# PREREQUISITE DOCUMENT

| | |
|---|---|
| **Name of Student** | **Komal Milind Deolekar** |
| **Class Roll No** | **14** |
| **Sign and Grade** | |

## Project Title:

Let's Fight Food Waste: ShareBite

## Purpose

The purpose of this document is to outline the technical and non-technical prerequisites required to successfully design, develop, and deploy the Food Waste Management System. The system will facilitate food donations, distribution, and management among various user roles: Donors, NGOs, Volunteers, and Admins.

## Abstract

Food waste and hunger are two problems that coexist in many communities. While edible food is discarded by individuals, restaurants, and supermarkets, millions go hungry every day. ShareBite is a social-impact, web-based platform that bridges this gap by connecting food donors with NGOs and volunteers. The goal is to efficiently redistribute surplus food to those in need. Through role-based access (Donors, NGOs, Volunteers, Admin), ShareBite ensures seamless coordination for real-time food donation, request, and delivery. The platform also highlights top donors and popular NGOs, encouraging active community engagement.

## Objective

- To reduce food wastage and fight hunger.
- To simplify the donation process for food providers.
- To empower NGOs to access food donations quickly.
- To involve volunteers in delivery logistics.
- To enable admins to monitor and maintain transparency.

## Scope

ShareBite is accessible to multiple user roles:

- Donors: Register, log in, post surplus food, manage previous donations.
- NGOs: Register, log in, browse food availability, place requests.
- Volunteers: Accept delivery tasks, update delivery status.
- Admins: Validate users, monitor operations, manage content, and handle disputes.

**The system also includes:**

- Public landing page with a call to action
- Display of top donors and NGOs
- Event announcements and social engagement highlights

## Features / Modules

| **Module** | **Description** |
|---|---|
| Donor Module | Allows donors to register, log in, post food availability, and manage history. |
| NGO Module | Lets NGOs request available food, track requests. |
| Volunteer Module | Assigns delivery jobs to volunteers; tracks completion. |
| Admin Module | Provides admin tools to monitor and manage users and content. |
| Homepage | Landing page to showcase ShareBite's purpose, login/register links, highlights. |

## Key Features

- Role-Based Logins: Donor, NGO, Volunteer, and Admin.
- Donation Posts: Donors can post available food.
- NGO Requests: NGOs can request food based on their needs.
- Volunteer Delivery Coordination: Volunteers deliver food from donors to NGOs.
- Popular NGO & Top Donors Section: Recognize and promote contributors.
- How It Works: Simplified 4-step guide to using the platform.

## Tech Stack

- Frontend: React.js, TailwindCSS
- Backend: Flask (Python)
- Database: MongoDB
- Authentication: JWT Auth (as per implementation)
- Hosting: Vercel for frontend, Render or Railway for backend (or EC2)

## Technical Pre-Requisites

### Frontend Development

1. **Technology**: React.js
    - Latest stable version of React.js (e.g., 18.x).
    - Installation of Node.js (v16 or above) and npm (Node Package Manager).
    - Development framework for styling: Tailwind CSS.
2. **Tools**:
    - Code Editor: VS Code.
    - Browser: Google Chrome or Mozilla Firefox for debugging.
    - React libraries for state management (e.g., Redux or Context API).
3. **Packages/Dependencies**:
    - Axios for HTTP requests.

- React Router for routing.
- TailwindCSS for styling.
- Framer Motion (optional) for animations.

## Backend Development

1. **Technology**: Flask (Python-based microframework)
    - Python (v3.9 or above).
    - Flask and Flask extensions like:
        - Flask-CORS: For handling cross-origin requests.
        - Flask-JWT-Extended: For authentication and authorization.
        - Flask-MongoEngine or PyMongo: For MongoDB integration.
2. **Tools**:
    - Code Editor: VS Code.
    - API Testing Tools: Postman.
3. **Packages/Dependencies**:
    - Werkzeug for security and password hashing.
    - gunicorn (for deployment).
    - Environment Management: pipenv or virtualenv.

## Database

1. **Technology**: MongoDB (NoSQL database).
    - MongoDB Community Edition or MongoDB Atlas for cloud-hosted databases.
    - MongoDB Compass for GUI-based database management.

2. **Schema Design**:
    - Collections: Donors, NGOs, Requests, Volunteers, Logs.
    - Indexing: To optimize frequent queries (e.g., user roles, post status, etc.).

## Deployment

1. **Hosting Platforms**:
    - Frontend: Vercel, Netlify, or AWS Amplify.
    - Backend: AWS EC2, Heroku, or Google Cloud Platform.
    - Database: MongoDB Atlas for scalability.
2. **Version Control**:
    - GitHub or GitLab repository for collaborative development.
    - CI/CD setup using GitHub Actions or Jenkins.

# System Requirements

## Hardware

- Minimum 8 GB RAM and a 4-core processor for development machines.
- For deployment servers:
    - 16 GB RAM and 4 vCPUs for the backend server.
    - Sufficient storage (minimum 20 GB) for MongoDB.

