

RV COLLEGE OF ENGINEERING[®], BENGALURU-560059

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



TITLE OF THE PROJECT
CROWD FUNDING MANAGEMENT SYSTEM

Mini - Project Report

Submitted by

Mahammad Rizwan
Arun
Mallikarjun M

USN 1RV22CY036
USN 1RV22CY013
USN 1RV22CY037

in partial fulfillment for the requirement of 5th Semester
DATABASE MANAGEMENT SYSTEMS (CD252IA)

Under the Guidance of

Dr. Sneha M, Associate Professor, CSE, RVCE

RV COLLEGE OF ENGINEERING®, BENGALURU 560059
(Autonomous Institution Affiliated to VTU, Belagavi)

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**



CERTIFICATE

Certified that the project work titled '**CROWD FUNDING MANAGEMENT SYSTEM**' is carried out by **Mahammad Rizwan (1RV22CY036)**, **Arun(1RV22CY013)**, **Mallikarjun M (1RV22CY037)** who are bonafide students of R. V. College of Engineering, Bengaluru, in partial fulfillment of the curriculum requirement of 5th Semester **Database Management Systems(CD252IA)** Laboratory Mini Project during the academic year **2024-2025**. It is certified that all corrections/suggestions indicated for the internal Assessment have been incorporated in the report. The report has been approved as it satisfies the academic requirements in all respect laboratory mini-project work prescribed by the institution.

Signature of Faculty In-charge

Head of the Department
Dept. of CSE, RVCE

External Examination

Name of Examiners

Signature with date

1

2

Acknowledgement

We are indebted to our guide and faculty, **Dr. Sneha M**, Associate Professor, **Dept of CSE, RV College of Engineering** for her wholehearted support, suggestions, valuable comments and invaluable advice throughout our project work and helped in the preparation of this project.

Our sincere thanks to **Dr. Shanta Rangaswamy** Professor and Head, Department of Computer Science and Engineering, RVCE for her support and encouragement.

We express sincere gratitude to our beloved Principal, **Dr. K. N. Subramanya** for his appreciation towards this project work.

We thank all the **teaching staff and technical staff** of the Computer Science and Engineering department, RVCE for their help.

Lastly, we take this opportunity to thank our **family** members and **friends** who provided all the backup support throughout the project work

Abstract

Crowdfunding has emerged as a powerful tool for raising funds for diverse purposes, ranging from startups and creative projects to medical emergencies and social causes. However, managing the complexities of crowdfunding campaigns demands an efficient and streamlined system. This project presents the design and implementation of a **Crowdfunding Management System (CFMS)** aimed at simplifying the end-to-end process for campaign creators, contributors, and platform administrators.

Crowdfunding has revolutionized fundraising by enabling individuals, startups, and organizations to gather financial support from a broad audience. However, the complexity of managing crowdfunding campaigns necessitates a structured and efficient system. This project introduces a Crowdfunding Management System (CFMS) that streamlines the entire process, enhancing accessibility, security, and transparency for all stakeholders, including campaign creators, contributors, and platform administrators.

The CFMS provides an intuitive platform where campaign creators can effortlessly launch projects, define funding goals, and manage multimedia content while tracking real-time contributions. Contributors benefit from a user-friendly interface to explore campaigns, make secure transactions, and stay informed about project updates. The system incorporates essential features such as Payment Gateway Integration for seamless transactions, Campaign Analytics to track engagement and performance, and Reward Management for perks-based crowdfunding models.

Additionally, the CFMS emphasizes transparency and trust by ensuring secure transactions and clear communication between campaign initiators and backers. The integration of data-driven analytics and automation enhances decision-making, providing valuable insights into campaign effectiveness. This platform ultimately fosters a more efficient, reliable, and engaging crowdfunding ecosystem, empowering creators and contributors alike.

Table of Contents

	Page No.
Acknowledgement	i
Abstract	ii
Table of Contents	iii
List of Figures	iv
1. Introduction	1
1.1 Objective.....	
1.2 Scope.....	
2. Software Requirement Specification.....	3
2.1 Software Requirements.....	
2.2 Hardware Requirements.....	
2.3 Functional Requirements.....	
2.4 Non-Functional Requirements.....	
2.5 RDBMS and NoSQL Integration.....	
3. Entity Relationship Diagram.....	7
4. Detailed Design.....	11
4.1 DFD Level 0.....	
4.2 DFD Level 1.....	
5. Relational Schema and Normalization.....	15
6. Conclusion.....	19
7. References.....	20
8. Appendix: Snapshots.....	

List of Figures

FIGURE NO.	TITLE	PAGE NO.
3.0	Entity Relationship Diagram for CFMS	8
4.1	DFD Level-0 for CFMS	12
4.2	DFD Level-1 for CFMS	14
5.0	ER Schema	17

CHAPTER 1

Introduction

In today's digital economy, crowdfunding has become a powerful tool for entrepreneurs, non-profits, and innovators seeking financial support for their projects. However, managing crowdfunding campaigns effectively while ensuring transparency, compliance, and donor trust remains a significant challenge. The Crowdfunding Management System (CFMS) is designed to simplify the entire crowdfunding lifecycle—from campaign creation and fund collection to donor engagement and regulatory compliance—by leveraging structured database management, secure payments, and blockchain integration.

The innovative aspect of CFMS lies in its scalable and secure architecture, ensuring seamless fundraising operations and financial integrity. The system employs a MySQL database for structured data management, maintaining high-speed retrieval and referential integrity. Secure authentication is enforced using bcrypt encryption, protecting user credentials from unauthorized access. Additionally, JWT (JSON Web Tokens) are used for secure authentication and role-based access. After login, users receive a JWT that authorizes their actions, whether they are campaign creators, donors, or admins. This enhances security and ensures a smooth, trustworthy user experience.

