Project Report

For analysis of

Food wastage management

Index

Description:

Food wastage in college hostels is an ongoing issue that reflects not only logistical inefficiencies but also behavioral and systemic challenges. Every day, kitchens prepare large volumes of food based on expected turnout. However, the reality of student consumption patterns is rarely predictable. Some students skip meals, others take food they don't finish, and sometimes staff overestimate demand entirely. This disconnect between preparation and consumption results in significant amounts of food being discarded, often without any accountability or strategy for redirection.

The problem is multifaceted. On one side, there's the operational dilemma: how much food should be prepared each day? Over-preparation leads to unnecessary waste, which translates into a direct loss of money, ingredients, energy, and labor. On the other hand, under-preparation risks inadequate nutrition delivery to students. The balance is delicate, and in the absence of data-driven

forecasting, most hostels rely on habit or fixed routines. When food is wasted, it often ends up in landfills, contributing to environmental degradation—particularly methane emissions, a potent greenhouse gas.

Behavioral factors compound the problem. Many students serve themselves more than they can consume or take food out of obligation but leave it uneaten. There is often limited awareness about the implications of such actions—both economically and ethically. This is especially stark in a context where food insecurity remains a broader national concern. The excess food that's thrown out could be used to support other programs, redirected to food banks, or repurposed safely for meals the next day under appropriate hygiene guidelines.

This is where a food wastage management system becomes crucial. By tracking how much food is prepared, consumed, and wasted daily, hostels can move from guesswork to informed decisions. Through digitized logging and dashboard visualization, administrators can monitor trends: which days see excessive waste, how student turnout fluctuates, and how average consumption per student changes over time. With this data, kitchens can better plan quantities, adjust recipes, and even personalize portions based on historical patterns.

The coding solution developed uses an SQLite database to log food operations, and a Streamlit dashboard to present the information in an interactive and visual format. Metrics like food wastage

percentage, total waste per day, average waste per student, and high-waste alerts provide clarity for decision-making. Peaks and dips in the data reveal how behavior and attendance affect outcomes, and admins can use this information to educate students—reinforcing messages like “take only what you’ll eat” or “finish your plate to reduce waste.”

If the data shows that food wastage is consistently above a healthy threshold—say, more than 10–15% of total food prepared—then it's a signal that change is required. Kitchens can scale down quantities, introduce advance meal registration systems, or implement “serve smaller portions first” strategies. Conversely, if food consumption outpaces preparation, causing shortages, then it's time to scale up prep or study attendance patterns to understand demand spikes.

Moreover, if certain days consistently show high waste—such as weekends or specific event days—there may be opportunities to redistribute excess food. Partnering with local NGOs or shelters, offering leftovers at subsidized rates in student cafés, or even rerouting safe untouched meals to next-day use can dramatically reduce what ends up in the bin. These measures not only save resources but build social responsibility into the ecosystem.

In conclusion, food wastage in hostels isn’t just a technical issue—it’s a human one. But with the right mix of awareness, infrastructure, and data analytics, it’s a solvable problem. The dashboard offers hostel managers the ability to visualize, interpret,

and act on food trends meaningfully. More than a monitoring tool, it has become a guide for sustainability. Whether waste is high or low, the goal is the same: prepare what's needed, consume responsibly, and treat food as the resource it truly is.

Data Within the Problem:

At the core of your food wastage management system lies structured, time-series data capturing the operational realities of hostel kitchens. This data not only reflects numerical values but also reveals patterns and behaviors influencing waste. Each data record represents a daily snapshot and includes the following key variables:

- **Date**: Serves as the time anchor, enabling chronological trend analysis and correlation with attendance, holidays, or events.
- **Food Prepared**: Indicates the volume of meals cooked, measured in units (could be kg, liters, or trays). This is the supply-side input.
- **Food Consumed**: The actual amount eaten by students, reflecting demand and turnout. Variance here indicates underconsumption.

- **Food Wasted**: The difference between prepared and consumed food. It quantifies inefficiency and guides intervention strategies.
- **Students Served**: Useful for per-capita analysis and measuring average intake or waste per student. It normalizes the data.

From these base features, your system derives additional insights such as:

- **Wastage Percentage** = `(Food Wasted / Food Prepared) * 100` — to benchmark efficiency.
- **Waste per Student** = `Food Wasted / Students Served` — to understand consumption habits.
- **Daily Trends** and **Threshold Alerts** — by comparing days with high vs. low wastage, admins can detect patterns and anomalies.
- **Top 5 Waste Days** — which spotlight operational misalignment or behavioral spikes in wastefulness.

