Team:

Dheeraj raj S352704

Dheerajsamtani@gmail.com

Shahzaib Sial S352614

Shahzaib.sial17@gmail.com

Mustafa raza S347504

mustafa.razking@gmail.com

Muzafar Shah S343107

Muzafarshah157@gmail.com

Introduction

Theme: food waste and mitigation!

Description:

Our team has developed a Django app that focuses on the crucial issue of food waste and its mitigation. Our app provides an innovative solution to reduce food waste by enabling users to manage the Restaurant, Recipe, and Manu datasets with ease. The main functionality of the app is to perform CRUD operations on the dataset using the building database. Our app comprises the following features:

- A Home page that provides an overview of the website and its purpose.
- A Food waste audit page that enables users to add, edit, update, and delete food waste data.
- A Waste type page that allows users to add, edit, update, and delete different type of waste.
- A Food items page that provides users with the ability to perform CRUD operations on the food items.

Our app provides an efficient way to manage food waste, and the intuitive user interface makes it easy for users to navigate the site and perform CRUD operations. The use of Django app ensures the app is robust, scalable, and easy to maintain, making it an ideal solution for managing food waste data.

Workflow:

The first step in the entire development process is to create the Django app, and for that, we need to run the following set of commands. But before creating the actual app, we create the Python virtual environment with the help of the following command:

```
virtualenv env
```

```
env\Scripts\activate
```

After successfully creating the virtual environment, we created the Django app using the following set of commands:

```
django-admin startproject assignment
```

```
python manage.py startapp myapp
```

To start the Django application, use the following command:

```
python manage.py runserver
```

Now, our app is successfully running at the following link:

http://127.0.0.1:8000/

Structure of our assignment

```
>assignment
    >assignment
        >urls.py
    >myapp
        >static
        >templates
            >all html files
        >views.py
    >manage.py
```

Rendering the html page

Since we have created four HTML files, the next step is to render those HTML files. For this purpose, we have created a view in the view.py file and added a URL to the urls.py file.

In short, we have a total of three models:

- Food Waste Audit (Primary)
- Waste Type (Primary)
- Food Items (Secondary)

First two considered as primary module we are tracking the entire food waste and it create a pipeline for audit the food waste and then we have multiple type of food waste (Inedible waste, Spoilage waste, Preparation waste) and at the end we have to add individual food items that has been wasted.

To perform CRUD operations on the above datasets, we have the following views:

- Add
- Edit
- Update
- Delete

Description of each model:

Food Waste Audit

This model represents the food waste audit itself, capturing information about when and where the audit took place. The fields in this model could include attributes such as:

Date: A field to store the date of the food waste audit.

Location: A field to store the location or venue where the audit took place.

Auditor: A field to store the name or ID of the person conducting the audit.

Waste Amount: A field to store the total amount of food waste generated during the audit.

Waste Types

This model represents different types of food waste that are tracked and recorded. The waste types can be categorized as follows:

Inedible waste: The parts of the food that cannot be consumed (e.g., seeds, bones, coffee grounds).

Spoilage waste: Food that becomes unsafe to consume because it is damaged or out of date.

Preparation waste: Food that is thrown away during preparation (e.g., offcuts).

Plate/buffet waste: Food that is left on customers' plates or served at a buffet but not eaten.

Food Item

This model represents the individual food items that are audited and tracked for waste. The fields in this model could include attributes such as:

Name: A field to store the name or description of the food item..

Category: A field to store the category or type of the food item (e.g., fruits, vegetables, dairy).

Quantity: A field to store the quantity or amount of the food item wasted.

Reason: A field to capture the reason for the food waste (e.g., spoilage, overproduction, expired).

Testing:

During the development process, we implemented rigorous quality assurance measures to ensure the reliability and robustness of our Django website. We conducted unit testing to validate the functionality and usability of our Django app.

