AMERICAN JOURNAL OF AI CYBER COMPUTING MANAGEMENT
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3069-0102
VOL.6, NO. 2(2026)

Food Link: A Real-Time Digital Food Donation and Distribution Platform

Mudadla Surendra
Reg. No. 24Q71F0033
surendramudadla143@gmail.com
Department of Master of Computer Applications
Avanthi Institute of Engineering and Technology (Autonomous)
Vizianagaram, Andhra Pradesh, India
Under the guidance of Mr. Ch. Bhupathi Raju, MCA, (M.Tech), Assistant Professor
challa.bhupati@gmai.com

Abstract*—Food wastage is a major global challenge, especially in developing countries where large quantities of edible food are discarded daily while many people struggle with hunger. The increasing gap between food surplus and food scarcity highlights the urgent need for an efficient system that can connect individuals or organisations possessing excess food with those who are in need. Traditional food-donation systems depend heavily on manual coordination, physical communication, and slow processes, resulting in delayed delivery, mismanagement, or wastage. The Food Link System is a digital food-donation and distribution platform developed using Python, Streamlit, and SQLite that enables donors to contribute leftover or excess food and allows receivers to request and collect food efficiently. The system eliminates manual dependencies by providing a real-time, automated, and user-friendly application that ensures food reaches the needy in the shortest possible time, and it brings all key stakeholders—donors, receivers, and administrators—onto a single unified interface. Donors can list food items, specify quantity, provide pickup location, and receive delivery updates; receivers can view available food items and request them instantly; and an Admin Analytics Panel provides insights into user data, donations, and request statistics through visual charts. The prototype is implemented in Python with a Streamlit user interface and an SQLite database (foodlink.db), and was validated through more than forty functional, validation, security, performance, and UI test cases spanning authentication, donor, receiver, admin, and analytics modules, all of which passed. By automating the donation lifecycle—from listing food to final delivery confirmation—the Food Link System acts as a smart digital bridge that connects surplus-food donors with individuals who require assistance.*

Keywords*—Food Donation; Food Waste Reduction; Real-Time Platform; Streamlit; SQLite; Donor–Receiver Matching; Admin Analytics; Social Welfare Technology.*

I. INTRODUCTION

Food wastage has become a critical global concern, especially in rapidly developing nations where food production and consumption patterns fluctuate dramatically. Large quantities of edible food are thrown away daily by households, restaurants, hotels, event organisers, and commercial establishments, while millions of people continue to struggle for basic meals. The coexistence of surplus food and widespread hunger indicates a severe lack of structured systems for food redistribution. Traditionally, food-donation activities rely on manual communication, volunteer-driven coordination, and physical verification, which

often lead to inefficiencies; delays in communication, poor tracking, and absence of real-time updates mean that many food items become spoiled before reaching the needy.

In a world where digital transformation is reshaping every industry, technology can play a revolutionary role in addressing this social challenge. A structured, real-time food-donation platform can help bridge the gap between food availability and food demand, making the donation process faster, transparent, and accessible to everyone, and ensuring efficient communication, data tracking, verification, and coordination so that food reaches vulnerable communities on time. The rise of web applications, cloud platforms, and lightweight databases has opened new opportunities to create fast and accessible systems: platforms such as Streamlit simplify interface development, allowing both technical and non-technical users to operate the system with ease, and SQLite provides an efficient, local data-storage mechanism for small to medium-scale applications without requiring complex database setups.

The Food Link System is a comprehensive digital platform developed to address the problem of food wastage and redistribute food efficiently. Built using Python, Streamlit, and SQLite, it connects donors with receivers in a simplified, automated manner, acting as a centralised, user-friendly application where donors can share excess food instantly, receivers can request food in real time, and administrators can monitor activities using analytics. The remainder of the paper discusses related work, problem analysis, system design, implementation, testing, and conclusions.

II. LITERATURE SURVEY

Food wastage and redistribution have been studied by international agencies, non-profits, and digital platforms. The FAO Report (2023) on the global food-waste status documents that approximately one-third of the world's food is wasted annually, with major environmental impact, but does not provide technological solutions to prevent household or commercial food waste, motivating the need for real-time redistribution systems. Gustavsson et al. found that food waste contributes roughly 8–10% of greenhouse-gas emissions, highlighting the environmental importance of redistribution at smaller community scales. Feeding America (2021) operates nationwide food-bank rescue operations, but depends heavily on manual logistics and volunteer support, indicating the need for automation and direct donor–receiver connections.