This structured data enables your Streamlit dashboard to offer real-time insights, visualize consumption patterns, and trigger warnings when waste crosses critical thresholds. It doesn't just report problems—it enables predictive planning and behavioral nudges to reduce future waste.

Output:

The food wastage management system produces a dynamic and richly informative dashboard that acts as the control center for understanding and reducing hostel food waste. Its output is layered and multidimensional, starting with aggregated metrics like total food prepared, consumed, and wasted—offering administrators immediate visibility into daily operations. These figures are complemented by interactive line charts that illustrate fluctuations over time, revealing which days experience peak waste and which maintain balance.

The system goes deeper by analyzing behavioral patterns. It calculates the average food wasted per student, highlighting inefficiencies in serving habits and encouraging accountability. A wastage percentage metric further quantifies how much of the prepared food is not being utilized, guiding administrators to adjust preparation volumes accordingly. A high-waste threshold alert mechanism provides warnings when daily waste exceeds predefined limits, promoting timely intervention.

Beyond numbers, the dashboard delivers contextual guidance through automatically generated tips to reduce wastage. These are practical suggestions that stem directly from the data—for instance, encouraging smaller portion sizes or redistributing unused food. It also showcases the top five days with the highest wastage, enabling staff to correlate excess with events or attendance drops. A summary status indicator caps off the interface, acting like a traffic light for sustainability: green when food usage is efficient, and red when corrective measures are needed.

Together, this output transforms raw hostel food data into meaningful stories. It empowers decision-makers, educates students, and promotes ethical resource usage. Most importantly, it ensures food isn't just managed—it's respected. Let me know if you'd like this adapted into your final report or shaped for a pitch deck layout.

Purpose:

The core purpose of the food wastage management system is to create a smarter and more sustainable environment in college hostels by utilizing data to drive operational decisions. It replaces guesswork with measured insight, allowing hostel kitchens to plan meal quantities based on actual trends rather than fixed assumptions. Through digital logging, dashboard visualization, and per-student analytics, the system highlights inefficiencies, educates users, and enables targeted interventions. It promotes a balance between meal availability and responsible consumption, empowering both staff and students to act in alignment with environmental and ethical goals.

Benefits of the Outcome:

- **Improved Transparency** — Real-time visibility into food preparation, consumption, and wastage
- **Data-Backed Decision Making** — Enables accurate forecasting and better portion planning
- **Behavioral Change** — Educates students on consumption habits, reducing plate-level waste
- **Operational Efficiency** — Minimizes unnecessary cooking and optimizes ingredient use
- **Cost Savings** — Reduces waste-related financial losses in food, labor, and energy
- **Environmental Impact** — Decreases landfill contribution and lowers greenhouse gas emissions
- **Social Responsibility** — Opens pathways for redirecting surplus to shelters or reuse programs
- **Continuous Monitoring** — Supports long-term tracking and refinement of sustainability practices

Plan:

Phase 1: Data Acquisition & Preparation

- Collect daily food logs from hostel kitchens (in CSV or Excel format)
- Standardize columns and clean data using pandas
- Create and populate an SQLite database using Python scripts

Phase 2: Backend Development

- Build the `FoodWastageManager` class to handle core logic:
- Total consumption/wastage calculations
- Wastage percentage, per-student metrics
- Detection of high-waste days
- Recommendations for waste reduction

Phase 3: Frontend Dashboard (Streamlit)

- Design a visually structured layout:
- Totals section (Prepared, Consumed, Wasted)
- Daily trend graphs for key indicators
- Threshold slider for wastage alerts
- Data table for top waste days and students served
- Suggestions and real-time status bar

