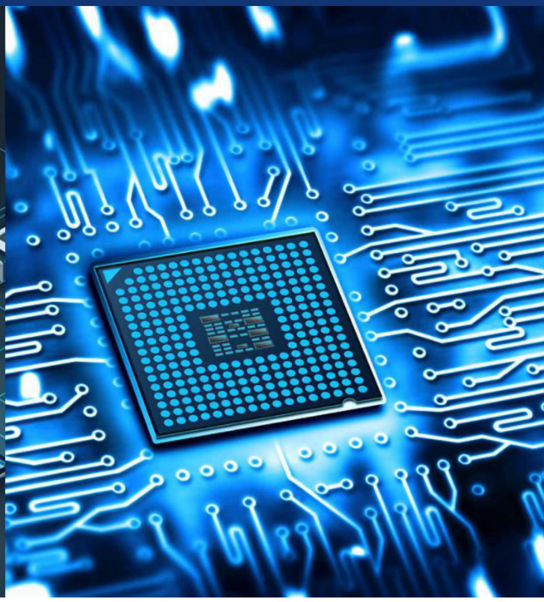
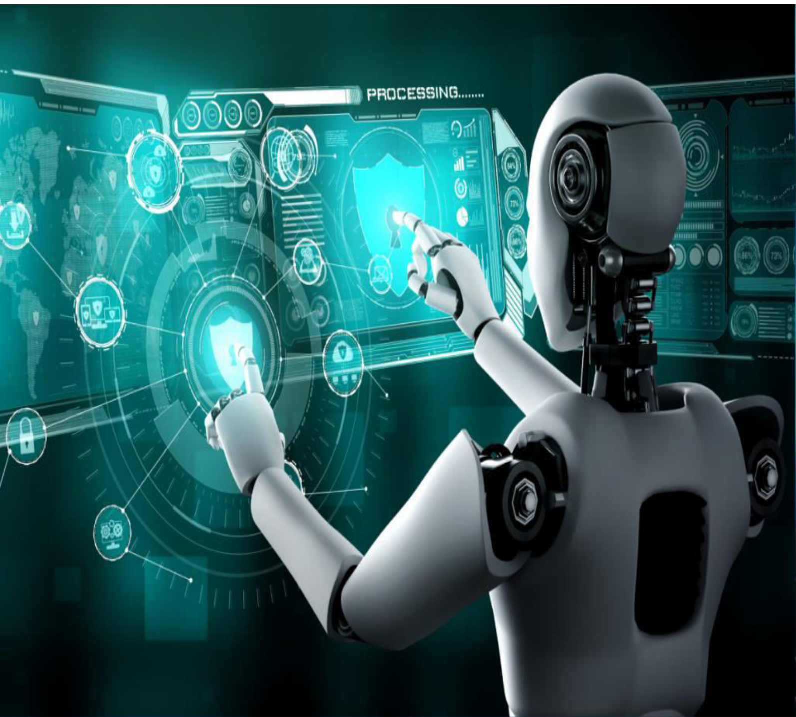
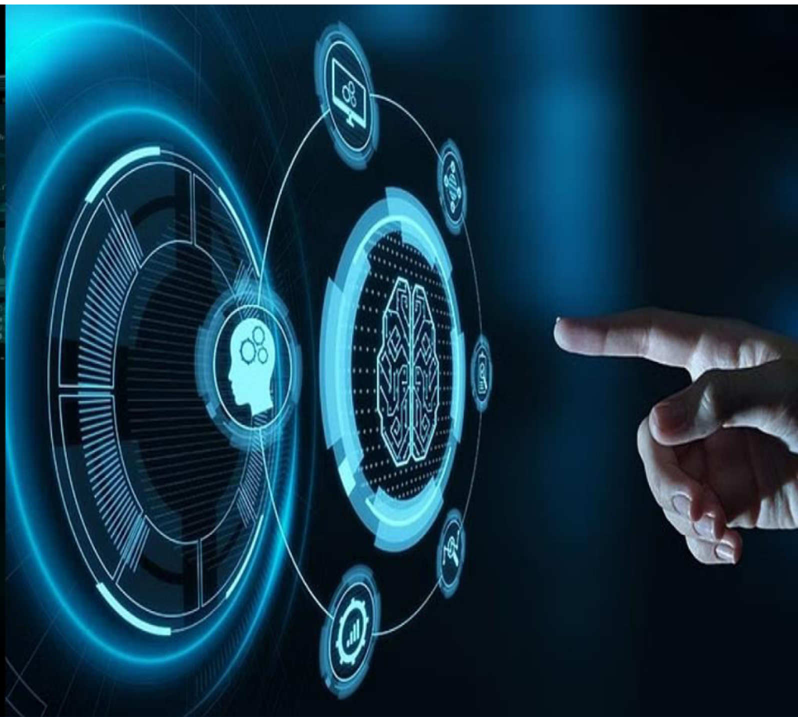
ISSN(O): 2320-9801
ISSN(P): 2320-9798


International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 8.771 **Volume 13, Issue 4, April 2025**

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025 |DOI:10.15680/IJIRCCE.2025.1304275

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Fundief: A Disaster Relief Donation Platform V5 – Working

Dr. Asra Sadaf, Mariya Dhanse, Muaz Dhanse, Shraddhesh Dhandar, Sanskriti Gupta

Assistant Professor, Dept. of CSE (IoT CS BC), A.C. Patil College of Engineering, Navi Mumbai, Maharashtra, India

UG Student, Dept. of CSE (IoT CS BC), A.C. Patil College of Engineering, Navi Mumbai, Maharashtra, India

UG Student, Dept. of CSE (IoT CS BC), A.C. Patil College of Engineering, Navi Mumbai, Maharashtra, India

UG Student, Dept. of CSE (IoT CS BC), A.C. Patil College of Engineering, Navi Mumbai, Maharashtra, India

UG Student, Dept. of CSE (IoT CS BC), A.C. Patil College of Engineering, Navi Mumbai, Maharashtra, India

ABSTRACT: Fundief is a disaster relief donation platform primarily brought into existence to yield the compounded effect of crowdfunding platforms (specially designed for startups) into the donation sector. It incorporates real-time data from geo-location services, governmental agencies, and crowd-sourced information to provide live disaster updates and automatically trigger donation campaigns. The use of Machine Learning and AI-driven recommendations further brings Fundief in the category of advanced software which is on par with the emerging technologies. Other key features include the use of a blockchain-based transaction ledger that ensures the directed release of funds to concerned authorities through the use of smart contracts ensuring transparency, maintaining the trust factor, eliminating any fraudulent money-gathering attempts and ensuring accountability through reliable efforts. Future work includes integrating predictive analytics for early disaster detection and enhancing disaster monitoring with IoT-based real-time sensors, further advancing the platform's capabilities. All the aforementioned aspects of the platform position Fundief as a scalable, impactful, and reliable tool for both immediate disaster response and long-term preparedness. This paper outlines the technical design and implementation of the platform, evaluates its performance, and explores its potential applications in global disaster management.

KEYWORDS: crowdfunding, disaster relief, automation, blockchain, geo-location, donation, fundraising, charity, software, Ethereum Virtual Machine (EVM), smart contracts, blockchain, ledger, non-fungible tokens (NFT).

I. INTRODUCTION

With the increase in frequency and intensity of disasters, natural and man-made, there is an increased need for efficient and transparent disaster relief efforts. The existing systems fail to cater to the affected communities, thus creating critical gaps in both funding and resource distribution. In light of current affairs, Fundief is a collective effort towards financial aid to disaster-stricken communities, ensuring transparent operations among the donors, volunteers and receivers, precisely focusing on bringing the essence of crowdfunding into the donation sector. According to the World Disasters Report, "crowdfunding is an essential tool in mobilizing funds quickly to address urgent humanitarian needs, particularly in areas affected by natural disasters" [7]. Thus, the platform attempts to provide a bigger picture of disaster scenarios by combining real-time data from both governmental and non-governmental sources. Fundief aims at utilizing newer technologies to ensure minimal technological gaps and leverage their advantages in terms of efficiency, security, transparency, and ease of use. The project involves an amalgamation of budding tools and technologies like blockchain technology that ensures utmost security and tamper-resistant records of/for proof; AI-enabled recommendations for quicker discovery of similar or related donation campaigns; NFT attributions to dedicated entities of the platform; Social Media Integration to ensure real-time, visual, and transparent familiarity of workflow under a particular campaign; Geolocation service integration, etc. Fundief strives to become a remedy of prompt financial support to global communities in need.

A. Background and Motivation

The existence of platforms all over the world dedicated to disaster relief is not less, however, these platforms tend to work privately and in an isolated manner. The purpose behind building a unified disaster relief platform like Fundief is to provide a means for people around the world to come together and contribute to disaster management. The United

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Nations Office for Disaster Risk Reduction (UNDRR) defines disaster management as "the organization, planning, and application of measures preparing for, responding to and recovering from disasters" [9]. This process involves four major steps: Mitigation – According to the World Health Organization (WHO), "mitigation refers to measures aimed at reducing or eliminating risk to people and property from the impacts of hazards"[8], Preparedness – The Federal Emergency Management Agency (FEMA) describes preparedness as "actions to ensure readiness to cope with the impacts of a disaster"[4], Response – According to UNDRR, a response is "the provision of emergency services and public assistance during or immediately after a disaster"[5], Recovery – FEMA defines recovery as the "activities and programs designed to return conditions to a level of normalcy after a disaster"[3]. Fundief aims to facilitate Disaster Management by bringing together all working bodies, governmental and non-governmental, organizations and individuals, to create a consolidated and compounded effect of those who want to donate to the cause towards the globally affected communities. Fundief will Fundief fosters trust, accountability, transparency, and efficiency, and adheres to the UN's Sustainable Development Goals (SDGs) to create a sustainable recovery plan.

B. Problem Statement

Crowdfunding platforms have solved the financial needs of startups and growing businesses by the principle of compounded effect where the aid is multiplied by working not on an individual scale, but on a cohesive one, that pitches in all the aids together, small or big. The disaster relief sector lacks such a cohesive effort and adequate resources as organizations work at individual levels to combat the aftereffects of disasters. This platform seeks to unify those eager to contribute to disaster management and recovery, providing a trusted centralized source for donations and support.

C. Objectives

The main objectives to be fulfilled by the development of this project adhere to the need for equalizing centralization and decentralization in disaster relief efforts, whilst facilitating real-time donations and support from/to anticipating/affected communities.

• To provide Financial Support by facilitating monetary donations to assist victims of natural, man-made, and biohazards.
• To bridge the Gap and connect communities in need with providers and volunteers using advanced technologies to enhance community engagement in Disaster Recovery.
• To provide Live Updates using Advanced Technology – gathering and displaying real-time requests from authorities to showcase the urgency and scale of damages, while using blockchain features to promote transparency and build trust among helpers and receivers.
• To raise awareness by educating the public about the impact of disasters and the importance of timely support and fostering innovation in disaster relief technologies.
• To align with the UN's Sustainable Development Goals (SDGs).

The goals fulfilled by Fundief are: Goal 11 (Sustainable Cities and Communities, Goal 13 (Climate Action), Goal 15 (Life on Land), Goal 3 (Good Health and Well-Being), Goal 9 (Industry, Innovation and Infrastructure)

• To leverage advanced data for precision.
• To support long-term recovery and resilience.

D. Scope of the Project

This project – Fundief, is a web-based disaster relief platform that integrates budding technologies like blockchain for transaction transparency, AI/ML for recommendation and pattern recognition, and geolocation services for real-time disaster tracking. The system provides disaster organizers with tools to create verified campaigns while enabling donors to contribute securely and track their impact.

The scope includes:

- Developing a full-stack web application.
- Integration of blockchain (Ganache – Ethereum-based development process) for implementing smart contracts to record transactions.
- Real-time data integration through geolocation and news APIs.
- Secure user authentication and campaign management.
- Flexibility in software design for future enhancements such as IoT sensors and mobile applications.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. LITERATURE REVIEW

Traditional crowdfunding platforms dedicated to disaster relief like GoFundMe, GlobalGiving, and JustGiving have played a pivotal role in raising funds for disaster relief efforts. These platforms provide a space for individuals and organizations to create campaigns, enabling donors to contribute financially to causes. However, these systems rely heavily on manual updates from campaign organizers, which can lead to delays in providing critical information to donors. Often, limited transparency can be seen regarding how funds are being distributed and used, which may strike donor trust over time. Despite their utility, these platforms lack integrated disaster-specific technologies such as real-time data feeds and predictive analytics to enhance relief efforts [1]. Platforms like Global Disaster Alert and Coordination System (GDACS) and NASA's Earth Observing System Data and Information System (EOSDIS) do provide real-time information about natural disasters, such as earthquakes, tsunamis, and floods and raise alerts to authorities, but they fail to provide an immediate connection needed between victims and helpers for a prompt disaster recovery response. These systems focus solely on disaster detection and alerts rather than creating an ecosystem where disaster relief and funding can be managed in real-time [2]. Though there is an emergence of platforms like BitGive and Alice.si that integrate blockchain technology to introduce transparency, these are relatively niche and do not incorporate real-time disaster updates or AI-based recommendation systems to further optimize donation flows and decision-making during disasters [6].

The existing crowdfunding platforms operate in an isolated manner, directly affecting the reach of user density willing to offer donations. While the systems – GDACS and NASA's EOSDIS provide real-time data alerts on disasters, they simply act as alert systems and not as profound disaster response or management platforms. This poses a huge gap between real-time disaster monitoring services and donation platforms. Due to the dispersed nature of donation platforms, donors find it difficult to find similar contributing campaigns in one place and are required to navigate several platforms to find their areas of interest. Besides, there is no integration of newer technologies like AI, Machine Learning, and Blockchain, which are considered to be the future of the Internet, thus making the currently existing platforms outdated. With the increase of digital and cryptocurrencies, sooner or later, these platforms will die out if not revolutionized. Another one of the biggest challenges of transparency and trust still exists with traditional platforms. There is often little to no knowledge of how the donations are being used, the progress report on the affected community or region, the real-time feed and condition of people, etc.

III. REQUIREMENT SPECIFICATIONS

This section highlights detailed requirements for the development and implementation of the Fundief platform, focusing mainly on functional behaviour, quality attributes, user role specifications, etc.

A. Functional Requirements

Functional requirements define what Fundief should be able to achieve to meet both user and client needs. The platform must:

ID	Requirement	Description	Priority
FR-01	RBAC	allow role-based authorization (RBAC) after logging in/signing up successfully;	Very High
FR-02	Admin Moderation	allow admins to: (a) add/remove/modify users and their roles as per verification; (b) approve/reject campaign approvals as per verification; (c) oversee all operations headed by all user roles as per RBAC; (d) receive and adhere/act according to customer feedback;	High
FR-03	Live Disaster Tracking	detect live disasters through integrated news APIs and allow all organization user roles to initiate campaign requests;	Moderate
FR-04	Volunteer Coordination for Donation Progress	allow volunteer user roles to make changes/updates in disaster stats and record changes in disaster fundraising goals as per spending of finances for relief efforts;	High
FR-05	Donation Process	allow regular registered users to donate to any disaster of their will, and search for any disaster based on name/priority/location,	Moderate

		etc;	
FR-06	Blockchain Ledger Access	record all financial transactions on the blockchain via smart contract calls and display all blockchain-recorded transactions publicly on the website;	Very High
FR-07	Live Disaster Updates	allow volunteer users assigned to disasters to broadcast live updates in the form of social media engagement or live feeds;	High
FR-08	Customer Support and Feedback	direct customer queries and suggestions to the designated team	Low

Table 1: Functional Requirements (FR)

B. Non-functional Requirements

This section outlines the quality constraints that the platform must adhere to and how it should behave. System should:

ID	Type	Description
NFR-01	Performance	support at least 10,000 concurrent users without lags or delays, especially at high disaster alerts
NFR-02	Usability (Accessibility)	be accessible worldwide, responsive and meet rapidly evolving web technology
NFR-03	Maintainability	be coded in modules for easy bug fixes and updates
NFR-04	Interoperability	support integration with third-party APIs (geolocation, news, payment gateways, etc.)
NFR-05	Security	have encrypted data exchange, without any chances of fraud with the help of smart contracts and blockchain
NFR-06	Transparency	capture and display all financial interactions as recorded on the blockchain, viewable to the public
NFR-07	Scalability	be horizontally scalable (with the preferable use of MongoDB) to handle increasing traffic and campaign data
NFR-08	Compliance	Ensure GDPR and data compliance policy, especially for user-sensitive data

Table 2: Non-functional Requirements (NFR)

C. Software and Hardware Requirements

Server Infrastructure: Includes Cloud Hosting Platforms – AWS (Amazon Web Services) or Google Cloud Platform (GCP); server specifications include CPU with multi-core processors (e.g., Intel Xeon or AMD EPYC), a minimum of 16 GB RAM for development; production systems may require up to 64 GB or more for scaling during peak load times, cloud-based storage systems like Amazon S3 for storing images, videos, and real-time data logs, local SSD storage with 500 GB to 1 TB for caching and quick access to high-priority data; load balancer to ensure scalability and availability, Elastic Load Balancing (ELB) on AWS or Cloud Load Balancing on GCP to distribute incoming traffic across multiple servers, ensuring optimal resource usage; AWS S3 with versioning enabled or Google Cloud Storage for disaster recovery and backup.

Client-side Hardware: Requirements are minimal with desktop PCs and laptops with Intel i3 processors (or higher), 4 GB RAM (minimum), and a reliable internet connection; or mobile devices with at least 2 GB of RAM, running Android 7.0 (Nougat) or above, or iOS 12 or above.

Development Hardware: Includes CPU – Intel i5 or AMD Ryzen 5 (or higher), RAM – 16 GB RAM for running multiple development tools (IDEs, emulators, etc.), storage – at least 256 GB SSD for fast I/O operations and sufficient storage for local databases, testing, and deployments.

Frontend Development: Done using React.js for a web-based user interface, React Native for cross-platform mobile app development (future scope), HTML5, CSS3, and JavaScript to build the structure, design, and interactions of the web version, Bootstrap to expedite UI development, API Integration Tools to handle API requests between the frontend and backend.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025 |DOI:10.15680/IJIRCCE.2025.1304275
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Backend Development: Involves the usage of Flask framework, Node.js for creating RESTful APIs, MongoDB (NoSQL Database) for storing disaster data, user information, donations, and real-time logs, real-time data processing (Socket.io) to implement real-time communication between the server and clients, blockchain integration (Ethereum-based Ganache) for transaction transparency, smart contracts written using Solidity (for Ethereum) to handle donations, ensuring secure and transparent fund management, AI Framework (TensorFlow or PyTorch) for disaster recommendation systems, payment gateways (Stripe or PayPal) for processing donations.

Data Management and Storage: Amazon S3 or Google Cloud Storage for handling large volumes of unstructured data, Redis for in-memory caching, speeding up real-time data retrieval processes.
Development and Testing Tools: Visual Studio Code for both frontend and backend development, Git and GitHub version control using Git for continuous integration (CI) and automated testing. Postman to test API endpoints.
Real-Time Data APIs and Libraries: Such as OpenWeatherMap API for real-time weather updates using meteorological data during disasters, Twitter API to pull in real-time crowd-sourced disaster updates from social media, and Governamental APIs (NASA) to visually represent disaster zones on the platform's interface.

D. User Requirements
There are 4 major users based on role-based access control (RBAC): **Admin** – responsible for verifying and approving organizations, volunteers, and other roles; approving or rejecting campaign requests; removing fraudulent campaigns/roles/activities; viewing platform analytics and logs. **Organization** – responsible for initiating campaign approvals; adding disaster details, images, donation targets, etc.; overseeing disaster statistics; verifying volunteer support requests. **Volunteer** – responsible for coordinating with nearby locations to aid disaster relief; getting assigned to field support tasks; reporting on activities and target work done. **User** – registered users can donate securely to campaigns; track personal history and impact; and view and search disasters. Unregistered users can view the live activity, and search for campaigns, however, cannot donate unless registered.

IV. SYSTEM DESIGN

The system follows a modular architecture consisting of:
Frontend Layer: Built with HTML, CSS, and JS – handles user interface, campaign listings, donation forms, and volunteer dashboards.
Backend Layer: Built with Flask framework – manages routing, user requests, API endpoints, database queries, and business logic.
Database Layer: Built with SQLAlchemy + MySQL/MongoDB – stores user data, campaign info, donations, roles, and logs.
Blockchain Layer: Built with Ganache and Solidity – records donation transactions with smart contracts.
Geolocation and APIs: Fetches real-time disaster data and auto-populates campaigns.

Navigation and Function
The navigation of Fundief revolves around five main tabs – Home (displays platform details), About (displays information about the developer team), Projects (displays disaster campaigns), Contact Us (for customer feedback), and Login/Register.

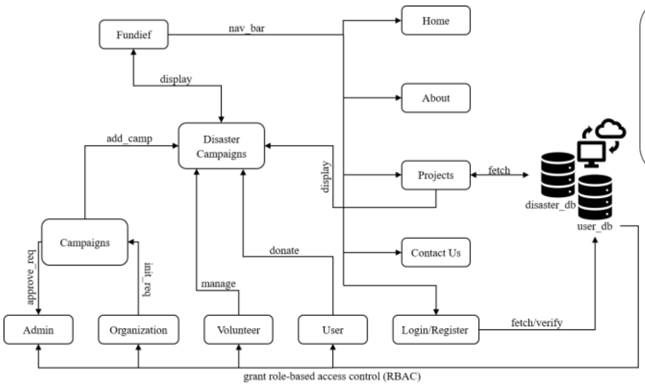


Figure 1: Navigation Flowchart

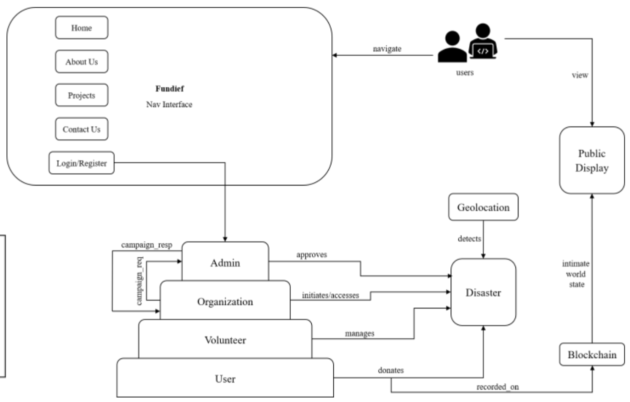


Figure 2: Functional Diagram

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

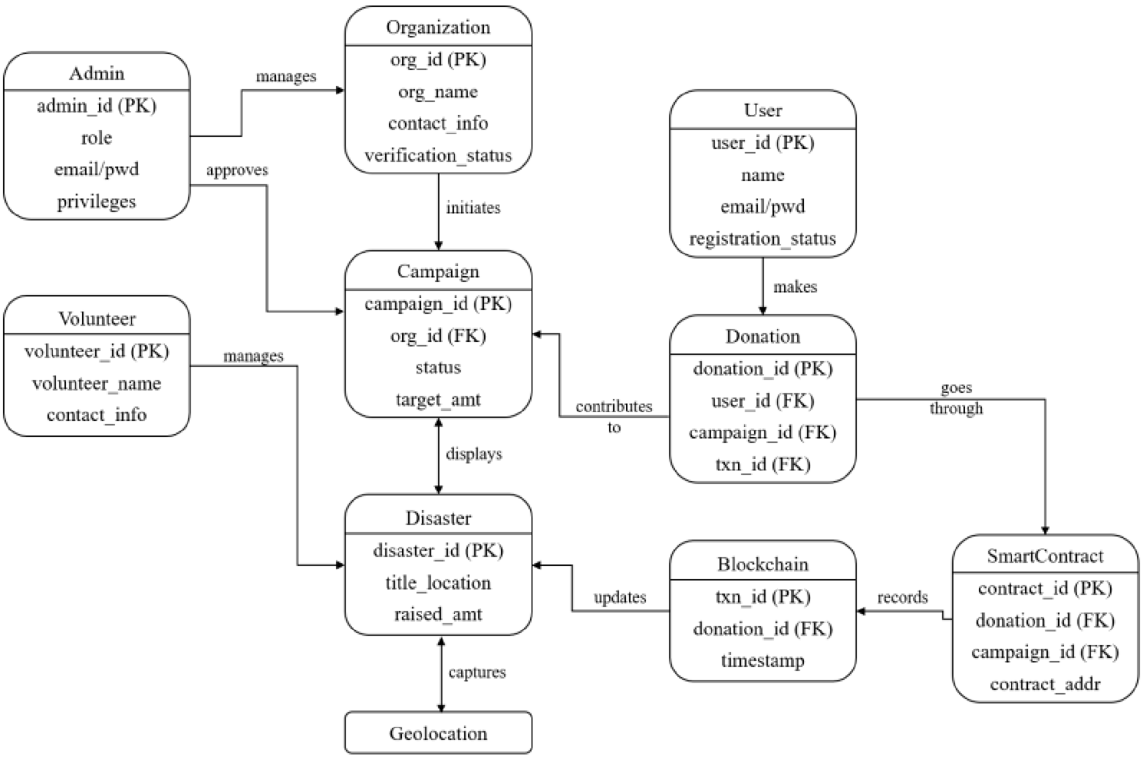
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

- Users (registered/unregistered) can navigate through five modules.
- Users can log in/register to get role-based access (four modules – Admin, Organization, Volunteer, User), based on which they can perform designated functions.
- Admins can verify organizations/registrations and campaign approvals.
- Organizations can raise campaign requests for admin approval, oversee disaster campaigns, and list disaster goals such as target amount, food required, clothing required, etc.
- Volunteers can update live feeds and coordinate disaster management by facilitating financial transactions.
- Users can search for disasters, donate to the disasters, and view overall donation activity.
- Disasters are incorporated either by approval requests of verified organizations or automatically by geolocation services.
- All donations are recorded on the blockchain and reflected publicly on the platform.

DATABASE DESIGN

-User – can be an Admin, Organization, Volunteer, or general user (1-to-1 role-based link).
-Admin – a managerial role for approvals and verifications.
-Volunteer – a type of user that may be assigned to a campaign, stores volunteer names, contacts and their assignment info.
- Organization – is linked to one user and can initiate and manage multiple Disaster Campaigns.
- Disaster Campaign – created by one Organization and can receive multiple Donations and Volunteer assignments.
- Donation – made by a User to a Disaster Campaign and logs a Blockchain Transaction.
- Smart Contract – linked to one Campaign and contains immutable blockchain data.

Figure 3: ER Diagram



Functional and Authentication Module

The platform contains four functional modules for different roles. An in-depth description of functional user modules has been explained above (Section III.D).

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025 |DOI:10.15680/IJIRCCE.2025.1304275

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

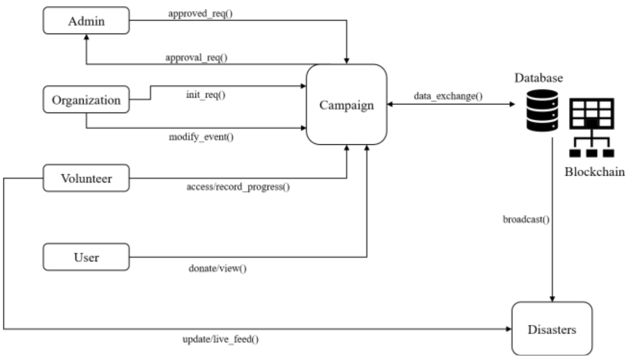


Figure 4: Functional Module

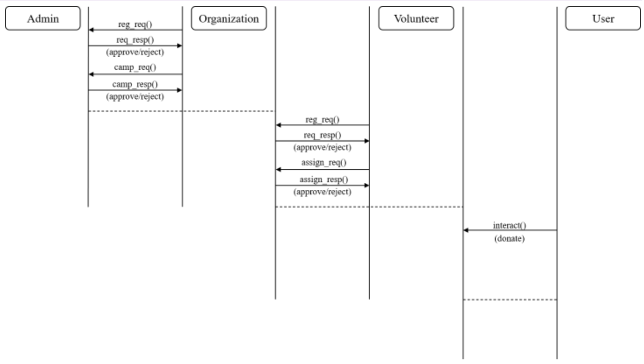


Figure 5: Auth Module

Admin – Organization authentication involves registration requests and campaign approvals. Organization – Volunteer authentication involves registration requests and assignment approvals. User – Volunteer interaction involves simple donations.

V. IMPLEMENTATION PLAN

The methodology for the development of Fundief is going to follow typical steps that are essential for baselining the foundation of a project. The process starts with problem identification & thorough research in the connected areas, design and development of core modules that involve frontend/backend development along with additional technologies required to create a full-stack project. Additional features involve AI Integration for recommendations and detection, blockchain for transparency, geo-location services and news APIs. Furthermore, testing and optimization are done to ensure ease of use and exceptional user experience. The platform is then continuously monitored and a feedback loop allows for quick response to any issues that might arise. The entire process starts again with the next iteration until the final loop of requirements is achieved. There is use of Spiral model integrated with Incremental Development for Software Development Lifecycle (SDLC). The Spiral Model allows for continuous refinement of requirements and iterative development of functional modules.

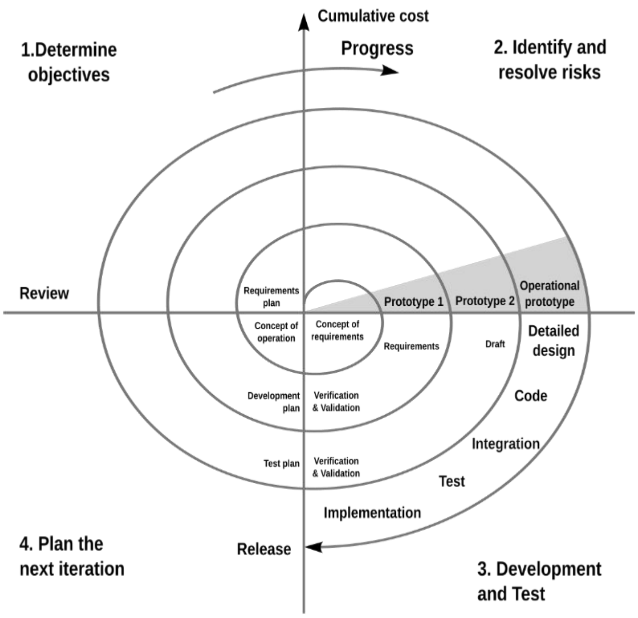


Figure 6: Spiral Model for SDLC

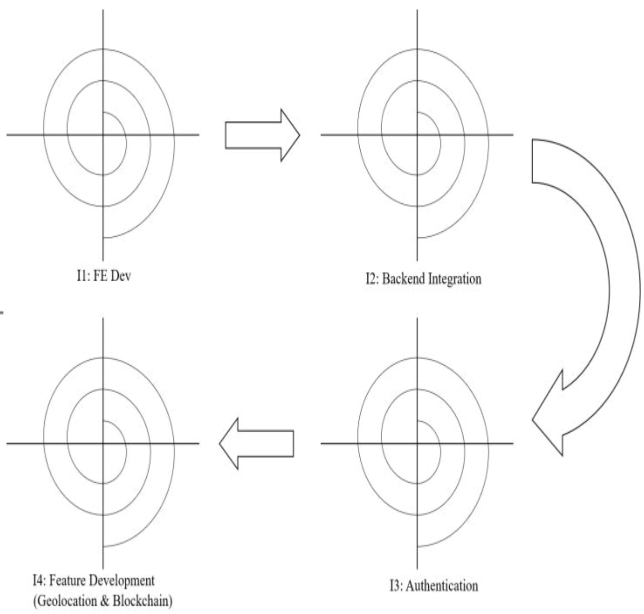


Figure 7: Incremental Model

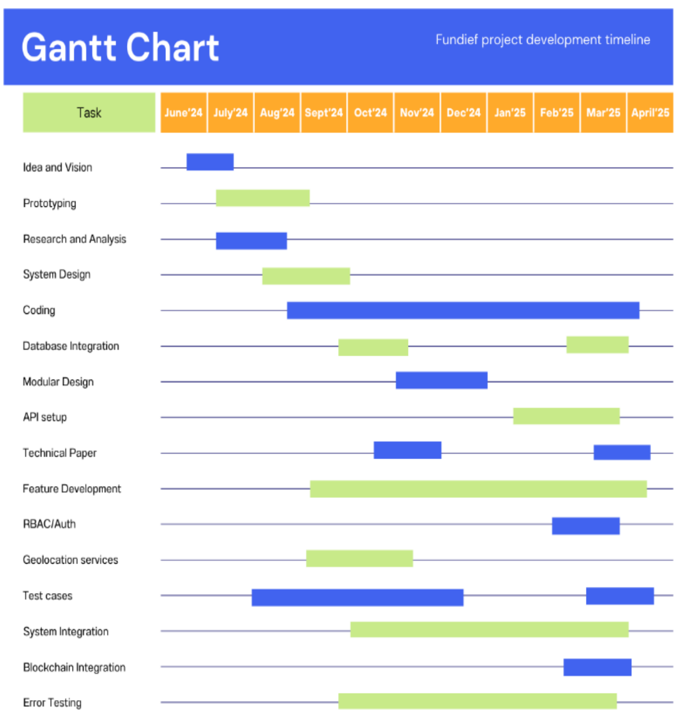
© 2025 IJIRCCE | Volume 13, Issue 4, April 2025 |DOI:10.15680/IJIRCCE.2025.1304275
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Spiral Model
The requirements phase for each incremental iteration of the spiral model consists of newer requirements, developmental strategy, integration and testing, as well as plans for the next iteration.

Incremental Model
Increment 1: Front-end development – consists of creating a base frontend that allows users to interact with the software interface. IT majorly involved creating routes, testing for i/o data, creating endpoints, and a minimal app.
Increment 2: Back-end development – consists of integrating the website frontend with APIs, databases, frameworks, a stretch towards full-stack app.
Increment 3: Authentication – consists of creating an RBAC (role-based access control) for authentication, generating restricted navigation based on permissions.
Increment 4: Feature Development – involves addition of the features like Geolocation detection, blockchain integration for transactions, etc.

Roles and Responsibilities
Project Manager – coordinate timeline, task assignment, sprint planning, and project testing; Frontend Developer – building responsive UI, integrating APIs, implementing auth logic, and collaborating with backend developments to ensure seamless frontend visibility; Backend Developer – developing REST APIs in Flask, connecting to the database, and securing endpoints; Blockchain Developer – writing and deploying smart contracts, handling transaction records, and ensuring web3 compatibility; Database Administrator – design schema, ensure data integrity and performance, create RBAC based on user_db; QA Tester – conduct unit, integration, and usability testing; report bugs, etc.; Documentation Head – keep track of all developments and changes, perform research and analysis, and document reports and technical papers.

Timeline
The entire project development timeline is illustrated below over the course of almost a year (precise 11 months – from June 2024 to April 2025).



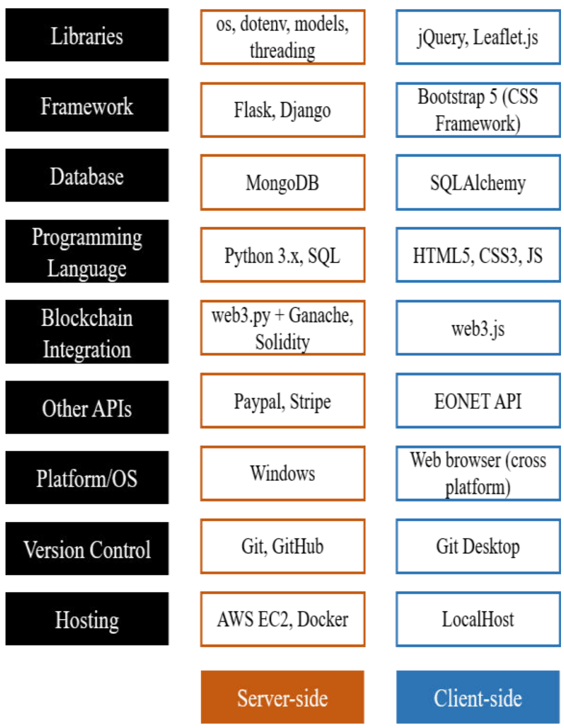


Figure 8: Fundief Development Timeline (Gantt Chart) Figure 9: Technology Stack

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025 |DOI:10.15680/IJIRCCE.2025.1304275
www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|
International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Tools and Platforms Used
The frontend is developed with HTML/CSS/JS web stack, while Python (Flask + Django) is used for the backend. Several databases (SQLAlchemy, MySQL, MongoDB) are used for project databases, as per need, as well as to create test modules in initial coding phrases. Free governmental APIs are used for geolocation tracking and blockchain development environments like Ganache (Ethereum-based) are used for blockchain integration. VS Code for development and testing use cases as well as Git for version control has been used. Designing and Modelling tools such as Canva, and PowerPoint are used to create flowcharts and visual representations. Refer to Technology Stack (Figure 9) for more detailed client-side and server-side use of technologies.

VI. TESTING STRATEGY

It is crucial to perform tests with proper strategies to understand potential vulnerabilities, system failures, dead endpoints, erroneous data flow, bugs, etc. The testing phase consisted of all team members posing as external entities to access and interact with the system. After internal testing, other individuals (classmates) were invited to navigate the entire system.

Types of Testing
The system is tested across 4 different types of testing:
- Unit testing – to verify whether the individual functions/modules are running without errors or not.
- Integration testing – to ensure whether the different modules are working together or not.
- System testing – to validate E2E system behaviour (frontend – backend – database – blockchain - APIs).
- User Acceptance testing (UAT) – a final test to confirm whether the product meets user needs and flows naturally or not. Conducted with real individuals posing as different modulated users – admin, organization, volunteer, donor users, etc.

Test Cases

Test Case ID	Case desc.	Expected output	Type of Test	Status
TC001	Login with valid credentials	Redirect to user dashboard	Unit	Pass
TC002	Register as a new organization	Verification pending screen	Integration	Pass
TC003	Submit donation to campaign	Blockchain confirmation & success alert	Integration/ System	Pass
TC004	Unauthorized access to admin panel	Access denied error	System	Pass
TC005	Volunteer requests disaster assignment	Assignment record created	Unit/ Integration	Pass
TC006	Add new campaign by verified org	Campaign listed in pending approval	Integration	Pass
TC007	Donation progress bar real-time update	Updated bar on campaign page	System	BUG!
TC008	Blockchain record for each donation	Transaction hash logged in ledger	Integration/ System	Pass
TC009	User searching for a disaster	Filtered campaigns listed	UAT	Pass

Table 3: Test Cases

Bug Tracking and Fixing
VS Code is used for localhost runtime and system audit. Blackbox.ai (VS code extension) is used for error evaluation and fixing. Postman is used for testing APIs. Browser Inspection is used for frontend logic evaluation and error detection.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Bug ID	Desc.	Severity	Fix
BUG01	Donation progress bar does not update in real-time	Moderate	Implemented WebSocket-based live updates using Flask-SocketIO to push real-time changes
BUG02	Volunteer approval not triggering	High	Corrected the route binding for the approval endpoint (POST /organization/ approve_volunteer/ <volunteer_id>)
BUG03	Campaigns Approval not showing	High	Updated query logic in the admin view function to fetch campaigns with status "pending"
BUG04	Smart contract not triggered for first-time donors	High	Corrected condition in Solidity function to trigger donate() regardless of donor history
BUG05	Campaign image not loading	Low	Verified image upload path and corrected static file serving logic in Flask by using url_for ('static', filename=...)
BUG06	Geolocation service not fetching data	Moderate	Switched from GDACS to NASA EONET API (v2.1), updated disaster_locator.py logic to parse EONET's event structure

Table 4: Bugs and Fixes

VII. RESULTS AND EVALUATION

The development results are tested against the targets set and evaluated against the outcome observed in actual project deployment and demonstration. More than 90% accuracy is expected in the early stages of development, achievable 80% is generally accepted. The platform has successfully completed the first and second phases of development (in the time frame of the last quarter of 2024 and the first quarter of 2025), the exact developmental timeline mentioned in Section (6.2). Testing across multiple users and devices has confirmed operability. The system is currently in the beta testing stage and is deployed on a local Flask server with a Ganache testnet for blockchain transactions.

A. Key Challenges Faced and Addressed

RBAC: Users with different roles (Admin, Donor, Organization, Volunteer) could accidentally access pages meant for other roles. The solution involved implementing strict role-based access control using Flask decorators. Each dashboard route was protected by @login_required and @role_required wrappers to ensure users could only access permitted areas.

SQL + NoSQL Integration: Managing data across SQLAlchemy (for structured entities like users, and donations) and MongoDB (for flexible disaster metadata and images) introduced complexity in CRUD operations. The solution involved modularized data access layers and creating unified service functions to interact with each database in isolation. Applied consistent schema validations in MongoDB using Mongoose models.

Geo-Location Service: Disaster locations were not being fetched or displayed due to issues with external API calls. For resolution, replaced the GDACS API with NASA EONET API v2.1, which offered a more reliable and structured event feed. Parsed response data effectively and integrated it with Leaflet.js to plot disaster locations on a live map.

Real-Time Donation Progress Update: The donation progress bar did not reflect live donation activity. Integrated Flask-SocketIO to enable real-time bidirectional communication between the server and client. The donation endpoint was modified to emit updates upon each successful donation, ensuring immediate UI refresh without page reloads.
Smart Contract Deployment: Initial Migrations were unsuccessful. Re-coded main.py code to successfully parse values from smart contract and corrected variable conflicts.

B. Performance Metrics

The performance of the platform was assessed based on predefined benchmarks. Below is a visual representation of the key metrics evaluated during testing.

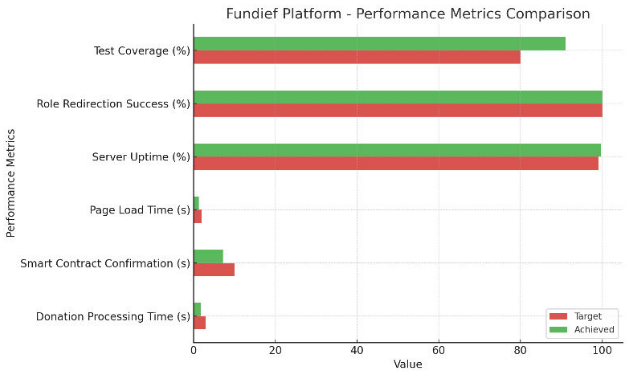


Figure 10: Performance Metrics v1

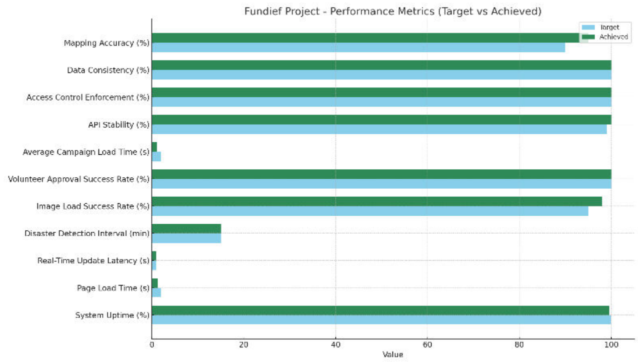


Figure 11: Performance Metrics v2

Figure 10 represents Modular Testing (when testing separate modules) and figure 11 represents Integrated/System Testing (after integrating all modules). All key performance indicators (KPIs) were met or exceeded, positioning the project well for deployment and further scaling.

VIII. FUTURE WORK

The future scope of the project lies in its expansion with different technological sectors such as IoT integration, multi-lingual capabilities for partnership with international organizations, smart contract integration for all functionalities, use of Metaverse to simulate and optimize disaster recovery plans, deeper Social Media integration, etc. IoT integration can help detect disasters on the ground level and immediately raise alerts to the authorities thus enhancing both disaster prevention and recovery. The use of Natural Language Processing (NLP) for translations to global languages can enable both native and foreign users to interact with the platform without language barriers. Furthermore, simulations and the use of Metaverse and AI technologies can help in visualizing the recovery plans and processes, and even draw out the risk areas based on historical data. Overall, Fundief has the potential to evolve into a crucial platform for both disaster relief and prevention, integrating advanced technologies and expanding its reach to create a globally impactful solution.

To expand the scope and effectiveness of the platform, several advanced features are proposed for future iterations.
NFT badges – providing NFTs to donors with a higher trustless level of donations to campaigns, thus positioning them as direct connoisseurs to be contacted in case of future disasters; **Loyalty Tokens** – Fundief will have its own token to be rewarded to the users in exchange for their level of interactivity and donation; **Cross blockchain compatibility and**

crypto payments – Enabling crypto payments across different blockchains; **Mobile Application Development** – developing a cross-platform mobile version of the Fundief app for wider accessibility; **Multilingual Support** – enabling NLP and other concepts for enabling regional or international languages for better user inclusivity;

AI-powered suggestions – using predictive analysis using machine learning to suggest similar campaigns to users, and connect volunteers/organizations/donors with similar interests.

IX. CONCLUSION

This working paper provides a comprehensive overview of the Fundief platform and an in-depth peak into the development mechanism of the project. The paper explores technologies, models, architecture, modules, and functionalities utilized for the development of the platform.

In conclusion, Fundief is an attempt to revolutionize the Donation and Relief Sector by enhancing the capabilities of existing crowdfunding platforms and newer trending technologies. This project creates an ecosystem that integrates the perks of tools such as blockchain technology, AI and ML, geo-locations, NFTs, etc., with a charitable cause of helping disaster-stricken areas, thus contributing to global disaster management. The initial phase of the project development realized foundational aspects such as system design, system architecture, endpoint creations, front-end development,

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

database connectivity, payment gateways, and disaster displays. This phase laid the groundwork for the project's visibility, functionality and navigation. Further developments include the integration of advanced features and algorithms such as filtering, live disaster detection, AI-driven recommendations, blockchain-based transparency, etc. with the help of API gateways for different functionalities. Through these innovations, Fundief addresses the key challenges faced by traditional crowdfunding platforms such as dispersed donation efforts, inefficiencies in resource allocation, and lack of transparency and trust, and brings disaster relief and management on par with modern technologies. Fundief is envisioned to be a long-term solution for disaster relief and management on a global scale.

REFERENCES

1. Federal Emergency Management Agency (FEMA). Annual Report. 2020.
2. United Nations Office for Disaster Risk Reduction (UNDRR). Global Assessment Report on Disaster Risk Reduction. 2020.
3. A. Ali, M. Haruna, and K. Rahman. "Blockchain technology in disaster management: A literature review." Journal of Information Technology in Construction. 2021.
4. Raghavendra Belur and A. Kumar. IoT and Disaster Management: Technologies, Trends, and Applications. Springer. 2020.
5. P. Gupta, A. Jain, and M. Mishra. "Applications of artificial intelligence in disaster management." Procedia Computer Science. vol. 167, pp. 1071–1080. 2020.
6. International Federation of Red Cross and Red Crescent Societies. World Disasters Report 2020. 2020.
7. S. Khan, P. Sharma, and R. Mishra. "Enhancing disaster resilience through IoT and big data analytics." In Proc. Int. Conf. Disaster Risk Reduction. 2019.
8. Melanie Mitchell. Artificial Intelligence: A Guide for Thinking Humans. Farrar, Straus and Giroux. 2019.
9. Melanie Swan. Blockchain: Blueprint for a New Economy. O'Reilly Media. 2015.
10. S. Nakamoto. "Bitcoin: A peer-to-peer electronic cash system." 2008.
11. S. Osborne and J. Anderson. "Work in Progress: Combining Real Time and Multithreading." in Proc. 39th IEEE Real-Time Systems Symposium, Dec. 2018, pp. 139–142.
12. IEEE. "IEEE Recommended Practice for Software Design Descriptions." IEEE Std 1016-2009.
13. M. Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd ed. Addison-Wesley. 2003.
14. United Nations. "Sustainable Development Goals." [Online]. Available: https://sdgs.un.org/goals
15. Leaflet. "Leaflet.js Documentation and Tutorials." Leaflet Official Docs, 2024. [Online]. Available: https://leafletjs.com/examples.html

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details