Several digital initiatives have been proposed. Zero Percent (Chicago) provides pickup scheduling for restaurants and NGOs but requires logistics partners and is not suited to individuals or small donors. Too Good To Go (Europe) reduces waste by selling excess food at a low price, but uses a commercial model rather than a donation-focused one. FareShare (UK) operates large-scale NGO redistribution with high dependency on volunteers and no automated system, while No Food Waste (India) operates an NGO food-collection app in selected cities, still largely manual with limited real-time features. Research by Sharma and Patel (2022) on community food sharing found that local events generate large food waste but lacked digital tracking and real-time information. Collectively, these studies establish a clear gap for a free, donation-based, automated, real-time platform that directly connects donors and receivers, which the Food Link System addresses.

TABLE I. SUMMARY OF REPRESENTATIVE PRIOR WORK

S.No	Source / Year	Focus	Gap / Relevance
1	FAO Report, 2023	Global food-waste status	No technological redistribution solution
2	Gustavsson et al.	Food-waste impact (GHG 8–10%)	Small-scale community gap
3	Feeding America, 2021	Food rescue / food banks	Manual logistics, volunteer-heavy
4	Zero Percent (Chicago)	Pickup scheduling app	Not suited to individuals
5	Too Good To Go (Europe)	Leftover-food sale app	Commercial, not donation-focused
6	FareShare (UK)	NGO food redistribution	Volunteer-heavy; no automation
7	No Food Waste (India)	NGO food-collection app	Mostly manual; limited real time
8	Sharma & Patel, 2022	Community food sharing	Lacks digital tracking

III. EXISTING SYSTEM AND PROPOSED SYSTEM

A. Existing System

The existing system suffers from several drawbacks that make food redistribution slow, unreliable, and unorganised. There is no platform that displays real-time available food items; donors may have food ready for distribution but receivers are unaware of it, and by the time communication reaches them the food may no longer be fresh. Manual coordination via calls and messages introduces time delays, lack of documentation makes it impossible to identify trends, and there is no transparent record of what was donated, requested, or delivered. The result is unreliable distribution, mismatch between supply and demand, and significant food spoilage before it reaches those in need.

Limitations of the existing system:

- No real-time view of available food.
- Heavy dependency on calls and messages.
- Lack of transparency and tracking.
- No centralised platform connecting donors, receivers, and admins.
- No analytics for monitoring or planning.

B. Proposed System

The proposed Food Link System is a digital, real-time food-donation and distribution platform that brings donors, receivers, and administrators onto a single, unified interface. Donors can list food items, specify quantity, provide pickup location, and receive delivery updates; receivers can view available items

in real time and request them instantly; and an Admin Analytics Panel provides insights into user data, donations, and request statistics through visual charts. Implemented with Python, Streamlit, and SQLite, the system is lightweight, free, and easily deployable for communities and NGOs.

Advantages of the proposed system:

- Real-time food-availability view for receivers.
- Direct donor–receiver connection, no manual coordination.
- Transparent tracking of donations, requests, and deliveries.
- Admin Analytics Panel with visual charts for monitoring.
- Lightweight, free, and cross-platform deployment.
- Scales to community, NGO, and small to medium institutional use.