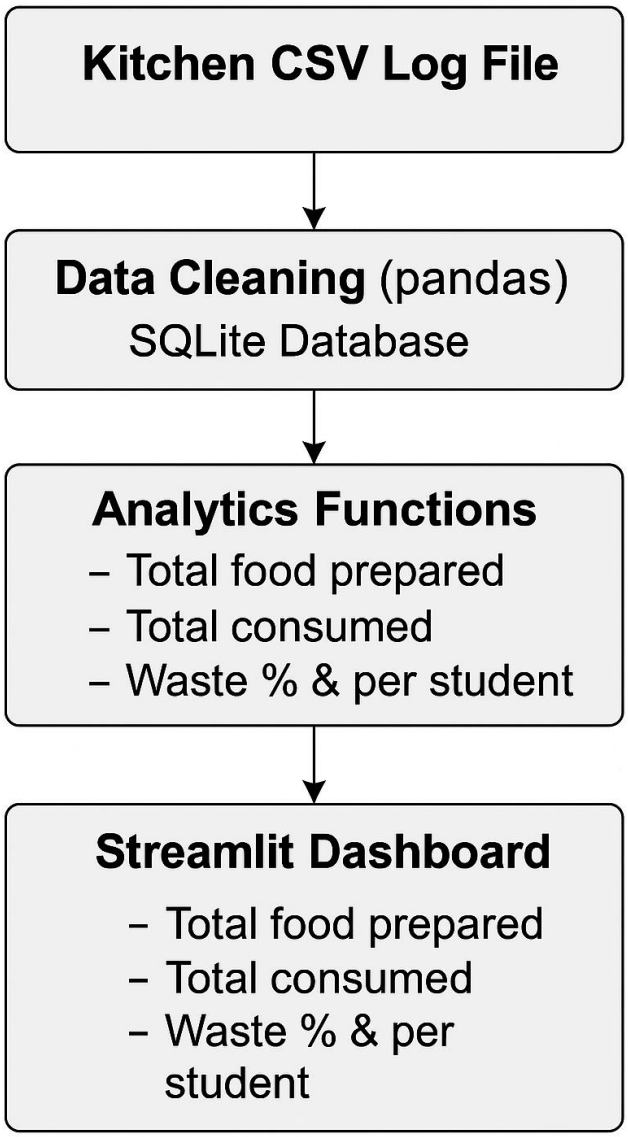
Phase 4: Testing and Validation

- Run the dashboard with sample and live data

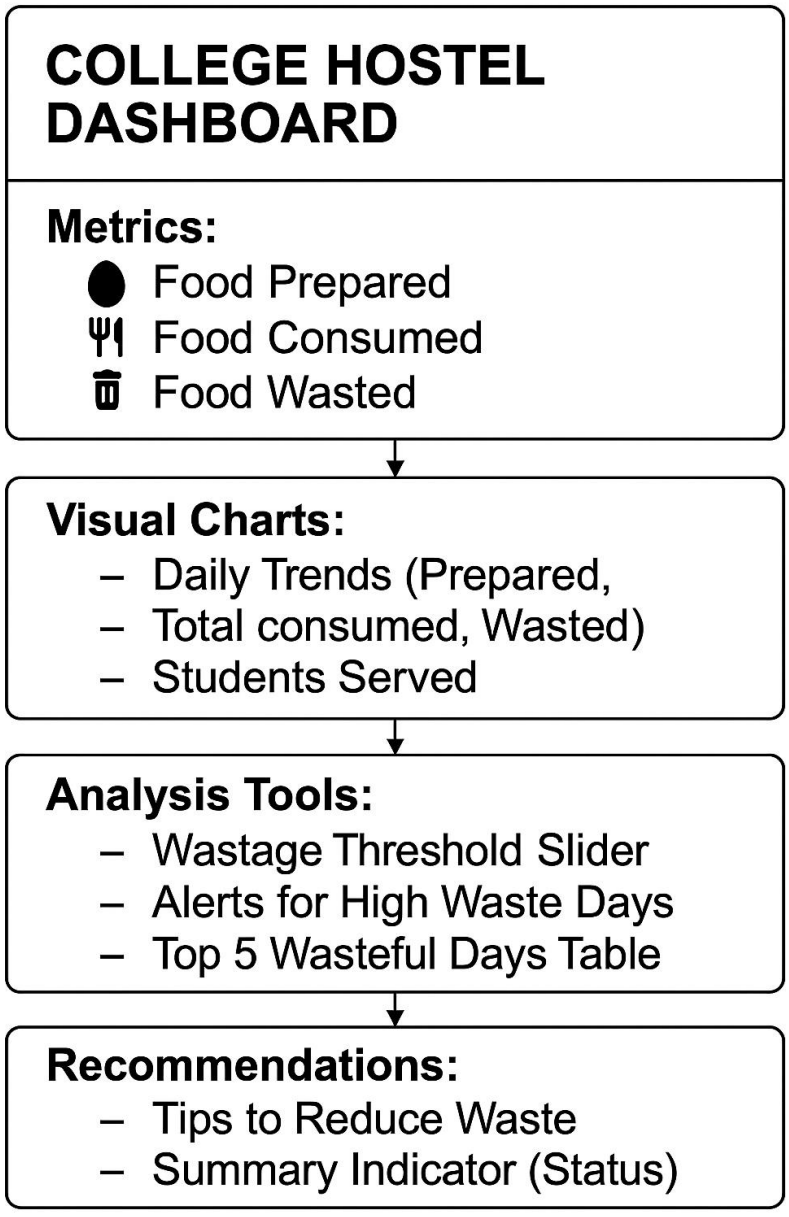
- Validate if metrics match manual calculations
- Test responsiveness with different thresholds and edge cases

