SMART FOOD STORAGE SYSTEMS

A Web-Based System for Locating Nearest Food Storage Containers.

**Authors:**

* Abhinav Sinha (23BCE1718)
* Abishek KG (23BCE1739)
* Naren Medarametla (23BCE1721)

**Institution:**Vellore Institute of Technology , Chennai

**Date of Submission:** 18th April,2025

**Problem Statement:** Inefficient access to food as well as inefficient ways to dispose of excess food which leads to food waste and logistical challenges during emergencies.

**Objective:** Develop a website to help users locate the nearest food storage facilities in real time.

**Methodology:** HTML with CSS and JavaScript along with Python Flask server , Rest API and Geolocation via Google Maps API and Openstreetmap API.

**Results:** A functional website which allows users to check for the closest food storage location for easy accessibility.

**Conclusion:** The website improves food accessibility, with future plans for IoT integration and disaster relief applications.

**INTRODUCTION**

* **Background:**
  + Food wastage costs the global economy $1 trillion/year, while 9% of the population lives in extreme poverty.
  + Storage inefficiencies lead to spoilage, especially in regions lacking cold-chain infrastructure.
  + This project will allow us to reduce one of the major problems troubling the citizens of our country.
* **Motivation:**
  + The excessive poverty and starvation present in our country.
  + Empower NGOs and governments to redistribute surplus food effectively.
* **Scope:**
  + **Included:** Real-time storage tracking, donor-NGO collaboration.
  + **Excluded:** Direct food delivery logistics (focus on storage-to-user navigation).
* **Objectives:**
  1. Reduce food wastage by 30% through optimized storage allocation.
  2. Improve access for low-income households by 50% within 1 year.
* **Expected Outcome:** A scalable tool to alleviate hunger, reduce waste, and promote equitable resource distribution.

**System Design & Architecture**

**Software Design**

* **Overview:** Users search by location; system returns nearest containers with directions.
* **Architecture:** Two-tier (frontend, backend).
* **Technologies:** HTML, Java Script , CSS , Python Flask and Google Maps API.
* **Use Case Diagram:** Includes user search.

**Hardware Requirements**

* Any basic mobile or devices which can access the internet can access the site..

**Implementation**

**Software Implementation**

* **Development Process**: Agile, Waterfall, V-Model, etc.
* **Module Description**: Description of each module and its function.
* **Algorithm/Pseudocode**: Core logic explanation

**Results and Testing**

* **Functional Testing:** 100% pass rate for search and registration.
* **Performance Testing:** Handles 1,000 concurrent users with <2s response time.

**Discussion**

* **Comparison:** Outperforms generic apps with specialized filters (e.g., container capacity).
* **Challenges:** Real-time data accuracy; solved via user validation.
* **Limitations:** Coverage depends on user-reported data.

**Conclusion & Future Scope**

* **Summary:** Successful deployment.
* **Applications:** Disaster relief, supermarket surplus redistribution.
* **Future Plans:** IoT sensors for inventory tracking, mobile app development.

**References**

* **Google Maps API :**

**Appendix**

* **Source Code:**
* **Screenshots:**