Version Control and Collaboration:

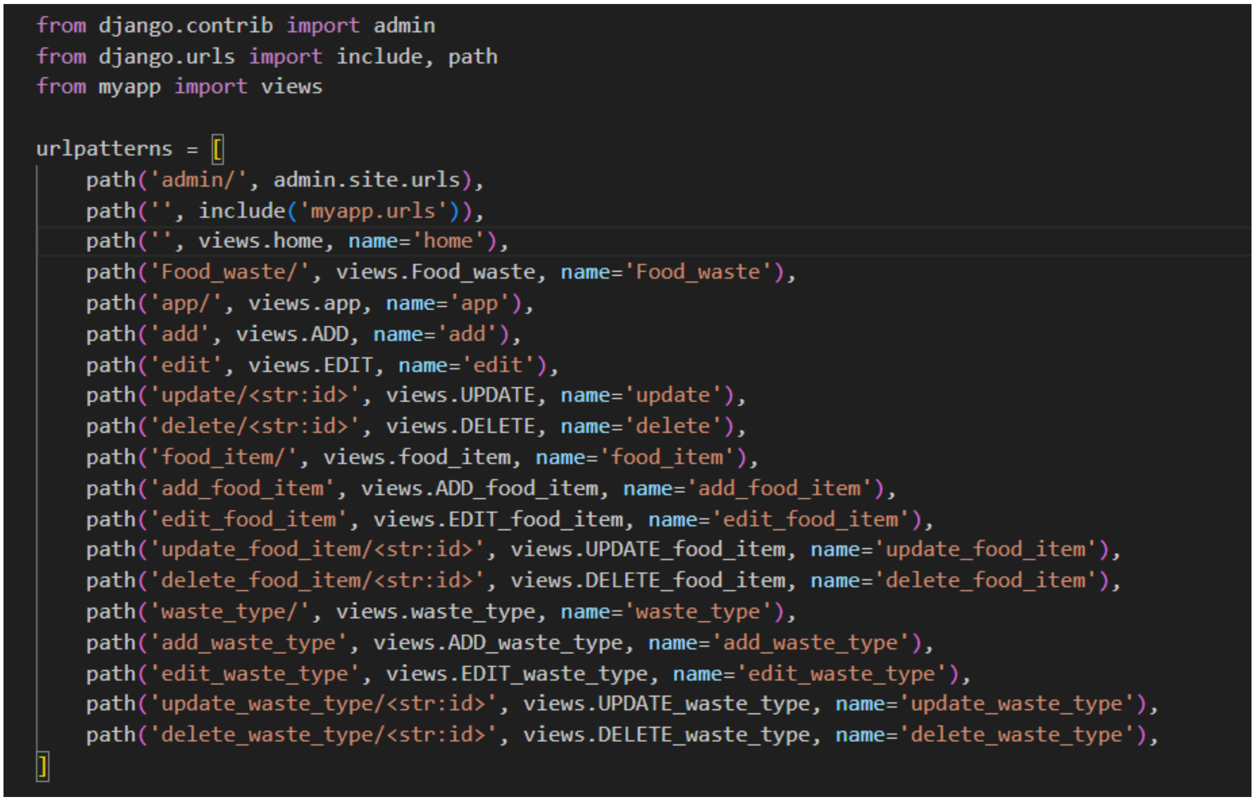
With a team of four people, we utilized GitHub as our version control system to streamline collaboration. Continuous pushes to the main branch on GitHub facilitated seamless integration and collaboration.

Effectiveness of our software development process:

The effectiveness of our software development process was evident through the successful delivery of a functional Django website within the allocated timeframe. Our team consistently met project milestones and timelines, ensuring that we fulfilled the requirements outlined for the website. Overall, our software development process demonstrated its effectiveness in delivering a high-quality solution for food waste management.