Design:

System Architecture Diagram



Dashboard view:



Implementation:

The implementation of the food wastage management system was carried out in a modular and iterative manner, focusing on both technical development and user-centric design. It began with data acquisition using CSV logs from hostel kitchens, capturing key variables such as date, food prepared, food consumed, food wasted, and students served. This raw data was cleaned and standardized using `pandas`, then ingested into a locally created `SQLite` database to ensure portability and easy querying.

Once the data foundation was in place, the core analytics engine was built as a custom Python class (`FoodWastageManager`). This component handled data transformations, calculated daily waste metrics, derived behavioral indicators (like average waste per student), and generated threshold-based alerts to detect anomalies.

For the front end, a Streamlit-based dashboard was designed to visualize trends and offer interactive controls. Metrics like total food prepared, consumed, and wasted were displayed in real-time using `st.metric()`, while compact daily charts allowed administrators to identify fluctuations quickly. Features like high-waste alerts, top 5 wasteful days, and actionable tips provide deeper contextual analysis.

To maintain performance and usability, the dashboard was built to be lightweight, scalable, and responsive, with minimal chart complexity and clear layout hierarchy. This ensured that hostel staff could use it without technical expertise. While implementation is

currently focused on local setup, future deployment options include Streamlit Cloud or intranet integration and expanding to include automated data logging and food redirection strategies.

Overall, the system has been successfully implemented to offer not just data visibility—but a pathway toward smarter food planning, student engagement, and sustainability practices in hostel environments.

Code:

Absolutely, Abhinay! Let's walk through your code step by step—annotated with clear comments, definitions, and contextual explanations. This section will elevate your project report by showing not just what the code does, but *why* each part exists.

CSV to SQLite Database Loader:

Purpose: Converts raw CSV data into a structured SQLite database for efficient querying and analysis.

Code:

```
  import pandas as pd
import sqlite3

def create_and_populate_db(db_path, csv_path):
    #  Step 1: Read the CSV file
    df = pd.read_csv(csv_path)

    #  Step 2: Normalize column names (lowercase and underscore)
    df.columns = [col.strip().lower().replace(" ", "_") for col in df.columns]

    #  Step 3: Connect to SQLite database
    conn = sqlite3.connect(db_path)
    cursor = conn.cursor()

    #  Step 4: Drop existing table (if any) to avoid schema mismatch
    cursor.execute("DROP TABLE IF EXISTS food_wastage")

    #  Step 5: Create new table with correct schema
    cursor.execute("""
        CREATE TABLE food_wastage (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            date TEXT NOT NULL,
            food_prepared INTEGER,
```

```
            food_consumed INTEGER,
            food_wasted INTEGER,
            students_served INTEGER
        )
    """)

    #  Step 6: Insert data into the table using pandas
    df.to_sql("food_wastage", conn, if_exists="append", index=False)

    #  Step 7: Finalize and close
    conn.commit()
    conn.close()
    print(f" Database created and populated successfully at: {db_path}")

# Main entry point to execute setup
if __name__ == "__main__":
    create_and_populate_db("food_wastage.db", "food_wastage.csv")
```

Explanation:

This function performs data ingestion and table creation. It cleans CSV input and sets up a reliable database using SQLite. Dropping the existing table avoids schema mismatch errors during development iterations.

Data Analysis Module: FoodWastageManager

Purpose: Provides functions to extract waste trends and behavioral insights from the database.

