

Charity Sphere: Building Trust Connecting Heart Empowering Change

Akshara S A¹, Jithya J M², L S Gourilal³, Sruthy M S⁴, Himaja B M⁵
^{1,2,3,4}Student, Sarabhai Institute of Science and Technology
⁵Assistant Professor, Sarabhai Institute of Science and Technology
doi.org/10.64643/IJIRTV12I10-204464-459

Abstract—Welfare organizations, such as orphanages and senior living facilities, constantly struggle to meet their basic needs in today's society.

- Food
- Hygiene supplies
- Medical supplies.

The absence of a centralized and transparent system frequently results in inefficiencies, mistrust, and inadequate resource distribution, even when donors and volunteers are willing to contribute. In order to solve these problems, Charity Sphere presents a single, AI-powered welfare platform that uses technology, openness, and trust to connect donors, volunteers, and welfare organizations. Scalability, security, and dependability are guaranteed by Charity Sphere's solid full-stack architecture, which includes MongoDB or PostgreSQL for data management, Node.js/Django for the backend, and React.js for the frontend.

Accessibility is improved by hosting services like Vercel and Render, and OpenAI's API incorporates a smart, multilingual chatbot that makes it easier for users to communicate in regional languages like Malayalam, Tamil, and Hindi.

Surplus food management, scheduling donations of clothing and necessities, institutional aid requests, volunteer coordination, real-time donation tracking, and emergency response alerts are some of the main features. The AI-powered chatbot, which not only helps users with questions but also recognizes sentiment and urgency, suggests local causes, and encourages tailored interaction, is a notable feature. Charity Sphere creates an open and cooperative welfare ecosystem by fusing technology and compassion. It guarantees effective volunteer coordination, permits safe and transparent donor participation, and speeds up the creation of requests by welfare organizations.

In the end, CHARITY SPHERE is a digital innovation platform that promotes cooperation, increases trust, and has quantifiable social impact.

Index Terms—AI, Vercel, Render, Node J S, PostgreSQL, MongoDB.

I. INTRODUCTION

CHARITY SPHERE represents a digital innovation platform that enhances trust, fosters collaboration, and measurable social impact. Welfare institutions including Orphanages, Old-age homes, Rehabilitation centers play a vital role in supporting vulnerable populations by providing essential services including Food, Clothing, Healthcare, Hygiene supplies. Despite their social importance, these institutions often struggle with inconsistent access to resources and limited outreach to potential donors and volunteers. Although many individuals and organizations are willing to contribute, the lack of a centralized and transparent system creates significant barriers. Donors are frequently uncertain about where their contributions are most needed and whether they reach the intended beneficiaries. At the same time, welfare institutions find it difficult to communicate their real-time requirements, particularly during emergencies. Volunteers also lack structured platforms to discover and participate in meaningful opportunities.

Additionally, because there are no systematic methods for collecting and redistributing excess food and clothing, large amounts of it are wasted every day.

These difficulties highlight the need for a technologically advanced welfare platform that can organize donations, guarantee responsibility, cut down on resource waste, and improve cooperation between volunteers, donors, and welfare organizations.

II. DESIGN METHODOLOGY

Charity Sphere is an all-inclusive web-based welfare management platform designed to digitally connect

*Donors *Volunteers *. registered organizations through a single, open system. The system eliminates the fragmentation found in traditional charitable processes by providing a centralized platform for institutional campaign administration, volunteer coordination, and donation management. The system is implemented using the modular Django framework. Dashboards with role-specific features, such as campaign management, donation tracking, and volunteer opportunities, are available to donors, institutions, and volunteers. Dynamic dashboards ensure user interaction and real-time data visualization. The system's objectives are to: Provide a transparent and accountable platform for donations; speed up the volunteer recruitment and task assignment process FRONTEND Django Template Language (DTL) is used to build the front end of CharitySphere.

Works with Bootstrap 5 to make the UI design responsive. The design is based on roles dashboard approach, where each verified user is sent to a dashboard that matches their job: Donor, Institution, or Volunteer. A common base.html template makes sure that the UI CSS, JavaScript, and icons should all be in one place, and navigation should be the same across all page's libraries.

Django context variables sent from views to templates make dynamic data rendering possible Templates. This method makes sure that backend data like donations is always up to date campaigns and volunteer work without having to reload the page

Frontend Analysis of the Donor Dashboard

The Donor Dashboard is meant to give donors a quick look at the effect of the contribution. A welcome banner with a gradient makes users more interested and personalization by showing the donor's name in real time.

Some important parts of the frontend are:

- Statistical cards that show the total amount of donations, the number of active campaigns, and the impact score
- A list of recent donations that changes based on activity logs from the back end
- Quick action buttons that make it easy to get to donation- related features
- An impact summary section that encourages people

to keep participating Bootstrap grid systems and utility classes make sure that everything works on desktops, tablets, and phones. Conditional rendering is used to deal with situations like when new users sign up no record of donations.

The Donor Dashboard's Frontend Analysis

The purpose of the Donor Dashboard is to provide donors with an overview of them the contribution's impact. Gradient welcome banners pique users' interest and personalization by instantly displaying the donor's name.

Among the frontend's crucial components are:

- Statistical cards that display the impact score, the number of ongoing campaigns, and the total amount of donations
- A list of recent donations that is updated in response to backend activity logs
- Fast action buttons that facilitate access to features related to donations
- A summary of the effects that motivates participants to continue

Utility classes and Bootstrap grid systems ensure that everything functions on PCs, tablets, and as well as phones.

Level 0 DFD (Context Diagram)

Donor, Volunteer, Institution, and Admin are the four external entities with which the system communicates. Donations, applications, and campaign information are processed as input, and reports and confirmations are produced as outputs.

DFD Level 1

User management, donation management, campaign management, and volunteer management are the modules that make up the system. To store and retrieve pertinent data, each module communicates with the database.

DFD Level 2

Donation validation, campaign fund updates, volunteer application approval, and impact calculation are examples of intricate procedures. Procedures guarantee consistent and accurate data flow. frontend of CharitySphere is implemented using Django Template Language (DTL) integrated with Bootstrap

5 for responsive UI design. architecture follows a role-based dashboard approach, where each authenticated user is routed to a dashboard corresponding to role: Donor, Institution, or Volunteer. A common base.html template ensures UI consistency, navigation uniformity, and centralized inclusion of CSS, JavaScript, and icon libraries. Dynamic data rendering is achieved through Django context variables passed from views to templates. This approach ensures real-time reflection of backend data such as donations, campaigns, and volunteer activities without page reload complexity Donor Dashboard Frontend Analysis the Donor Dashboard is designed to provide donors with an immediate overview of their contribution impact. gradient-based welcome banner enhances user engagement and personalization by dynamically displaying the donor's name.

Important front-end elements

Statistical cards that show the impact score, active campaigns, and total donations Backend activity logs are used to dynamically populate the list of recent donations. Quick action buttons that facilitate quick access to features related to donations The impact summary section encourages ongoing involvement. Complete responsiveness on desktop, tablet, and mobile devices is ensured by Bootstrap grid systems and utility classes. Conditional rendering is used to deal with situations like new users who have never made a donation. Dashboard for Institutions Frontend Evaluation Campaign administration and monitoring are supported by the Institution Dashboard's design. The welcome banner reinforces authenticity by dynamically displaying registration details and institutional identity. Front- end functionalities Campaign data, including the number of active campaigns, volunteers, and money raised Campaign funding status is visually represented by progress bars. Links between campaign management and backend task modules.

Quick action controls for reporting, volunteer directory access, and campaign creation Using Bootstrap cards and contextual color coding, the interface maintains a clear visual hierarchy while emphasizing administrative efficiency and clarity.

III. DATABASE DESIGN

Organizations Donor Profile User Profile of Volunteers Campaign

Donation Relationships with Volunteer Applications A donor is able to make more than one donation. An organization may develop several campaigns. A volunteer may apply to more than one campaign. Every donation is associated with a single campaign. Redundancy is reduced and data integrity is guaranteed by the relational design. Donation Processing Algorithm Algorithms. The campaign is chosen by the donor. The donation amount has been verified. The transaction is noted. Campaign funds have been updated.

Three Volunteer Application Algorithms are used to recalculate the impact score.

1. The opportunity is chosen by the volunteer
2. Validation of application details
3. The record is kept in a database.

The Campaign Progress Algorithm notifies the institution

1. The total amount raised is computed
2. Progress is calculated as a percentage.
3. Metrics on the dashboard are updated.

Frontend Analysis of the Volunteer Dashboard Engagement and opportunity discovery are the main goals of the Volunteer Dashboard. Metrics like hours volunteered, forthcoming events, and finished projects are used in the design to highlight volunteer contributions. Important UI components consist of: Listings of opportunities that are created dynamically using campaign data Components of skill management and profile updates Easy access to invitations and task lists Activity panels are used to monitor volunteer participation. Conditional content rendering is supported by the frontend to guarantee meaningful feedback even in the absence of opportunities or activities. Backend Architecture Overview: CharitySphere's backend was created using the Model-View-Template (MVT) architectural pattern and the Django framework. This guarantees scalability, maintainability, and separation of concerns. Role-based access control is supported by extending Django's built-in authentication system. User authentication, role-based authorization, campaign management, donation processing, volunteer coordination, and safe database interactions are all handled by the backend system. System responsibilities are divided as follows by the MVT pattern: View Layer: Manages response generation,

validation, and request processing. For client-side presentation, the template layer renders dynamic content. In order to ensure safe and efficient data operations, the backend functions as a centralized logic processor that interacts with the database via Django's Object Relational Mapper (ORM). Relationships and Database Design.

The following crucial relationships are enforced by the backend database schema:

Donation by User (One-to-Many) Campaign → One-to-Many Donation Campaign → One-to-Many Application Task → Volunteer (Many-to-Many) Institution → One-to-Many Volunteer Campaign
Referential integrity is guaranteed by foreign key constraints, and performance is enhanced by indexing on frequently queried fields. Database Integration and Model Design User, Donor Profile, Volunteer Profile, Institution, Campaign, Donation, and Volunteer Application are examples of core models. Referential integrity is ensured by each model's correspondence to a relational database table with foreign key relationships. Effective data retrieval for dashboard statistics, recent activities, and progress calculations is made possible by ORM-based queries.

IV. DEMONSTRATION TEST SETUP

To ensure the dependability, usability, security, and accuracy of all features, including user registration (Donor, Volunteer, Institution), login, and AI chatbot interaction, testing is a crucial stage in the development of the Charity Sphere platform. The testing procedure confirms that the system satisfies both functional and non-functional requirements and operates correctly under various input scenarios.

The Django framework and Bootstrap-based user interface templates were used in the development of the application. Several levels of testing were carried out, such as user interface validation, integration testing, and unit testing.

Unit tests ensure each module behaves correctly in isolation. Integration Testing Integration testing verified the interaction between:

- Templates (HTML forms)
- Views (handling POST/GET requests)
- Database models (user roles and chat history)

This ensured complete workflows like (registration → login → chatbot) communication functioned properly.

Functional Testing

Functional testing focused on user-level features:

- Donor Registration
- Volunteer Registration
- Institution Registration
- Login and authentication
- Chatbot conversation flow

Each feature was tested using valid and invalid inputs to verify expected behavior Analysis of Results with Respect to Test Data Registration Testing Analysis When valid test data was used, the system successfully created user accounts for Donor, Volunteer, and Institution roles. Invalid test cases such as empty fields and mismatched passwords produced appropriate error messages, demonstrating strong server-side validation.

Chatbot Testing Analysis

For valid conversational inputs, the chatbot generated context-aware responses and sentiment indicators. Random or irrelevant inputs triggered fallback responses, ensuring robustness against unexpected queries.

Validation Effectiveness The test data confirmed:

- Username uniqueness enforcement
- Secure password validation
- Optional vs mandatory field handling
- System Robustness

Testing with both correct and incorrect datasets proved that:

- The system prevents invalid database entries
- Errors are handled gracefully without crashes
- User experience remains consistent even under edge case.

The testing phase confirmed that the Charity Sphere system operates reliably across all modules. The combination of unit testing, integration testing, functional testing, and manual testing ensured comprehensive validation of user registration, authentication, and chatbot communication.

The results demonstrate that the platform is stable, user-friendly, and capable of handling diverse real-

world inputs, making it suitable for deployment in a transparent charity management ecosystem

V. FUTURE OF CHARITYSPHERE

CharitySphere's Prospects

1. Development of a Welfare

Ecosystem on a Large Scale At the moment, the system only partially links institutions, volunteers, and donors. It has the potential to develop into a nationwide or global welfare ecosystem in the future, bringing together numerous NGOs, governmental organizations, and foreign donors on a single, integrated platform.

2. Intelligent Resource Optimization in Real-Time

Through data-driven decision-making, the platform can be improved to optimize resource distribution automatically in addition to managing donations. This would guarantee that resources are distributed where they are most needed, cutting down on waste and increasing productivity.

Connectivity with Official and Governmental Systems Integration with verified NGO registries and government

welfare databases may be a future development. This would boost credibility, enhance transparency, and make it possible to monitor charitable endeavors more effectively on a bigger scale.

3. Integration with Official and Governmental Systems

Integration with government welfare databases and verified NGO registries may be a future development. This would increase credibility, enhance transparency, and make it possible to better monitor charitable endeavors on a bigger scale.

4. Advanced System for Disaster Management

By coordinating relief efforts, monitoring resources, and mobilizing volunteers in real time across regions, the system can be expanded into a full-fledged disaster response platform.

5. International Cooperation and Cross-Border Assistance

Beyond local use, CharitySphere can facilitate international humanitarian collaboration, enabling organizations and donors from various nations to support global causes.

6. Data-Based Social Impact Evaluation

Large datasets can be produced by the system with constant use. The platform is useful for research and governance because these can be used for analyzing social impact, identifying poverty patterns, and improving policy decisions.

7. Enhanced Digital Inclusion and Accessibility

In order to ensure broader adoption, future scope includes enhancing accessibility for underserved and rural populations through multilingual support, simplified interfaces, and low-bandwidth compatibility.

8. Transformation into an Ecological Digital Welfare Framework

In the end, CharitySphere has the potential to develop into a core digital infrastructure for social welfare, facilitating ongoing participation, long-term funding models, and organized humanitarian end.

VI. ARCHITECTURE

ARCHITECTURAL PARTS

Presentation Layer with Django Templates, Bootstrap 5, and HTML

Role-based view logic in the application layer

Django ORM models are used to manage the data layer. Enforcing authorization and authentication through the Security Layer

MODULES OF FUNCTION

Module for Access Control and Authentication This module controls safe login procedures and prevents unwanted access to secure opinions. Decorators are used in conjunction with Django's authentication mechanism to guarantee only Internal features are interacted with by verified users. Implemented Functions:

- Logging in and out
- Validation of the session
- Role-based restrictions on access

Module for User Profile Management

Each verified account is linked to a single profile entity that characterizes the user. function. To keep databases up to date, additional role-specific data is kept in distinct linked models.

Overall, the testing phase confirmed that every module of the CharitySphere system operates dependably. Key features like user registration, authentication, and chatbot interaction were thoroughly validated through a combination of unit testing, integration testing, functional testing, and manual testing.

The outcomes unequivocally demonstrate that the platform is reliable, user-friendly, and able to process a variety of real-world inputs. This makes it ideal for implementation in an open and effective charity management setting.

VII. CONCLUSION

A comprehensive web-based welfare management platform called CharitySphere was created to facilitate communication between volunteers, donors, and welfare organizations like elderly homes and orphanages. The system tackles important issues with resource allocation—such as food, clothes, hygiene kits, and medical supplies—by offering a centralized, open, and technologically advanced solution. The Django framework was used for backend development in the platform's implementation, and responsive front-end templates were used to guarantee device usability. The system offers secure authentication, role-based registration for donors, volunteers, and institutions, and an AI-powered chatbot assistant that helps users navigate the workflows for donations, volunteer work, and institutional requests. CharitySphere effectively creates a cohesive ecosystem that improves trust, transparency, and effective coordination among all stakeholders by combining user-friendly interfaces, real-time interaction features, and scalable architecture.

REFERENCES

- [1] S. Parkes, "High-Speed, Low-Power, Excellent EMC: LVDS for OnBoard Data-handling," in *Proc. 6th Int. Workshop on Digital Signal Processing Techniques for Space Applications (DSP 1998)*, ESTEC, Noordwijk, 1998.
- [2] A. Sharma and R. Gupta, "Cloud-Based E-Donation System for Transparent Charity Management," *IEEE Access*, vol. 9, pp. 45231–45245, 2021.
- [3] M. K. Singh, P. Verma, and S. Das, "Design and Implementation of Digital Charity Platforms Using Cloud Computing," in *Proc. Int. Conf. Computing, Communication and Automation (ICCCA)*, 2020, pp. 112–117.
- [4] L. Chen and Y. Zhang, "Artificial Intelligence for Social Good: Optimizing Volunteer Allocation in Disaster Management," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 4, pp. 942–950, 2020.
- [5] J. Patel and N. Shah, "AI-Based Chatbot for Multilingual User Interaction in Social Welfare Systems," in *Proc. IEEE Int. Conf. Artificial Intelligence and Virtual Reality (AIVR)*, 2021, pp. 85–92.
- [6] K. Nakamoto et al., "Blockchain-Based Donation System for Enhanced Transparency and Trust," *IEEE Transactions on Engineering Management*, vol. 69, no. 2, pp. 345–356, 2022.
- [7] R. Kumar and S. Mehta, "Secure and Transparent Charity Management Using Distributed Ledger Technology," in *Proc. Int. Conf. Blockchain Technology*, 2021, pp. 210–215.
- [8] T. H. Nguyen and M. T. Tran, "ICT-Based Emergency Communication Systems for Disaster Response," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 275–302, 2021.
- [9] P. Roy, S. Banerjee, and A. Ghosh, "Integrated Emergency Alert Systems Using IoT and Mobile Platforms," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4567–4575, 2021.
- [10] D. Wilson and E. Brown, "Community-Driven Digital Platforms for Resource Sharing and Social Welfare," in *Proc. Int. Conf. Information Systems for Crisis Response and Management (ISCRAM)*, 2020, pp. 134–140.