## Software

1. **Operating System**:
    - Development: Windows, macOS, or Linux.

    - Deployment: Linux (Ubuntu preferred).

2. **Development Tools**:
    - Docker (optional): For containerization.
    - Git for version control.
    - PM2 or Supervisor for managing backend services.

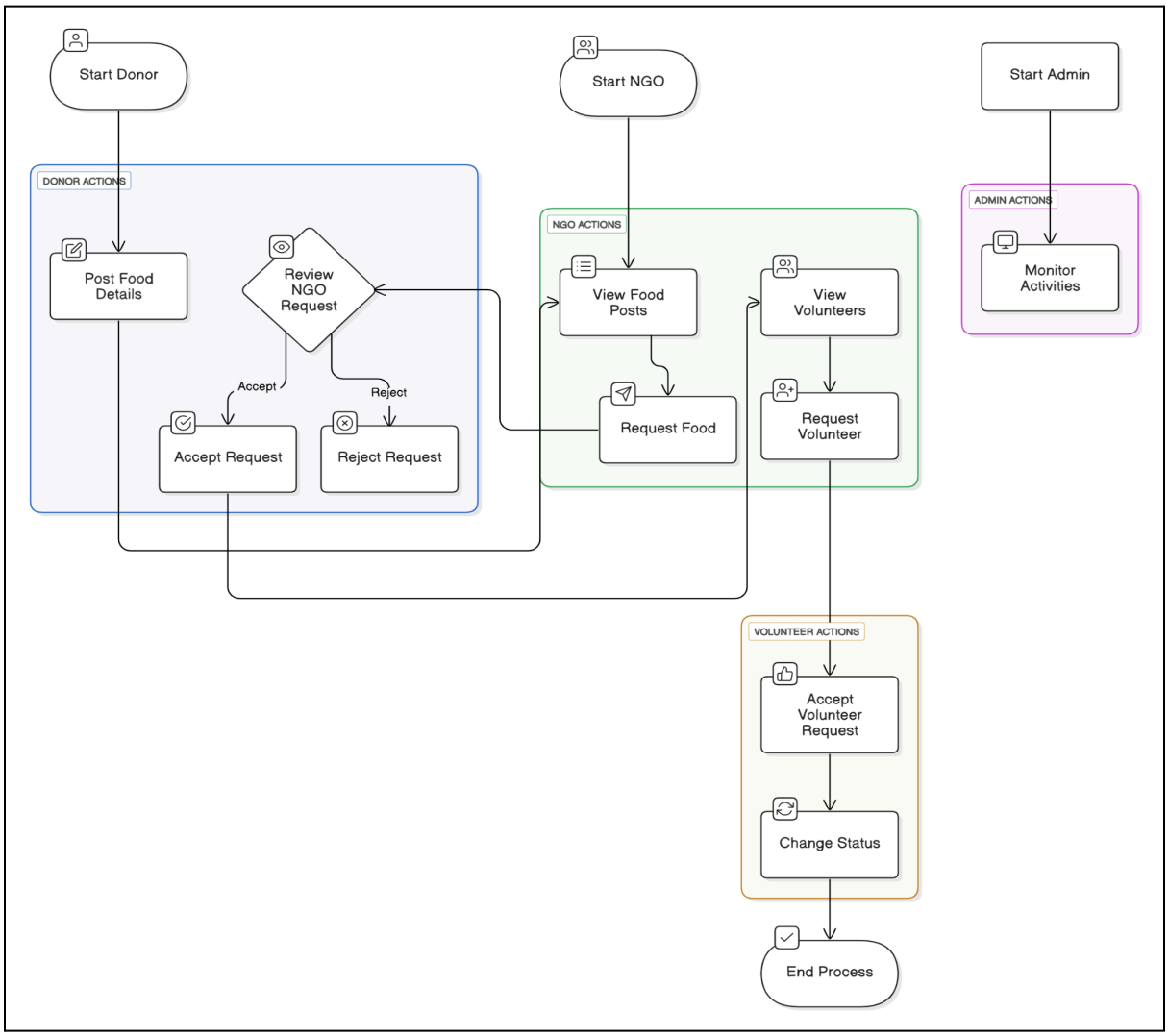
## Functional Requirements

- User registration and login with role selection
- Role-based dashboard rendering
- CRUD operations for food posts
- Request and approval flow for NGOs
- Task assignment for volunteers
- Admin interface for oversight

## Non-Functional Requirements

- Responsive UI for all screen sizes
- Optimized performance
- Secure authentication
- Real-time updates (optional, via sockets or polling)
- Easy navigation and user-friendly interface

# WorkFlow Diagram



# Setup Instructions

## Step-by-Step Instructions

1. **Clone the repository:**

   git clone https://github.com/yourusername/sharebite.git

2. **Backend Setup (Flask):**

   cd sharebite/backend
   python -m venv venv
   source venv/bin/activate  # On Windows use: venv\Scripts\activate
   pip install -r requirements.txt