A key innovation of CFMS is its blockchain integration for cryptocurrency donations, enabling users to contribute securely using Ethereum, Solidity, and MetaMask wallets. By leveraging blockchain technology, CFMS ensures transparency, immutability of transactions, and donor anonymity, eliminating fraud risks and enhancing trust. The system also supports both fiat and crypto transactions, providing users with multiple payment options, including UPI, credit/debit cards, and decentralized finance (DeFi) solutions.

Furthermore, CFMS incorporates real-time analytics and transaction tracking, allowing donors and campaign creators to monitor fundraising progress effectively. By integrating blockchain for crypto payments and ensuring a secure, transparent, and user-friendly experience, CFMS enhances the efficiency of modern crowdfunding while maintaining compliance with financial regulations.

1.1 Objectives

- **Develop a Scalable and Secure Crowdfunding Platform** – Establish a robust system that supports multiple concurrent campaigns with seamless scalability.
- **Enhance Transaction Transparency and Security** – Implement blockchain technology to provide a tamper-proof and auditable record of donations.
- **Improve Donor Trust and Engagement** – Enable real-time tracking of donations, ensuring accountability and fostering donor confidence.
- **Ensure Secure Payment Processing** – Integrate MySQL database management with encrypted transactions and multi-currency support for seamless fund collection.
- **Implement Strong Authentication and Role-Based Access** – Use bcrypt encryption and RBAC to protect user credentials and restrict unauthorized access..

1.2 Scope

- **Campaign Management** – Users can create, manage, and track crowdfunding campaigns with structured data storage.
- **Secure Fund Collection** – The platform supports multiple payment gateways, including fiat (credit/debit cards, UPI) and cryptocurrencies (Ethereum-based donations via MetaMask).
- **User Authentication and Access Control** – Secure user credentials using **bcrypt encryption** and enforce role-based access control (RBAC) to prevent unauthorized access.
- **Blockchain Integration** – Ensure transparency, security, and immutability of financial transactions through **Ethereum and Solidity-based smart contracts** for cryptocurrency donations.
- **Real-Time Analytics & Tracking** – Provide real-time insights into fundraising progress, donor engagement, and financial transactions for both campaign creators and donors.
- **Regulatory Compliance** – Ensure adherence to financial and legal regulations for transparent and ethical fundraising.

CHAPTER 2

Software Requirement Specification

1.3 Software Requirements

1. **Backend:** NODE JS, EXPRESS JS.
2. **Frontend:** HTML, CSS, JavaScript, ejs template, bootstrap.
3. **Database:** MySQL, MONGO DB (NO SQL).

1.4 Hardware Requirements

Application Server:

- **Processor:** 8-core CPU (Intel Xeon/AMD EPYC) - Powerful processing capabilities to handle complex cybersecurity data analysis and multiple concurrent user requests.
- **Memory:** 32 GB RAM minimum - Sufficient memory to support high-performance data processing, analytics, and real-time breach monitoring.
- **Storage:** 1 TB SSD with RAID configuration - High-speed, redundant storage to ensure quick data access and system reliability for extensive vulnerability records.
- **Network:** 1 Gbps Ethernet Connection - High-speed networking to support real-time data synchronization, breach tracking, and system communications.

Database Server:

- **Processor:** 8-core CPU (Intel Xeon/AMD EPYC) - Robust processing power for complex database queries and data management operations.
- **Memory:** 64 GB RAM - Enhanced memory capacity to support advanced data analytics, complex reporting, and efficient data retrieval.
- **Storage:** 2 TB RAID-configured SSDs - Extensive, redundant storage to accommodate growing cybersecurity records with high-speed data access.
- **Network:** 10 Gbps Ethernet - Advanced networking capabilities to ensure seamless communication and minimal latency in data transmission.

1.5 Functional Requirements

1. User Features

- Users can view all ongoing campaigns on the landing page.
- Users can click on a specific campaign to view detailed information (e.g., title, goal, raised amount, description).
- Users can support campaigns by donating a specified amount via PayTM.
- Users can view transaction details (amount and payment confirmation) for each donation on the respective campaign page (anonymity is maintained for donors).
- User can perform CRUD operations on campaigns created by him.
- Users can submit queries via a form provided in the 'Contact Us' section.

2. Admin Features:

- Approve or Reject Campaigns: Admins can accept or reject user-created campaigns for valid reasons.
- Deactivate Campaigns: Admins can deactivate campaigns once the goal is achieved or for other valid reasons.
- Manage Users: Admins can view all registered users and have the ability to delete or block users if necessary.
- Monitor Donations: Admins can view all donations made by users, including the amount, transaction details, and associated campaign.
- Transaction Oversight: Admins can track all financial transactions related to donations.
- Add New Admins: Admins have the authority to add other admins to manage the platform.

3. Eligibility Verification :

- Validate users' eligibility for certain campaign benefits based on predefined criteria (e.g., donation history, participation level, or referrals).

4. Audit and Logs:

- Maintain detailed logs of all admin and user actions (e.g., campaign creation, donations).

- Provide visibility of all campaign activities for internal auditing.

5. **Blockchain Integration :**

- **Crypto Payments:** Integrated Ethereum-based crypto payments, allowing users to donate using Ether (ETH).
 - **Smart Contracts with Solidity:** Utilized Solidity smart contracts to handle secure and transparent donation transactions on the Ethereum blockchain.
 - **MetaMask Integration:** Enabled seamless crypto donations through MetaMask, providing users with a familiar and secure method to contribute directly from their digital wallets.
 - **User Anonymity:** Ensured donor anonymity by storing only essential transaction details (wallet address and amount) without linking to personal information, maintaining privacy and security.
- This blockchain integration enhances transparency, security, and privacy while providing a decentralized payment option for supporting campaigns.

6. **Real-Time Status Tracking :**

- Enable users to track campaign progress, including:
 - The total amount raised.
 - Their donation details and contribution status.
 - The overall status of the campaign (Active, Completed).

1.6 Non-Functional Requirements

1.Scalability

- The platform must efficiently handle a high volume of users, campaigns, and transactions.
- Design to scale horizontally for increased demand.