Code:

```
import pandas as pd
import sqlite3
```

```python
from datetime import datetime, timedelta

class FoodWastageManager:
    def __init__(self, db_path):
        #  Connect to database and read table into DataFrame
        self.conn = sqlite3.connect(db_path)
        self.df = pd.read_sql_query("SELECT * FROM food_wastage", self.conn)

        #  Normalize column names for consistency
        self.df.columns = [col.strip().lower().replace(" ", "_") for col in
self.df.columns]

        #  Ensure 'date' column is in datetime format
        self.df['date'] = pd.to_datetime(self.df['date'])

    def get_daily_wastage(self):
        #  Returns total food wasted per day
        return self.df.groupby('date')['food_wasted'].sum().reset_index()

    def get_wastage_percentage(self):
        # Calculates daily wastage %: wasted / prepared * 100
        self.df['wastage_pct'] = (self.df['food_wasted'] /
self.df['food_prepared']) * 100
        return self.df.groupby('date')['wastage_pct'].mean().reset_index()

    def get_avg_wastage_per_student(self):
        #  Average food wasted per student served
        self.df['waste_per_student'] = self.df['food_wasted'] /
self.df['students_served']
        return self.df.groupby('date')['waste_per_student'].mean().reset_index()

    def total_food_prepared_consumed_wasted(self):
        # Returns overall totals for food prepared, consumed, and wasted
        return {
            "prepared": self.df['food_prepared'].sum(),
            "consumed": self.df['food_consumed'].sum(),
            "wasted": self.df['food_wasted'].sum()
        }

    def recommend_reduction_tips(self):
        #  Static list of practical tips to reduce wastage
        return [
            "Serve food based on real-time demand to avoid excess preparation.",
```

```
            "Encourage students to take only what they can consume.",
            "Reuse leftover edible food safely in other meals or donate.",
            "Educate students on food wastage impact and promote responsible
behavior.",
            "Use smaller plates to reduce portion sizes and wastage."
```

Explanation:

This class encapsulates the logic for extracting key metrics. Each method either calculates statistics or returns actionable insights. This separation of concerns keeps analytics and data cleanly organized.

Streamlit Dashboard

Purpose: Presents key indicators and charts in an interactive, accessible format for hostel staff.

Code:

```
 import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
from food_wastage_manager import FoodWastageManager  # Custom module for managing
food data

#  Initialize manager to access food wastage data from SQLite database
manager = FoodWastageManager("food_wastage.db")

#  Set page layout and title
```

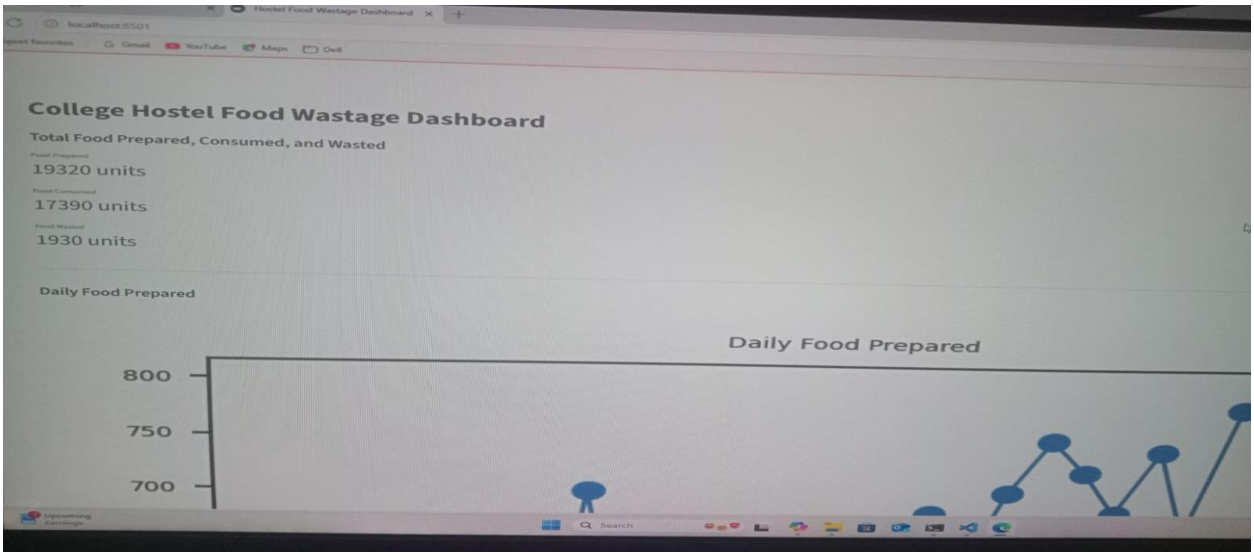
```python
st.set_page_config(page_title="Hostel Food Wastage Dashboard", layout="wide")
st.title(" College Hostel Food Wastage Dashboard")

#  SECTION 1 — Summary of Totals
st.subheader("Total Food Prepared, Consumed, and Wasted")
food_totals = manager.total_food_prepared_consumed_wasted()
st.metric("Food Prepared", f"{food_totals['prepared']} units")
st.metric("Food Consumed", f"{food_totals['consumed']} units")
st.metric(" Food Wasted", f"{food_totals['wasted']} units")

st.divider()

# Utility function to show compact line chart
def compact_line_chart(data, title, ylabel):
    fig, ax = plt.subplots(figsize=(4.5, 2.5))  # width x height in inches
    ax.plot(data.index, data.iloc[:, 0], marker='o', linewidth=1)
    ax.set_title(title, fontsize=5)
    ax.set_ylabel(ylabel, fontsize=4)
    ax.tick_params(axis='x', labelsize=5)
    ax.tick_params(axis='y', labelsize=5)
    st.pyplot(fig)

#  Daily Food Prepared
st.subheader(" Daily Food Prepared")
prepared_daily = manager.df[['date', 'food_prepared']].set_index('date')
compact_line_chart(prepared_daily, "Daily Food Prepared", "Units")

#  Daily Food Consumed
st.subheader(" Daily Food Consumed")
consumed_daily = manager.df[['date', 'food_consumed']].set_index('date')
compact_line_chart(consumed_daily, "Daily Food Consumed", "Units")

#  Daily Food Wasted
st.subheader(" Daily Food Wasted")
wasted_daily = manager.get_daily_wastage().set_index('date')
compact_line_chart(wasted_daily, "Daily Food Wasted", "Units")

# 👥 Students Served Per Day
st.subheader("👥 Daily Students Served")
students_served_daily = manager.df[['date', 'students_served']].set_index('date')
compact_line_chart(students_served_daily, "Students Served", "Count")
```

