Blood Bank Project Documentation

Overview

The Blood Bank project is a simple web-based application that allows users to register as blood donors and request blood based on their required blood group. The project consists of a frontend built using HTML, CSS, and JavaScript and a backend implemented using Python with a simple HTTP server and SQLite database.

Features

- **Donor Registration**: Users can register as blood donors by providing their name, mobile number, blood group, and address.
- **Blood Request**: Users can request blood by selecting their required blood group.
- **Database Management**: Stores donor and request details in an SQLite database.
- **Server-Side Processing**: A Python-based HTTP server processes registration and request data.
- **JSON Responses**: The server returns responses in JSON format to update the frontend dynamically.

Technologies Used

- **Frontend**: HTML, CSS, JavaScript
- **Backend**: Python (http.server, socketserver, sqlite3, urllib.parse, json)
- **Database**: SQLite

File Structure

```
BloodBank/
├── index.html            # Home page
├── register.html         # Donor registration page
├── request.html          # Blood request page
├── blood.css             # Stylesheet
├── server.py             # Backend server
└── bloodbankdb.db        # SQLite database file
```

Implementation Details

Frontend (HTML, CSS, JavaScript)

1. **index.html**
   - Displays project title and navigation links.
   - Provides an introduction to the Blood Bank system.
2. **register.html**
   - Contains a form for donors to register with fields for name, mobile number, blood group, and address.
   - Uses JavaScript to submit form data asynchronously to the backend.
3. **request.html**
   - Contains a form for users to request blood.
   - Displays available donors based on the selected blood group.
   - Uses JavaScript to fetch donor details from the backend.

Backend (server.py)

1. **Initialize Database**
    - Creates an SQLite database (`bloodbankdb.db`) with tables:
        - `donors` (id, name, mobile, blood_group, address)
        - `requests` (id, requester_name, blood_group)
2. **Handle Donor Registration (`/register.html`)**
    - Parses form data from the POST request.
    - Inserts donor details into the `donors` table.
    - Responds with a success message in JSON format.
3. **Handle Blood Requests (`/request.html`)**
    - Parses form data from the POST request.
    - Stores the requester's details in the `requests` table.
    - Fetches matching donors from the `donors` table.
    - Responds with a JSON object containing available donors or a message indicating unavailability.
4. **Enable CORS Support**
    - Allows cross-origin requests from the frontend.
    - Handles OPTIONS requests for preflight checks.
5. **Start HTTP Server**
    - Uses `http.server.SimpleHTTPRequestHandler` to serve requests.
    - Runs on port 8000 and listens for incoming HTTP requests.

How to Run the Project

Prerequisites

- Python installed on your system

Steps

1. Navigate to the project directory:
2. `cd BloodBank`
3. Run the Python server:
4. `python server.py`
5. Open `index.html` in a web browser:
6. `http://localhost:8000`

Future Enhancements

- Implement user authentication for better security.
- Add email/SMS notifications for donors.
- Improve the UI/UX for a better user experience.
- Enhance database management using a full-fledged backend framework like Django or Flask.

Conclusion

The Blood Bank project is a simple yet effective system for managing blood donors and requests. It provides a foundation that can be expanded to create a more robust application in the future.