2.Performance:

- Ensure low-latency operations, especially during campaign participation and payment processing.
- Optimize database queries for real-time transaction updates.

3.Security:

- Encrypt all sensitive user data (e.g., payment details, email addresses).
- Use secure protocols (e.g., HTTPS) and authentication mechanisms (e.g., OAuth2, JWT) to prevent unauthorized access.
- Regularly monitor and patch vulnerabilities to safeguard the system.

4.Usability:

- Provide a user-friendly and intuitive interface for both users and admins.
- Ensure responsive design for seamless operation across devices (desktop, tablet, mobile).

1.6 RDBMS and NoSQL Integration

While the Crowdfunding System relies primarily on SQL for structured data, certain types of unstructured data that may vary in format—such as user comments, engagement content, and organizer-generated content—could benefit from a NoSQL solution. However, in this system, we have designed SQL tables to handle even multimedia content (such as event images and videos), ensuring consistent data structure across the platform.

SQL efficiently manages structured data while ensuring referential integrity and high retrieval speeds. For handling certain unstructured data, such as cryptocurrency donations, additional flexibility can be achieved using JSON fields in SQL.

For crypto donations, a NoSQL (MongoDB) schema is used to store transaction details separately:

The crypto donation schema defines how cryptocurrency donation data is stored in the database:

- `donor_address` → Stores the donor's crypto wallet address (required).
- `amount_in_ether` → Stores the donation amount in Ethereum with high precision (required).
- `transaction_hash` → Stores a unique transaction ID to track the payment (required and must be unique).
- `campaign_title` → Stores the name of the campaign the donation is made for (required).
- `created_at` → Automatically records the date and time when the donation was made

By combining SQL for structured data (campaigns, users, fiat donations) and NoSQL (MongoDB) for flexible, unstructured crypto donation data, the platform ensures efficient data management, integrity, and scalability

CHAPTER 3

Entity Relationship Diagram (ERD)

The **Entity Relationship Diagram (ERD)** for the **Crowdfunding Management System (CFMS)** represents the structural framework of the database, outlining the relationships between key entities involved in campaign management, donor engagement, fund distribution, and compliance tracking. It serves as a **foundational blueprint** for efficient data management, ensuring seamless interaction between campaign organizers, donors, payment transactions, regulatory policies, reports, and platform administrators.

This ERD is designed to ensure an organized and systematic approach to crowdfunding management, enabling the platform to:

- Facilitate the creation, tracking, and management of crowdfunding campaigns.
- Record and process donor contributions securely.
- Assign platform administrators to oversee compliance and fund distribution.
- Maintain detailed transaction records and ensure transparency in fund allocation.
- Generate analytical reports to support decision-making and regulatory compliance

By visually representing the relationships between these entities in **Fig. 3.0**, the **ERD provides a structured approach** to managing fundraising campaigns, optimizing financial tracking, and ensuring trust and transparency in the crowdfunding ecosystem. This structured design supports a **scalable and efficient crowdfunding platform**, enhancing the ability of individuals and organizations to successfully raise and manage funds for various causes.