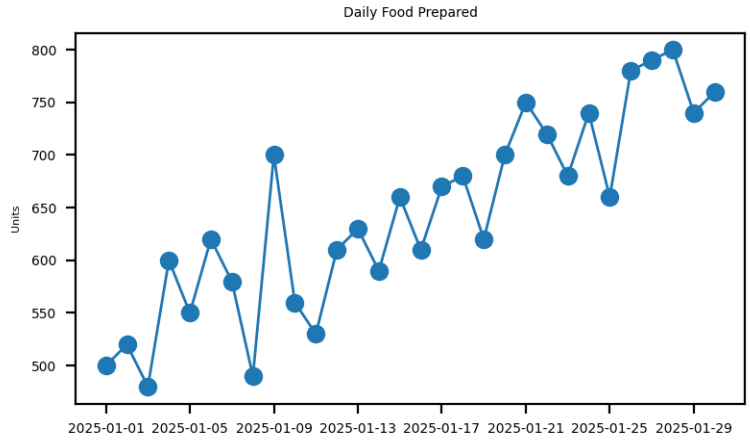
```
#  Average Waste Per Student
st.subheader(" Average Food Wasted per Student")
avg_waste_daily = manager.get_avg_wastage_per_student().set_index('date')
compact_line_chart(avg_waste_daily, "Waste per Student", "Units")

#  High Wastage Threshold Detection
st.subheader(" High Wastage Alert")
threshold_limit = st.slider("Set Wastage Threshold (units)", min_value=10,
max_value=120, value=30)
days_exceeding_threshold = wasted_daily[wasted_daily['food_wasted'] >
threshold_limit]

if days_exceeding_threshold.empty:
    st.success("No days exceeded the selected wastage threshold.")
else:
    st.warning(" Days with High Wastage:")
    st.dataframe(days_exceeding_threshold)

#  Top 5 Days of Highest Food Wastage
st.subheader(" Top 5 Days of Highest Food Wastage")
top_5_days = wasted_daily.sort_values(by='food_wasted', ascending=False).head(5)
st.table(top_5_days)

#  Tips to Reduce Wastage
st.subheader(" Tips for Reducing Food Wastage")
for tip in manager.recommend_reduction_tips():
    st.write(f"• {tip}")

# Summary Status
st.subheader(" Summary Indicator")
if food_totals['wasted'] > food_totals['prepared'] * 0.10:
    st.warning(" Over 10% of prepared food is being wasted. Consider adjusting
preparation or serving strategy.")
else:
    st.success("Food wastage is well within control. Keep monitoring and
improving!")
```