3. **Create a .env file in backend folder:**

   FLASK_APP=run.py
   FLASK_ENV=development
   MONGO_URI=mongodb+srv://<your_mongo_url>
   JWT_SECRET=your_secret_key

4. **Run the Flask backend:**

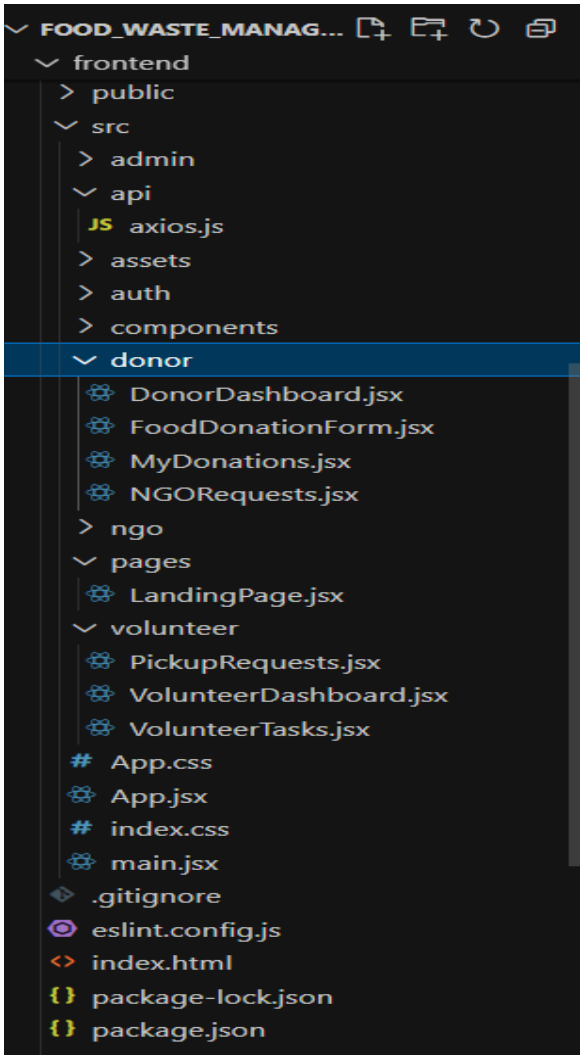
   python run.py

5. **Frontend Setup (React.js):**

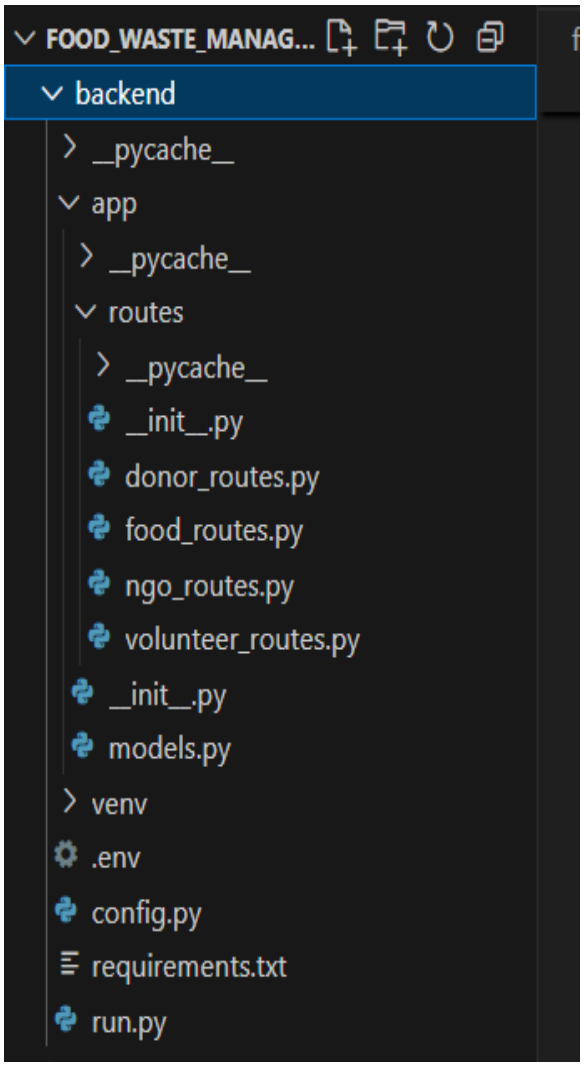
   cd ../frontend
   npm install
   npm run dev

## Project Directory Structure

Frontend :




Backend :



## References

1. MongoDB Documentation: https://www.mongodb.com/docs
2. React Documentation: https://reactjs.org/docs
3. Flask Documentation: https://flask.palletsprojects.com/
4. Tailwind CSS Documentation: https://tailwindcss.com/docs

## Conclusion

ShareBite is more than just a platform—it's a social movement against hunger and food waste. With its simple interface and impactful purpose, it encourages individuals and organizations to collaborate for a better tomorrow. By digitizing the food donation chain, we aim to bring efficiency, transparency, and empathy into every bite shared.

Let's fight food waste, together.