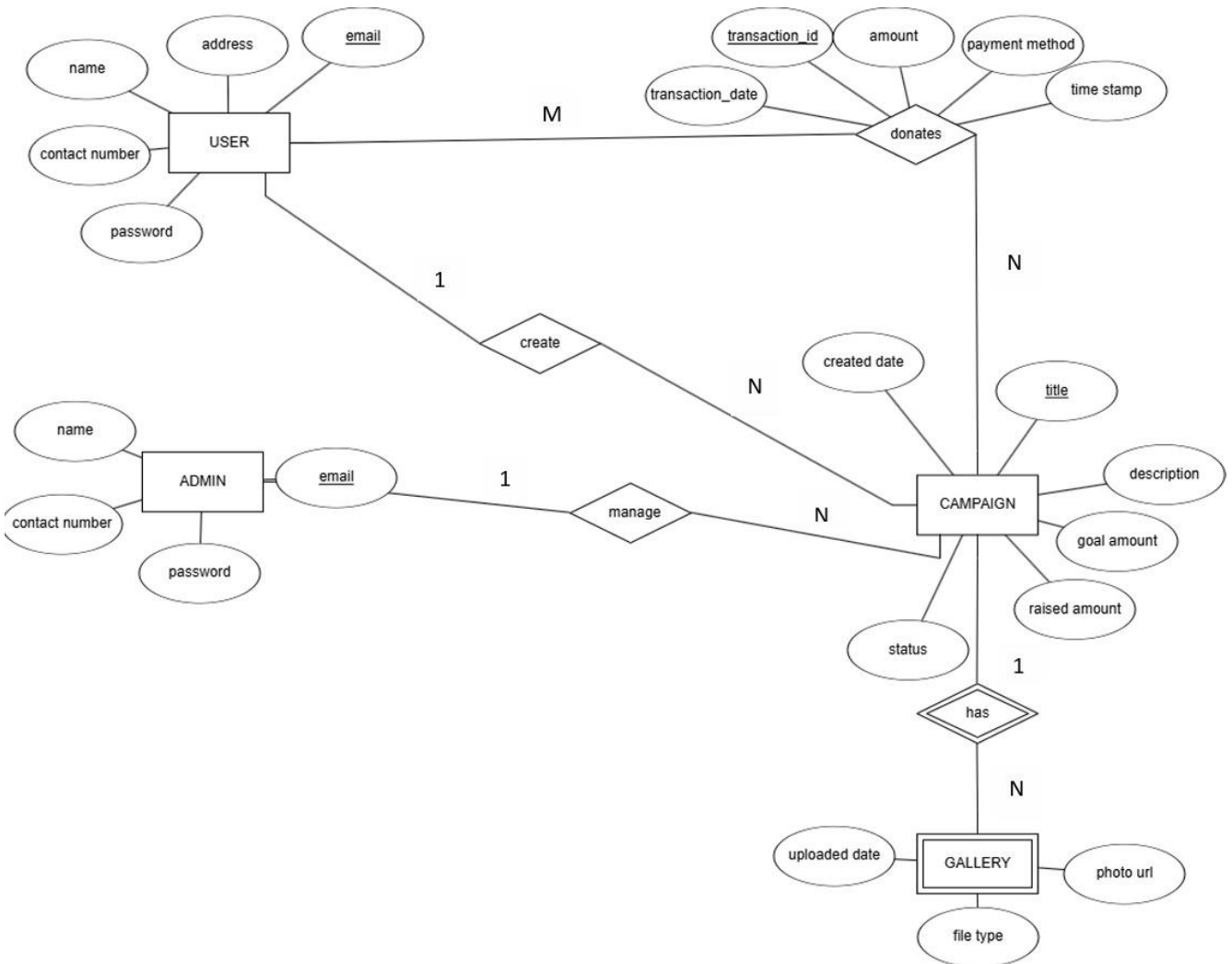


Fig. 3.0. Entity Relation Diagram for CFMS

Entities and Their Attributes

1. ADMIN

Attributes:

- name: Name of the admin.
- email: Email address (Primary Key).
- Contact Number: Contact number .
- password: Password .

2. USER

Attributes:

- name: Name of the user.
- email: Email address (Primary Key).
- Contact Number: Contact number .
- password: Password .

3. CAMPAIGN

Attributes:

- title: Title of the campaign (Primary Key, String).
- description: Description of the campaign (Text).
- Goal Amount: Goal amount for the campaign (Decimal).
- Raised Amount: Amount raised so far (Decimal).
- Created Date: Date the campaign was created (Timestamp).
- status: Status of the campaign (ENUM: 'hold', 'ongoing', 'completed')

4. Gallery

Attributes:

- Campaign Title: Title of the campaign the photo belongs to (Foreign Key, String, references campaign_title).
- Photo Url: URL to the photo (String).
- Upload Date: Date the photo was uploaded (Timestamp).

Relationship Entity

1. DONATION

- transaction_id: Unique transaction ID (Primary Key, String).
- campaign_title: Title of the campaign (Foreign Key, String, references campaign.title).
- donar_email: Email of the donor (Foreign Key, String, references users.email).
- donar_name: Name of the donor (String).
- amount: Amount donated (Decimal).
- payment_method: Payment method (ENUM: 'upi', 'qr', 'card').
- timestamp: Date and time of the donation (Timestamp).

Relationships

1. **USER** creates **CAMPAIGN**

- A **USER** can create one or more (**1:N**) campaigns.
- A **CAMPAIGN** is created by exactly one (**1**) user).

2. **USER** donates to **CAMPAIGN**

- A **USER** can make multiple donations (**M:N**) to different campaigns.
- A **CAMPAIGN** can receive multiple donations from different users.
- This is implemented using the **DONATES** relationship, which includes attributes like `transaction_id` and `amount`.

3. **ADMIN** manages **CAMPAIGN**

- An **ADMIN** can manage multiple campaigns (**1:N**).
- Each **CAMPAIGN** is managed by exactly one admin (**1**).

4. **CAMPAIGN** has **GALLERY**

- **CAMPAIGN** can have multiple (**1:N**) multimedia files associated with it in the **GALLERY**.
- Each file in the **GALLERY** belongs to exactly one (**1**) campaign.

CHAPTER 4

Detailed Design

Preamble

The **Data Flow Diagram (DFD)** for the **Crowdfunding Management System (CFMS)** illustrates how data flows between various components, including campaign creators, donors, administrators, and the underlying database and processes. It serves as a **visual representation** of data movement within the system, offering insights into how information is processed, stored, and utilized to manage crowdfunding campaigns efficiently.

This DFD helps in understanding:

- How campaign creators, donors, and administrators interact with the system.
- The processes involved in user registration, campaign creation, donation processing, and fund disbursement.
- The flow of data between external entities (users and donors), internal processes, and databases.
- How different components contribute to transparency, security, and successful campaign management.

The DFD is structured into different levels to **progressively detail** the internal workings of the system.

4.1 DFD Level 0

The **DFD Level-0** fig 4.1 in the provided diagram for the **Crowdfunding Management System (CFMS)** offers a high-level representation of the system. It shows the main entities, the system as a central process, and the key data flows. Here's the explanation:

1.User (External Entity))

- Inputs: Provides login credentials to access the system and Submits information to register, create campaigns, or donate to active campaign
- Output: Receives system responses, such as login success or failure, campaign status, and donation receipts.

2. Admin (External Entity)

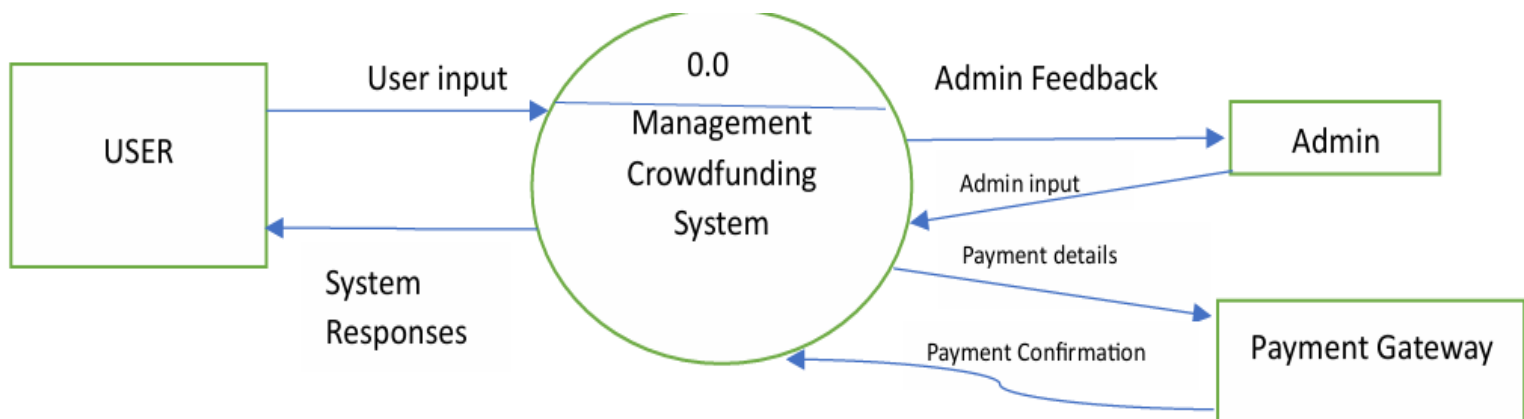
- Inputs: Submits administrative requests to manage campaigns, approve or reject campaign submissions, and oversee donations
- Outputs: Receives system responses, such as campaign approval status, system reports, or transaction summaries.