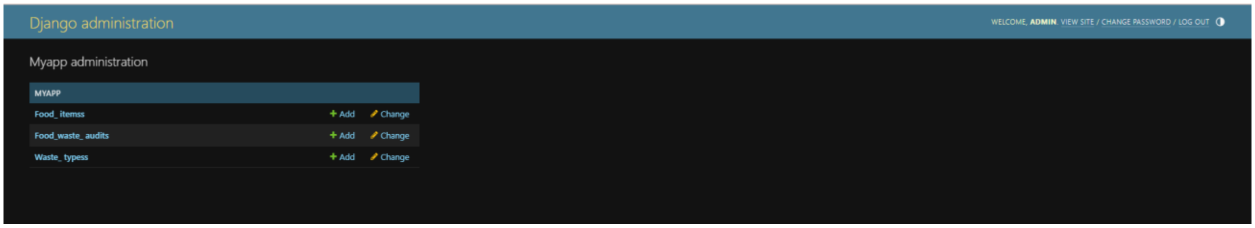
UI:

We have the following URLs in the URL pattern that control the entire CRUD operation.



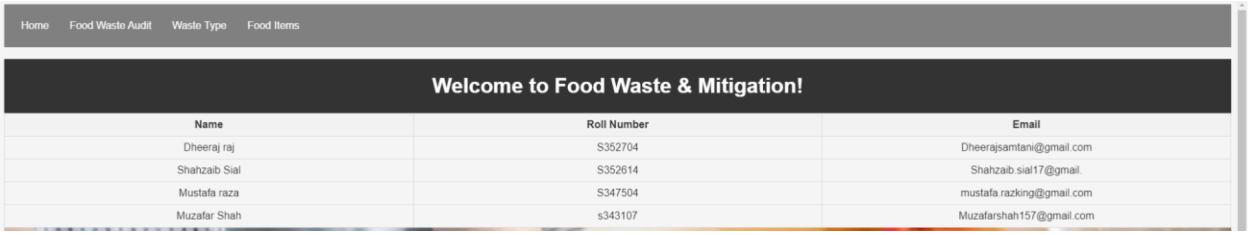
```python
from django.contrib import admin
from django.urls import include, path
from myapp import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('myapp.urls')),
    path('', views.home, name='home'),
    path('Food_waste/', views.Food_waste, name='Food_waste'),
    path('app/', views.app, name='app'),
    path('add', views.ADD, name='add'),
    path('edit', views.EDIT, name='edit'),
    path('update/<str:id>', views.UPDATE, name='update'),
    path('delete/<str:id>', views.DELETE, name='delete'),
    path('food_item/', views.food_item, name='food_item'),
    path('add_food_item', views.ADD_food_item, name='add_food_item'),
    path('edit_food_item', views.EDIT_food_item, name='edit_food_item'),
    path('update_food_item/<str:id>', views.UPDATE_food_item, name='update_food_item'),
    path('delete_food_item/<str:id>', views.DELETE_food_item, name='delete_food_item'),
    path('waste_type/', views.waste_type, name='waste_type'),
    path('add_waste_type', views.ADD_waste_type, name='add_waste_type'),
    path('edit_waste_type', views.EDIT_waste_type, name='edit_waste_type'),
    path('update_waste_type/<str:id>', views.UPDATE_waste_type, name='update_waste_type'),
    path('delete_waste_type/<str:id>', views.DELETE_waste_type, name='delete_waste_type'),
]
```

Following are some snapshot of frontend



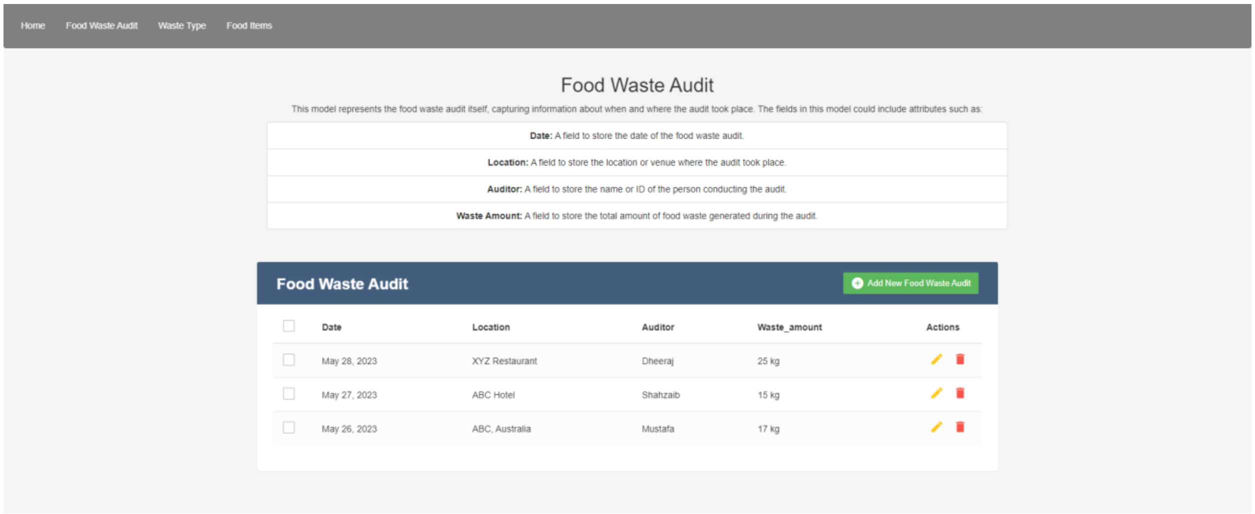
Myapp administration

MYAPP		
Food_ itemss	+ Add	Change
Food_waste_ audits	+ Add	Change
Waste_ typess	+ Add	Change



Welcome to Food Waste & Mitigation!

Name	Roll Number	Email
Dheeraj raj	S352704	Dheerajsamtani@gmail.com
Shahzaib Sial	S352614	Shahzaib.sial17@gmail.
Mustafa raza	S347504	mustafa.razking@gmail.com
Muzafar Shah	s343107	Muzafarshah157@gmail.com



Food Waste Audit

This model represents the food waste audit itself, capturing information about when and where the audit took place. The fields in this model could include attributes such as:

Date: A field to store the date of the food waste audit.

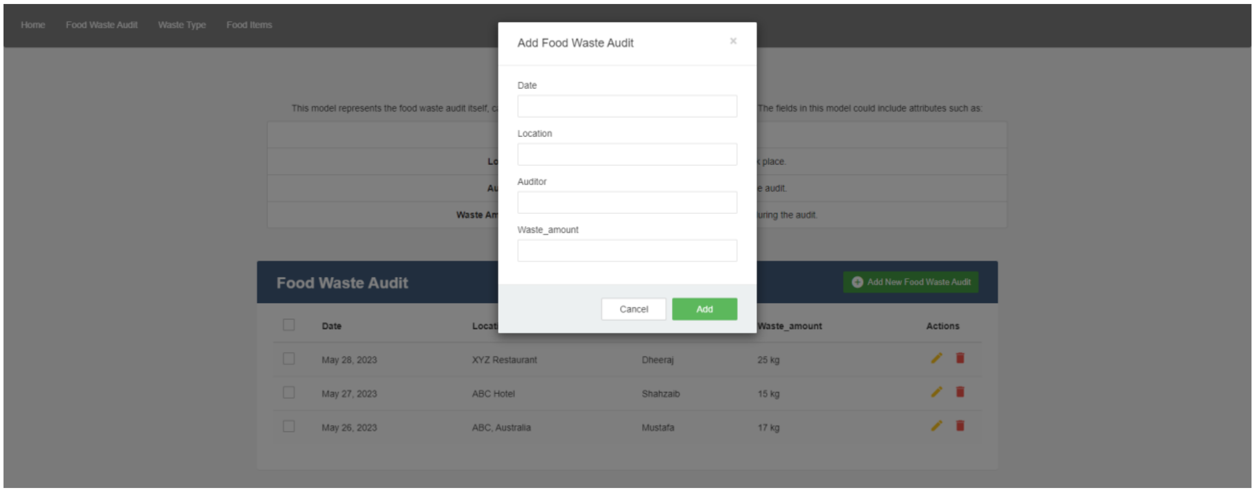
Location: A field to store the location or venue where the audit took place.

