Project Title: NourishNet (*"Bridging surplus and need for a hunger-free future."*)

---

1. Introduction

1.1 Purpose:

The purpose of this application is to bridge the gap between food donors, NGOs, and volunteers by creating a console-based platform for efficient redistribution of surplus food. This app will minimize food waste while providing nourishment to underserved communities.

1.2 Scope:

The app will allow user registration, connectivity, and interaction between key stakeholders:

- Food Donors
- NGOs
- Volunteers
- Administrators

The app will include tools for managing donations, assigning tasks to volunteers, and overseeing NGO operations, with administrative controls for overall management.

1.3 Objectives:

- Reduce food insecurity.
- Minimize food waste.
- Facilitate collaboration among stakeholders using a user-friendly console-based interface.

---

2. System Overview

2.1 Features:

1. **User Registration and Login**:
    - Role-based registration (Donor, NGO, Volunteer, Admin).
    - Secure authentication.
2. **NGO Features**:
    - Manage donation requests.
    - Approve or reject donations.

- Monitor inventory.

3. **Donor Tools**:
    - Create donation entries (e.g., surplus food, quantity, expiry date).
    - View donation status.
4. **Volunteer Management**:
    - Assign tasks (e.g., pickup and delivery).
    - Track volunteer availability.
5. **Administrative Oversight**:
    - Monitor system activities.
    - Generate reports on donations, inventory, and task status.

2.2 Non-Functional Requirements:

- **Usability**: Easy-to-navigate console interface.
- **Performance**: Handle up to 100 concurrent users.
- **Reliability**: Ensure data integrity during transactions.
- **Security**: Role-based access control and encrypted user data.

---

3. Tech Stack

3.1 Programming Language:

- **Python**

3.2 Database:

- **MongoDB** (local or cloud-hosted via MongoDB Atlas)

3.3 Libraries and Tools:

- **PyMongo**: For database interaction.
- **Rich**: For enhanced console output.
- **Argparse**: For handling command-line arguments.

3.4 Version Control:

- **Git/GitHub**

3.5 IDE/Text Editor:

- **VS Code**

---

4. System Design

4.1 Entity Relationships:

- **Users** (Donors, NGOs, Volunteers, Admins): Login details, roles, and permissions.
- **Donations**: Food type, quantity, expiry date, status, and associated donor.
- **Volunteers**: Tasks assigned, availability, and performance tracking.

4.2 Workflow:

1. Users register and log in based on their roles.
2. Donors list surplus food for donation.
3. NGOs approve donations and manage inventory.
4. Volunteers are assigned pickup and delivery tasks.
5. Admins oversee operations and generate reports.

---

5. Deliverables

- Fully functional console-based app.
- User manual for operation.
- GitHub repository with documented code.

---

6. Development Plan

1. Week 1: Setup environment, finalize database schema, and implement user registration/login.
2. Week 2: Develop donor and NGO tools.
3. Week 3: Implement volunteer management.
4. Week 4: Add administrative features and test the application.

---

7. Limitations

- Console-based interface limits user experience compared to a GUI.
- Relies on user internet connectivity for MongoDB Atlas.