3. Payment Gateway (External Entity)

- Input : Processes login requests from cyber professionals and administrators.
Handles breach data processing and reporting functions.
- Output :Sends payment confirmation or failure status back to the system for further action.

4. Manage Crowdfunding System (Central Process)

- Functionality: Acts as the central hub for managing all data flows and interactions.
- Processes:
 - Validates user login credentials and admin access.
 - Manages campaign creation, updates, and approval workflows.
 - Tracks donation transactions and interacts with the payment gateway for secure payment processing.
- Output: Sends relevant feedback and updates to users and administrators based on their requests and system operations.



4.1 DFD Level-0 for CFMS

4.2 DFD Level 1

The DFD Level-1 in fig 4.2 expands on Level-0 by detailing the internal processes of the CFMS. It breaks the main system into four core processes:

1.0 User Management

- Users (including donors) register or log in to the system.
- Login details are verified using the User Database.
- Upon successful login, users gain access; otherwise, the login is denied.
- Data Flow:

Input: User submits registration or login details.

Output: The system verifies the credentials by checking against the User Database and provides a login response (approved or rejected).

Storage: User data is stored and retrieved from the User Database.

2.0 Donation Processing

- Users donate funds or confirm donations.
- The system interacts with the Payment Gateway to process the payment and ensure payment confirmation.
- Donation records, including payment details, are stored in the Donation Records database.
- Data Flow:

Input: User submits donation details.

Output: Payment confirmation details are processed via the Payment Gateway and stored in Donation Records.

Storage: All donation records are maintained for tracking and reporting.

3.0 Admin Functions

- Admins review and approve or reject campaigns submitted by users.
- Approval decisions are stored and used to update the campaign status.
- Data Flow:
 - Input: Admin reviews campaigns submitted by users.
 - Output: Approval or rejection response is provided to the user.
 - Storage: Admin decisions and related campaign statuses are saved in the Campaign Database.

4.0 Campaign Management:

- Users create or update campaign details for fundraising initiatives.
- Campaign statuses and details are retrieved or updated as required.
- Data flow:

Input: User submits campaign details or requests campaign status.

Output: Campaign information is stored, updated, or displayed.

Storage: Campaign details are saved in the Campaign Database.

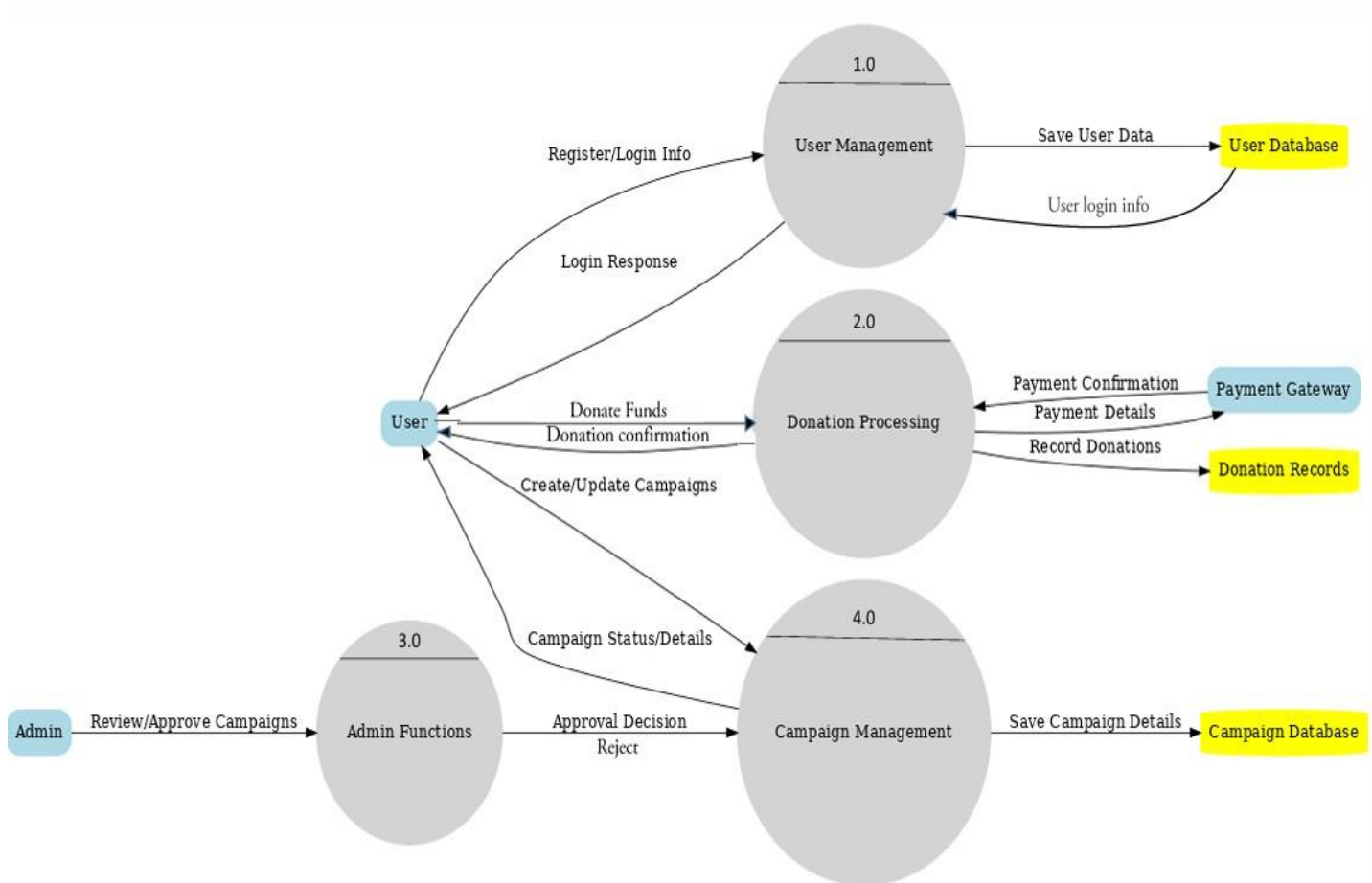


Fig. 4.2. DFD Level-1 for CFMS

CHAPTER 5

Relational Schema and Normalization

The Relational Schema for the Crowd Funding Management System (CFMS) is designed based on the Entity-Relationship (ER) Model to structure data efficiently. This schema defines the tables, attributes, and relationships among various entities like departments, systems, cyber professionals, breaches, compliance policies, and reports. The goal of this schema, fig 5.0, is to ensure data consistency, eliminate redundancy, and improve database efficiency.

Relational Schema

The following tables represent the normalized relational schema for CFMS:

1. USER

- Email (Primary Key)
- Name
- Contact_no
- Password
- Address

2. ADMIN

- Email (Primary Key)
- Name
- Contact_no
- Password

3. CAMPAIGN

- Title (Primary Key)
- Description
- Goal_amount
- Raised_amount
- Created_date
- Status
- User_email (Foreign Key → USER)

4. GALLERY

- File_type
- C_title (Foreign Key → CAMPAIGN)
- Photo_url
- Uploaded_date.

5. DONATION

- Transaction_id (Primary Key)
- C_title (Foreign Key → CAMPAIGN)
- Donor_email (Foreign Key → USER)
- Donor_name
- Amount
- Payment_method
- Time_stamp

Schema Diagram

A schema diagram visually represents the structure of the database (tables, attributes, and relationships).

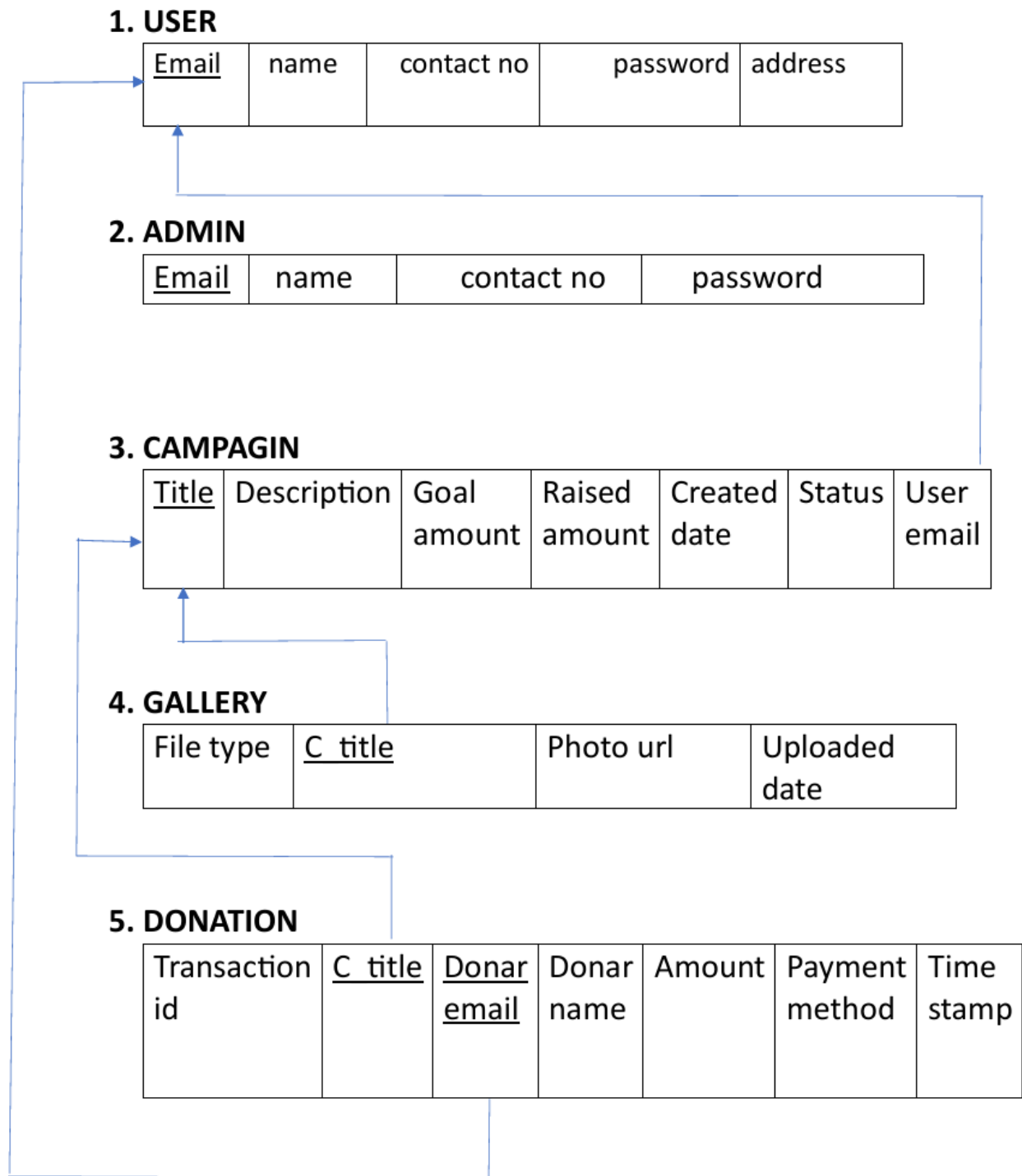


Fig 5.0. ER Schema for CFMS

Normalization Process

The CFMS database schema is already in **First Normal Form (1NF)**, **Second Normal Form (2NF)**, and **Third Normal Form (3NF)**.

1. First Normal Form (1NF)

Criteria for 1NF:

- All attributes contain **atomic values** (no multi-valued or composite attributes).
- **Primary keys** are defined for each table.
- No **repeating groups** in any table.

Applying 1NF to the Schema:

- Each column contains a **single** value, not multiple values in one field.
- Example: The **DONATION** table does not store multiple donation amounts in a single row.

Since all tables have atomic values, defined primary keys, and no repeating groups, the schema is in 1NF.

2. Second Normal Form (2NF)

- Meets 1NF requirements
- No partial dependencies (Each non-key attribute is fully dependent on the primary key)
 - There are no partial dependencies because all non-key attributes are fully dependent on the entire primary key.
 - Example:
 - In the **DONATION** table, **C_title** is already a **foreign key** referring to **CAMPAIGN** instead of repeating campaign data.
 - The **GALLERY** table stores only **one campaign per row**, avoiding partial dependencies.

Since there are no partial dependencies, the schema is in 2NF.

3. Third Normal Form (3NF)

- Meets 2NF requirements
- No transitive dependencies
 - Each non-key attribute is directly dependent on the primary key and not dependent on another non-key attribute.
 - Example:
 - The **CAMPAIGN** table stores **User_email** as a **foreign key**, linking it to the **USER** table instead of repeating user details.
 - The **DONATION** table avoids storing redundant user details by referring to **Donor_email** instead of keeping donor names repeatedly.

Since all non-key attributes directly depend on the primary key without transitive dependencies, the schema is in 3NF.

Conclusion

The Crowdfunding Management System (CFMS) represents a groundbreaking solution tailored for the modern digital economy, addressing critical challenges such as transparency, compliance, and donor trust in crowdfunding campaigns. By integrating cutting-edge technologies and emphasizing scalability, security, and user-friendliness, CFMS redefines how crowdfunding campaigns are managed and executed.

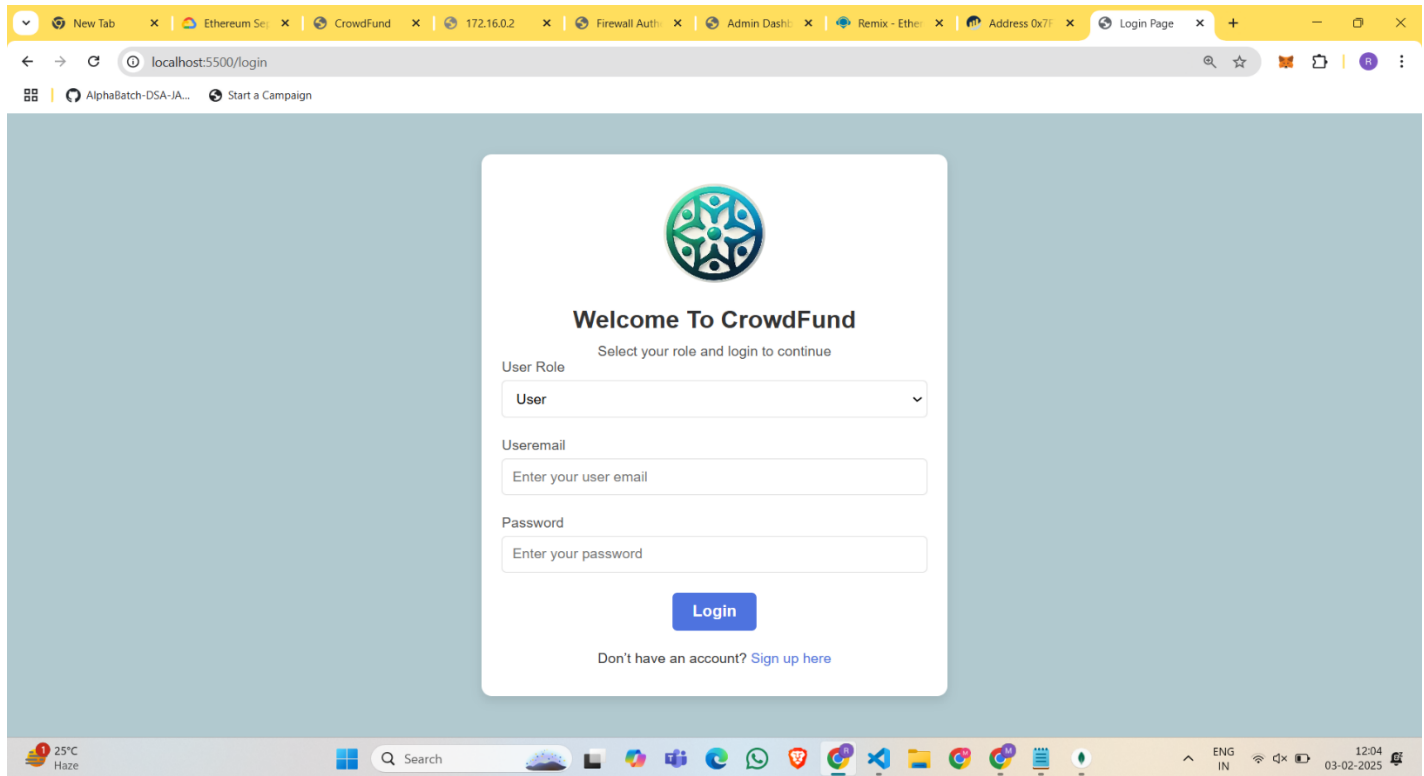
One of the core strengths of CFMS lies in its robust architecture, underpinned by structured database management through MySQL. This ensures fast, reliable data retrieval while maintaining the referential integrity necessary for seamless operations. Security is further reinforced with bcrypt encryption, which safeguards user credentials, and Role-Based Access Control (RBAC), which ensures that only authorized users have access to sensitive financial and operational functionalities. These measures collectively provide a strong foundation for operational efficiency and trust.