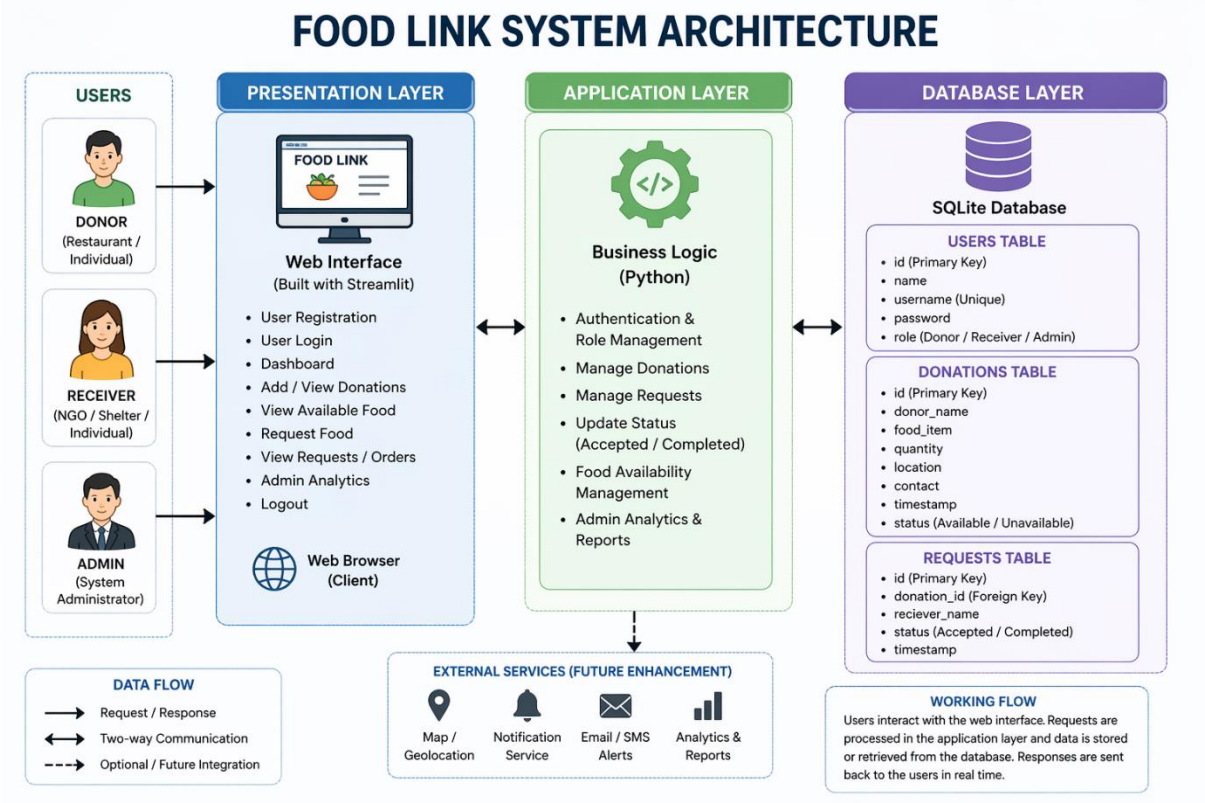
IV. SYSTEM DESIGN AND ARCHITECTURE

A. Requirements

Functionally, the system must support user registration with role selection (donor / receiver / admin), authentication, donor food listing (item, quantity, pickup location), receiver browsing of real-time available food and instant requests, status tracking through to delivery completion, and an admin analytics panel showing user, donation, and request statistics. Non-functional requirements include simplicity, accessibility, scalability, security (credential checks, validation, session control), and a responsive UI suitable for both technical and non-technical users.

B. System Architecture

The architecture is organised around three modules connected through a shared interface and database. The Streamlit-based presentation layer provides separate panels for donors, receivers, and administrators. The application layer implements the donation lifecycle (listing, requesting, status update, completion), authentication, validation, and analytics. The SQLite database (foodlink.db) is the persistent local data store and is connected via the sqlite3 module. The analytics component aggregates database records and renders visual charts in the admin panel. The system's UML diagrams—use case, class, sequence, activity, deployment, component, and object—describe interactions among donors, receivers, administrators, and modules.



C. Workflow

A donor registers, logs in, and lists food items with quantity and pickup details. A receiver registers, logs in, browses real-time available items, and requests one or more items. The donor sees the request and coordinates pickup or delivery; status is updated through to completion. The administrator monitors users, donations, and requests through the analytics panel, which presents visual charts derived from the database. Authentication, validation, and security checks gate each step to ensure data correctness and access control.

V. SYSTEM IMPLEMENTATION

A. Technology Stack

TABLE II. TECHNOLOGY STACK

Component	Technology / Tool
Programming Language	Python
User Interface	Streamlit
Database	SQLite (foodlink.db) via sqlite3
Roles	Donor, Receiver, Administrator
Core Modules	Authentication, Donor, Receiver, Admin, Analytics
Analytics	Visual charts in Admin panel

Component	Technology / Tool
Deployment	Local; cross-platform; cloud-ready

B. Implementation Details

The Food Link System is implemented in Python, with Streamlit rendering the graphical user interface and SQLite providing persistent local storage. Python provides the core logic, Streamlit enables rapid interface development that both technical and non-technical users can operate, and SQLite is selected because it is lightweight, serverless, and supports fast transactions, making it ideal for small to medium-scale donation systems. The backend connects to foodlink.db using the sqlite3 module, manages user accounts, donations, requests, and statuses, and the frontend exposes role-specific panels: donors list and update items, receivers browse available food and submit requests, and the admin views aggregated data and charts.

C. Donor, Receiver, and Admin Modules

The donor module lets a registered donor add a food item with quantity and pickup location and view incoming requests. The receiver module lists available donations in real time, supports the Request action with a single click, and shows the receiver's own request history and statuses up to completion. The admin module aggregates database data and presents counts and trends in visual charts, supports user/donation/request tables, and enforces admin-only access via credential checks. Together these modules cover the full donation lifecycle from listing through delivery confirmation.

VI. SYSTEM TESTING AND RESULTS

The system was validated through more than forty test cases grouped into authentication and registration, donor module, receiver module, admin and database, UI, validation, security, and performance categories. Cases included successful and failing login, registration with role selection and duplicate-username checks, password-strength validation, donor food listing, receiver browsing and request submission, duplicate-request prevention, status update on completion, admin authentication, viewing users/donations/requests, analytics chart generation, large-dataset rendering, admin session timeout, special-character handling, mobile-number validation, and performance under multiple inputs. All test cases passed and behaved as expected.

TABLE III. REPRESENTATIVE TEST CASES

ID	Description	Input	Expected Output	Status
TC001	Login (valid)	Correct credentials	Login successful	**Pass**
TC005	Register new user	Valid details	Registration successful	**Pass**
TC021	View available donations	Receiver login	All available items shown	**Pass**
TC022	Request food item	Click Request	Request sent	**Pass**

ID	Description	Input	Expected Output	Status
TC025	Status update on completion	Click 'Completed'	Status updated	**Pass**
TC031	Admin login (correct)	Admin credentials	Access granted	**Pass**
TC036	Analytics chart	Database data	Chart displayed	**Pass**

A. Observed Results

The implemented system supports the full donation lifecycle: registration with role selection, authentication, donor food listing, real-time receiver browsing, instant requests with duplicate-prevention, status updates through to completion, and an admin analytics panel with visual charts. Edge cases such as empty databases, invalid characters in food names, malformed contact numbers, and large datasets are handled with appropriate error messages or efficient rendering, and security testing confirmed credential checks and session control. The source describes these outcomes qualitatively; no specific numeric performance values are claimed here.

Representative screenshots from the prototype implementation:

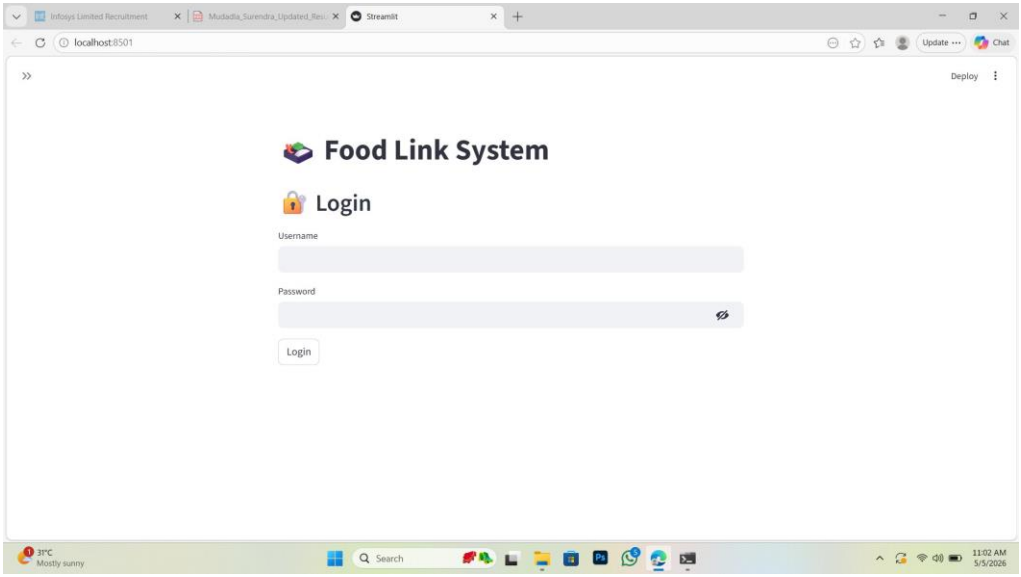


Fig. 1. Registration / login and role selection.

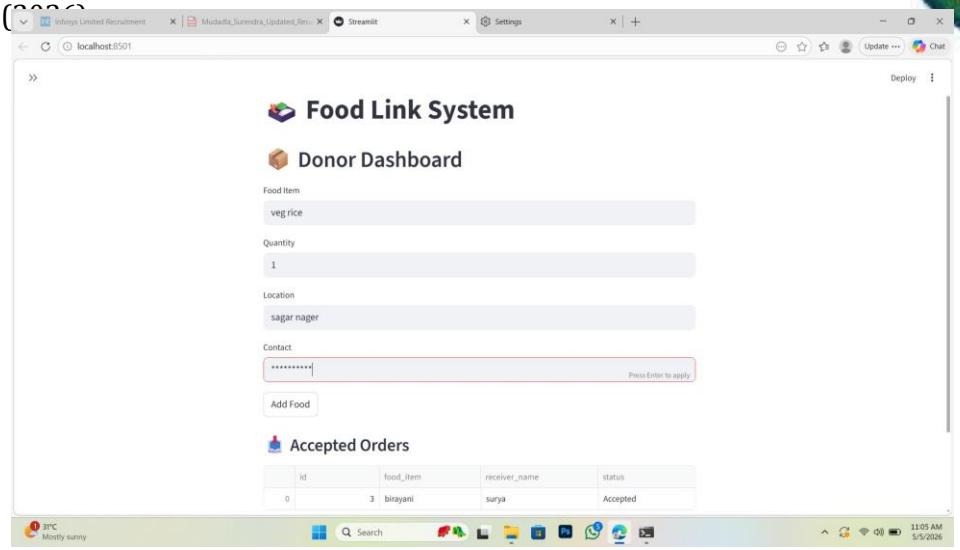


Fig. 2. Donor module (list a food item).

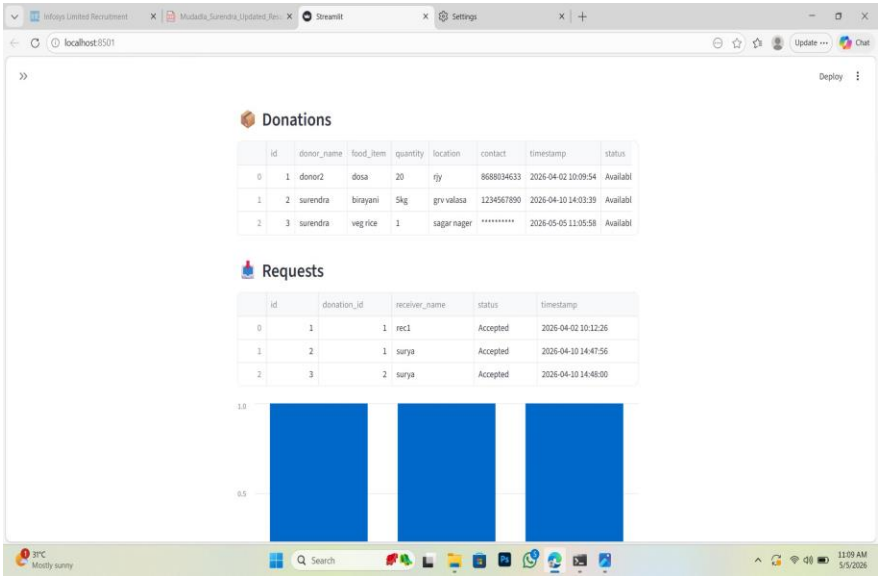


Fig. 3. Receiver module (browse and request).

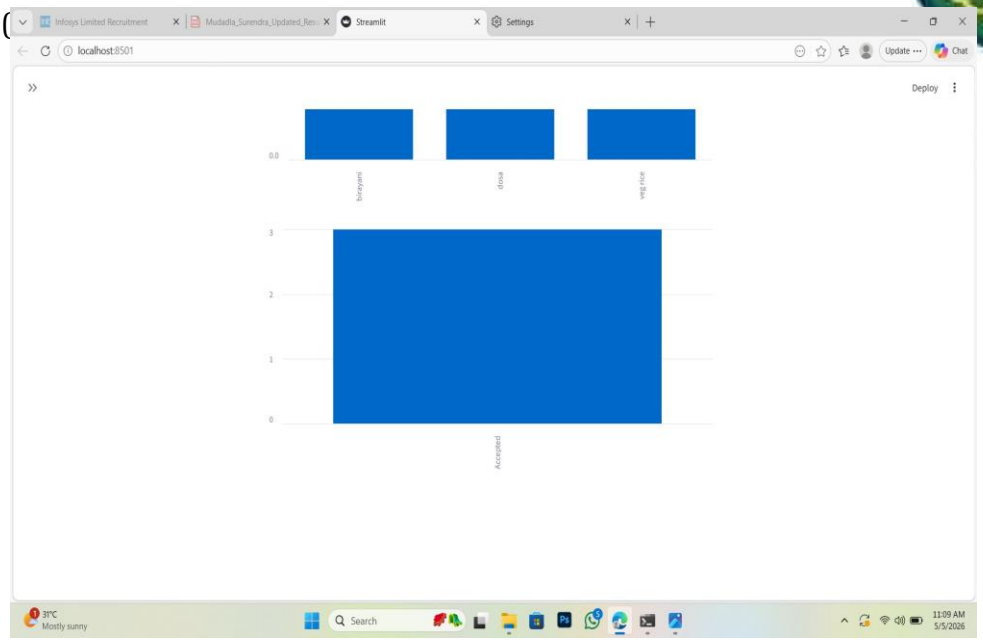


Fig. 4. Admin analytics panel with charts.

VII. CONCLUSION AND FUTURE SCOPE

The Food Link System successfully demonstrates how a lightweight digital platform can address the long-standing problem of food wastage by directly connecting surplus-food donors with individuals and organisations in need. Built using Python, Streamlit, and SQLite, the system replaces manual coordination with a unified, real-time interface in which donors list food, receivers browse and request items instantly, and administrators monitor activity through an analytics panel with visual charts. The donation lifecycle is automated end-to-end, with role-based authentication, transparent tracking, validation, and security controls. Compared with traditional sensor-less, volunteer-driven, or commercial models examined in the literature, the system is free, donation-focused, automated, and accessible to both technical and non-technical users, and functional testing confirmed correct behaviour across more than forty test cases.

Several enhancements can extend the system. Technical improvements include migrating to a cloud database for larger deployments, adding push notifications and email alerts, integrating maps and geolocation for nearest-donor matching, and exposing APIs for integration with NGO and government platforms. Real-world expansion can connect the platform with local kitchens, shelters, and orphanages, and a mobile application would broaden access. Advanced features can include machine-learning-based demand prediction and donor-recommendation, multilingual support, image-based food verification, and stronger security controls such as two-factor authentication and audit logs. Global expansion would require localisation, partnerships with international NGOs, and compliance with relevant food-safety regulations.

REFERENCES

[1] Food and Agriculture Organization of the United Nations, “Global Food Waste Status,” FAO Report, 2023.

[2] J. Gustavsson et al., “Global Food Losses and Food Waste: Extent, Causes and Prevention,” FAO Study.

[3] Feeding America, "Food Rescue Operations — Annual Report," 2021. [Online]. Available: https://www.feedingamerica.org/

[4] Zero Percent, "Digital Food Donation Scheduling Platform," Chicago.

[5] Too Good To Go, "Leftover Food Marketplace," Europe. [Online]. Available: https://toogoodtogo.com/

[6] FareShare, "NGO-Based Food Redistribution," United Kingdom. [Online]. Available: https://fareshare.org.uk/

[7] No Food Waste, "NGO Food Collection App," India. [Online]. Available: https://nofoodwaste.org/

[8] R. Sharma and N. Patel, "A Study on Community Food Sharing and Local Event Food Waste," 2022.

[9] Streamlit Inc., "Streamlit — The Fastest Way to Build Data Apps." [Online]. Available: https://streamlit.io/

[10] SQLite Consortium, "SQLite Documentation." [Online]. Available: https://www.sqlite.org/