecting food wastage** during events or large-scale food preparation settings (like weddings, corporate events, etc.). It includes attributes like food type, guest count, storage methods, seasonality, and actual food waste recorded. Columns are given below of data.csv

1. Type of Food
2. Number of Guests
3. Event Type
4. Quantity of Food
5. Storage Conditions
6. Purchase History
7. Seasonality
8. Preparation Method
9. Geographical Location
10. Pricing
11. Wastage Food Amount
12. FEATURED IMPLEMENTATION

The **Food Wastage Analysis Dashboard** is a Flask-based web application designed to help users analyze and reduce food waste using CSV data. It processes key information such as food type, guest count, storage conditions, and waste quantity. By simulating dates, the system enables monthly and weekly trend analysis, which is visualized through interactive charts using Chart.js. The dashboard highlights the best and worst months in terms of food wastage, identifies common issues like improper storage, and offers practical suggestions to prevent further loss. With a clean HTML interface and modular Python backend, the application provides actionable insights in a user-friendly and offline-ready format.

5. TECHNICAL ARCHITECTURE

* 📥 **CSV Integration**: Loads food wastage data from a CSV file using pandas.
* 📅 **Date Simulation**: Adds a simulated Date column to enable time-based analysis.
* 📊 **Monthly & Weekly Analysis**: Aggregates wastage by month and week for trend insights.
* 🏆 **Best & Worst Month Detection**: Identifies the month with the least and most food waste.
* ⚠️ **Problem Detection**: Finds the most common issue (e.g., storage conditions) causing wastage.
* 💡 **Preventive Suggestions**: Provides actionable tips based on detected issues.
* 📈 **Data Visualization**: Displays interactive charts (line and bar) using Chart.js.
* 🔁 **Dynamic Dashboard**: All data is processed live on each load — no need for a database.

6. CODE WALKTHROUGH:

### **app.py – The Backend (Python)**

1. **Import Tools**  
    Uses Flask for web app, Pandas for data, and NumPy for random dates.
2. **Read CSV File**  
    Loads your food\_wastage\_data.csv file into Python.
3. **Group Data**
   1. Groups wastage by **month** and **week**.
   2. Finds which month has **least** (best) and **most** (worst) waste.
4. **Find Common Problem**
   1. Checks which **storage condition** (e.g. room temperature) causes most waste.
   2. Gives a simple **suggestion** based on the problem.
5. **Send to Website**
6. **Run the App**
   1. Starts the Flask web server so you can view the dashboard in your browser

7. SCREENSHOT OF OUTPUT:

