Food Donation Tracking and Analytics System

Objective

Create a web-based system to track and analyze food donation data. The project involves both development (frontend and backend) and data analysis components.

Features

1. **Donor and Recipient Management**
    - Allow users to register as either **donors** or **recipients**.
    - Donors can log food donations with details like food type, quantity, and expiration date.
    - Recipients can view available donations and claim them.
2. **Data Analytics Dashboard**
    - Build a dashboard for admins to:
        - Visualize donation trends (e.g., quantity donated over time, most donated food types).
        - Generate insights using **Power BI** or Python for data analysis.
3. **Donation Status Tracking**
    - Show the status of donations: Available, Claimed, or Delivered.
    - Include timestamps for tracking donation history.
4. **Notification System**
    - Send email notifications using Python (or Firebase if comfortable) to notify recipients of new donations in their area.
5. **Responsive Design**
    - Build a responsive interface that works seamlessly across devices.

Tech Stack

1. **Frontend:**
    - HTML, CSS, and JavaScript (for basic interface design)
    - Optional: Flutter (if comfortable, for mobile interface)
2. **Backend:**
    - Python with Flask/Django or Java with Spring Boot (for API and server-side logic)
3. **Database:**
    - Firebase (for simple setup) or MySQL (if more comfortable)
4. **Data Analysis:**
    - Use **Python** for creating donation data reports and insights.
    - Integrate **Power BI** for live data visualization.
5. **Optional Tools:**
    - Use Firebase for notification services (or Python libraries like smtplib for email).

Instructions

1. **Project Scope:**
    - Choose a backend framework you're comfortable with (Python/Java).
    - Build a lightweight frontend interface for basic user interaction.
2. **Data Analytics:**
    - Extract insights from the donation data (e.g., monthly donation patterns, areas with the highest demand).
    - Use **Power BI** to create visual reports or charts that can be embedded into the admin dashboard.
3. **Documentation:**
    - Include a `README.md` file explaining how to set up the project and any dependencies required.
    - Document the data analysis process and key insights.
4. **Submission:**
    - Push the source code to a public GitHub repository.
    - Record a video walkthrough of the project, demonstrating:
        - How users interact with the system.
        - The analytics dashboard and insights generated.