Auditor: A field to store the name or ID of the person conducting the audit.

Waste Amount: A field to store the total amount of food waste generated during the audit.

Food Waste Audit

□	Date	Location	Auditor	Waste_amount	Actions
□	May 28, 2023	XYZ Restaurant	Dheeraj	25 kg	✎ 🗑
□	May 27, 2023	ABC Hotel	Shahzaib	15 kg	✎ 🗑
□	May 26, 2023	ABC, Australia	Mustafa	17 kg	✎ 🗑



Add Food Waste Audit ×

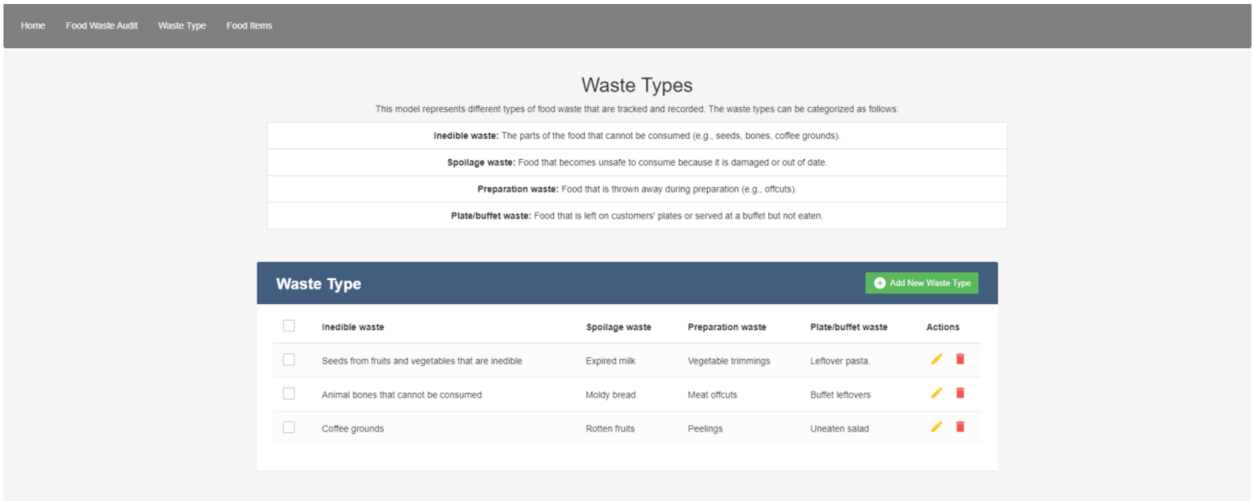
Date

Location

Auditor

Waste_amount

□	Date	Location	Auditor	Waste_amount	Actions
□	May 28, 2023	XYZ Restaurant	Dheeraj	25 kg	✎ 🗑
□	May 27, 2023	ABC Hotel	Shahzaib	15 kg	✎ 🗑
□	May 26, 2023	ABC, Australia	Mustafa	17 kg	✎ 🗑



Waste Types

This model represents different types of food waste that are tracked and recorded. The waste types can be categorized as follows:

Inedible waste: The parts of the food that cannot be consumed (e.g., seeds, bones, coffee grounds).

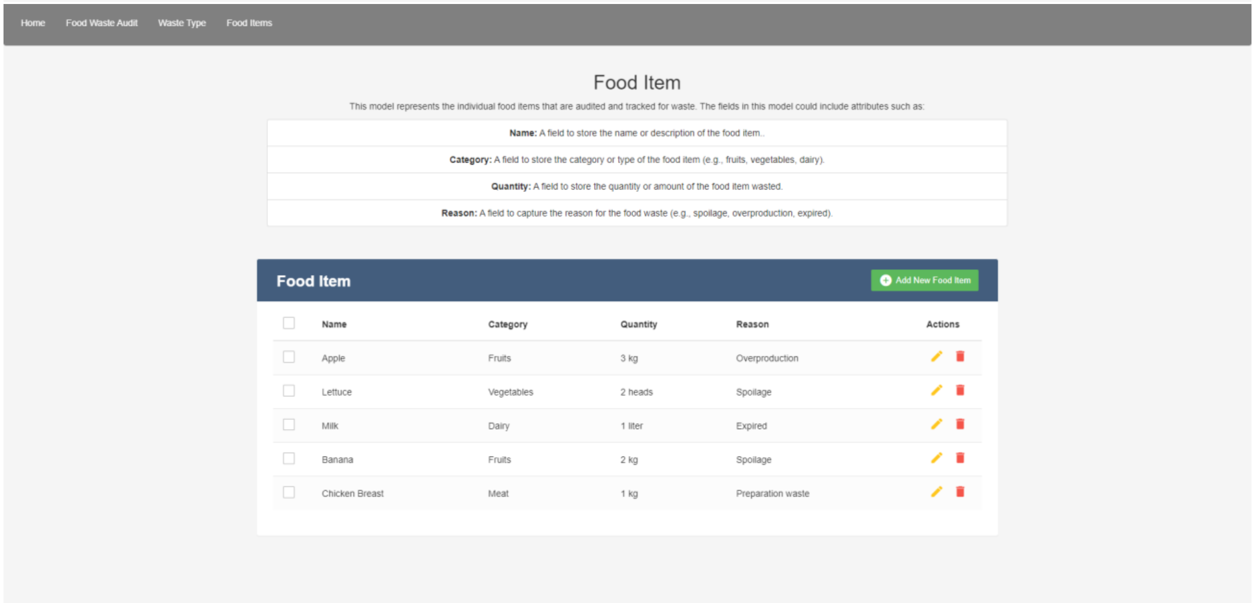
Spoilage waste: Food that becomes unsafe to consume because it is damaged or out of date.

Preparation waste: Food that is thrown away during preparation (e.g., offcuts).

Plate/buffet waste: Food that is left on customers' plates or served at a buffet but not eaten.

Waste Type

□	Inedible waste	Spoilage waste	Preparation waste	Plate/buffet waste	Actions
□	Seeds from fruits and vegetables that are inedible	Expired milk	Vegetable trimmings	Leftover pasta.	✏ 🗑
□	Animal bones that cannot be consumed	Moldy bread	Meat offcuts	Buffet leftovers	✏ 🗑
□	Coffee grounds	Rotten fruits	Peelings	Uneaten salad	✏ 🗑



Food Item

This model represents the individual food items that are audited and tracked for waste. The fields in this model could include attributes such as:

Name: A field to store the name or description of the food item..

Category: A field to store the category or type of the food item (e.g., fruits, vegetables, dairy).

Quantity: A field to store the quantity or amount of the food item wasted.

Reason: A field to capture the reason for the food waste (e.g., spoilage, overproduction, expired).

Food Item

□	Name	Category	Quantity	Reason	Actions
□	Apple	Fruits	3 kg	Overproduction	✏ 🗑
□	Lettuce	Vegetables	2 heads	Spoilage	✏ 🗑
□	Milk	Dairy	1 liter	Expired	✏ 🗑
□	Banana	Fruits	2 kg	Spoilage	✏ 🗑
□	Chicken Breast	Meat	1 kg	Preparation waste	✏ 🗑