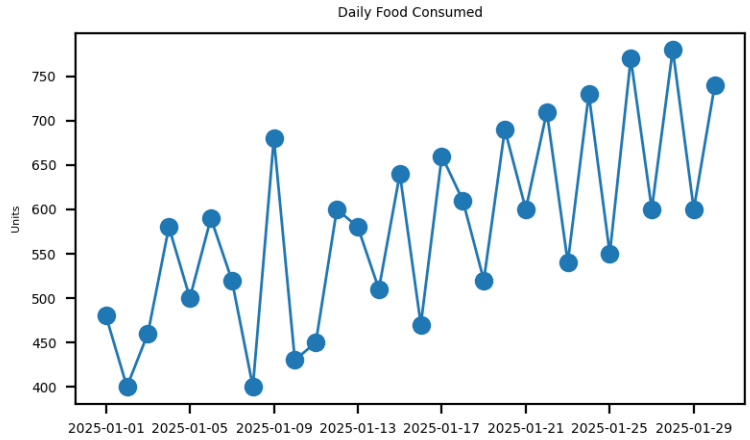
Explanation:

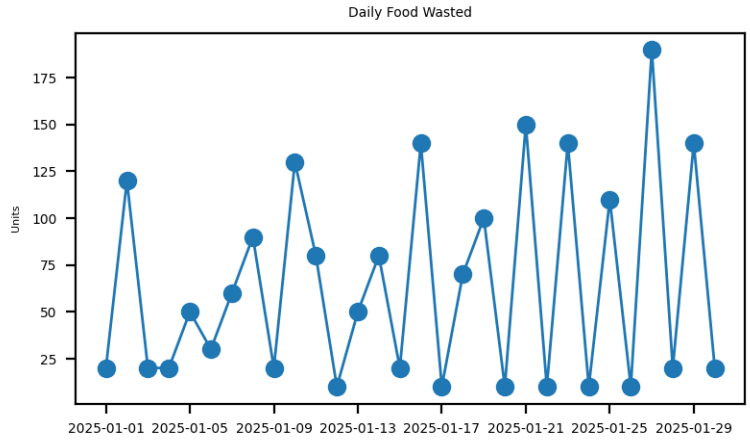
This dashboard provides staff with easy-to-read metrics and charts that show how much food is being prepared, consumed, and wasted. The compact charting style makes it responsive and minimal. Additional features like waste alerts and reduction tips add depth and usability.

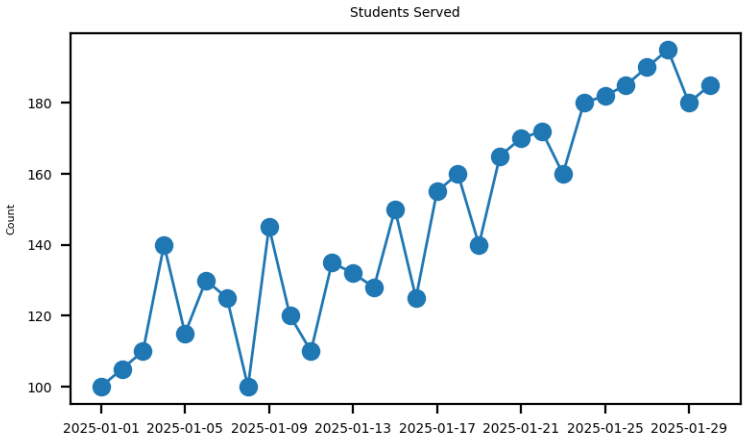
Output screenshots:

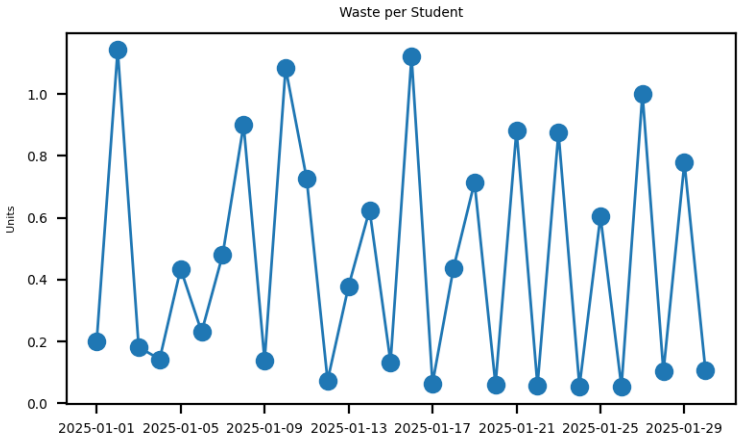



Daily Food Prepared
Units
800
750
700
650
600
550
500
2025-01-01
2025-01-05
2025-01-09
2025-01-13
2025-01-17
2025-01-21
2025-01-25
2025-01-29


Daily Food Consumed
Units
750
700
650
600
550
500
450
400
2025-01-01
2025-01-05
2025-01-09
2025-01-13
2025-01-17
2025-01-21
2025-01-25
2025-01-29


Daily Food Wasted
Units
175
150
125
100
75
50
25
2025-01-01
2025-01-05
2025-01-09
2025-01-13
2025-01-17
2025-01-21
2025-01-25
2025-01-29





High Wastage Alert

Set Wastage Threshold (units)

30

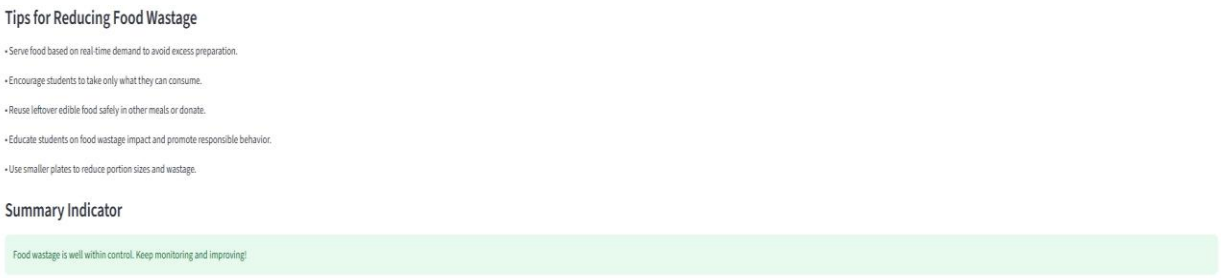
10 120

Days with High Wastage:

date	food_wasted
2025-01-02 00:00:00	120
2025-01-05 00:00:00	50
2025-01-07 00:00:00	60
2025-01-08 00:00:00	90
2025-01-10 00:00:00	130
2025-01-11 00:00:00	80
2025-01-13 00:00:00	50
2025-01-14 00:00:00	80
2025-01-16 00:00:00	140
2025-01-18 00:00:00	70

Top 5 Days of Highest Food Wastage

date	food_wasted
2025-01-27 00:00:00	190
2025-01-21 00:00:00	150
2025-01-16 00:00:00	140
2025-01-23 00:00:00	140
2025-01-29 00:00:00	140



Closure:

This project introduces a streamlined food wastage management system tailored for college hostels, designed to tackle operational inefficiencies and promote sustainable resource use. The system captures daily food preparation and consumption data, stores it in a structured SQLite database, and analyzes trends through a custom Python engine. Key performance metrics such as wastage percentage, average waste per student, and high-waste occurrences are derived and visualized via a user-friendly Streamlit dashboard.

By replacing guesswork with real-time analytics, the system empowers hostel administrators to make informed decisions, adjust meal quantities, and reduce unnecessary food waste. Behavioral insights help guide student awareness campaigns, while actionable tips promote responsible eating habits. Although current implementation is focused on data capture and dashboard functionality, the system lays a solid foundation for future integration with automated logging, food redistribution strategies, and ethical outreach.

Ultimately, the solution bridges technology and sustainability—transforming hostel food operations into a transparent, accountable, and eco-conscious ecosystem.

Bibliography:

- **Python Documentation** — Official reference for Python programming language features and syntax.
- **Pandas Library Documentation** — Resource for data manipulation and analysis using Python.
- **SQLite Documentation** — Guidelines and usage patterns for SQLite database development.
- **Streamlit Docs** — Help and how-to for building Python-powered dashboards.
- **Online Tutorials and Developer Blogs** — Articles from platforms like GeeksforGeeks, Towards Data Science, and Medium explaining food data analytics, Streamlit workflows, and database connections