The system's integration with blockchain technology elevates it to a new level of innovation, particularly in the realm of cryptocurrency donations. By leveraging Ethereum, Solidity, and MetaMask wallets, CFMS ensures secure, immutable, and transparent transactions. Blockchain integration not only eliminates fraud risks but also upholds donor anonymity, a critical factor in fostering trust among contributors. The dual support for fiat and cryptocurrency transactions enhances accessibility, making CFMS adaptable to a diverse range of users and payment preferences, from UPI and credit/debit cards to decentralized finance (DeFi) solutions.

CFMS's real-time analytics and transaction tracking features further empower users by providing live insights into fundraising progress. Campaign creators and donors alike can monitor transactions and campaign milestones, fostering greater engagement and accountability. This focus on transparency and data-driven insights enhances user confidence and campaign effectiveness.

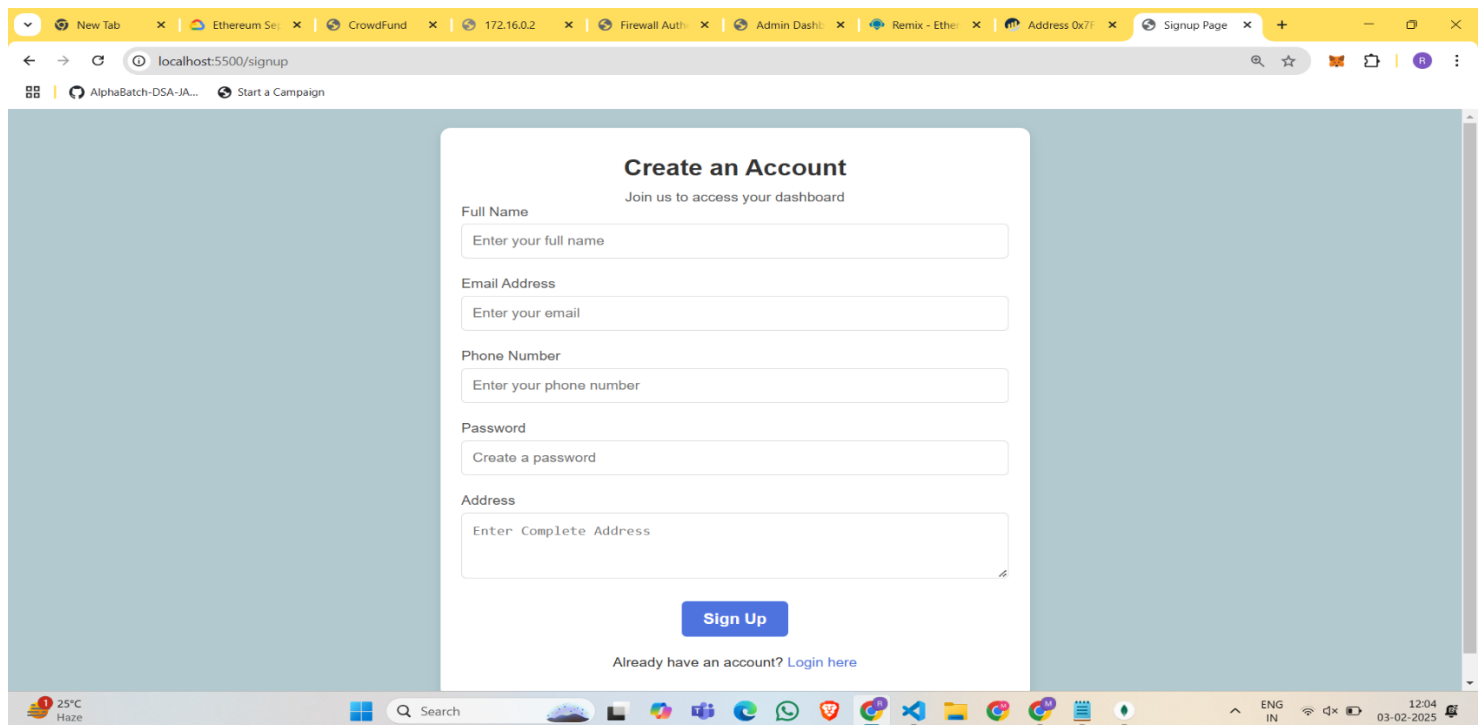
References

A. Appendix: Snapshots



The screenshot shows a web browser window with the URL `localhost:5500/login`. The page features a central white login card on a light blue background. At the top of the card is a circular logo with a green and blue geometric design. Below the logo, the text "Welcome To CrowdFund" is displayed, followed by the instruction "Select your role and login to continue". The form includes a "User Role" dropdown menu with "User" selected, a "Useremail" text input field with the placeholder "Enter your user email", and a "Password" text input field with the placeholder "Enter your password". A blue "Login" button is positioned below the password field. At the bottom of the card, there is a link: "Don't have an account? [Sign up here](#)". The browser's taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system tray on the right indicates the date and time as "03-02-2025 12:04".

Fig A.1 Role based Login Page of CFMS



The screenshot shows a web browser window with the URL `localhost:5500/signup`. The page features a central white sign-up card on a light blue background. At the top of the card, the text "Create an Account" is displayed, followed by the instruction "Join us to access your dashboard". The form includes five text input fields: "Full Name" (placeholder: "Enter your full name"), "Email Address" (placeholder: "Enter your email"), "Phone Number" (placeholder: "Enter your phone number"), "Password" (placeholder: "Create a password"), and "Address" (placeholder: "Enter Complete Address"). A blue "Sign Up" button is located below the address field. At the bottom of the card, there is a link: "Already have an account? [Login here](#)". The browser's taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system tray on the right indicates the date and time as "03-02-2025 12:04".

Fig A.2 SignUp page of CFMS

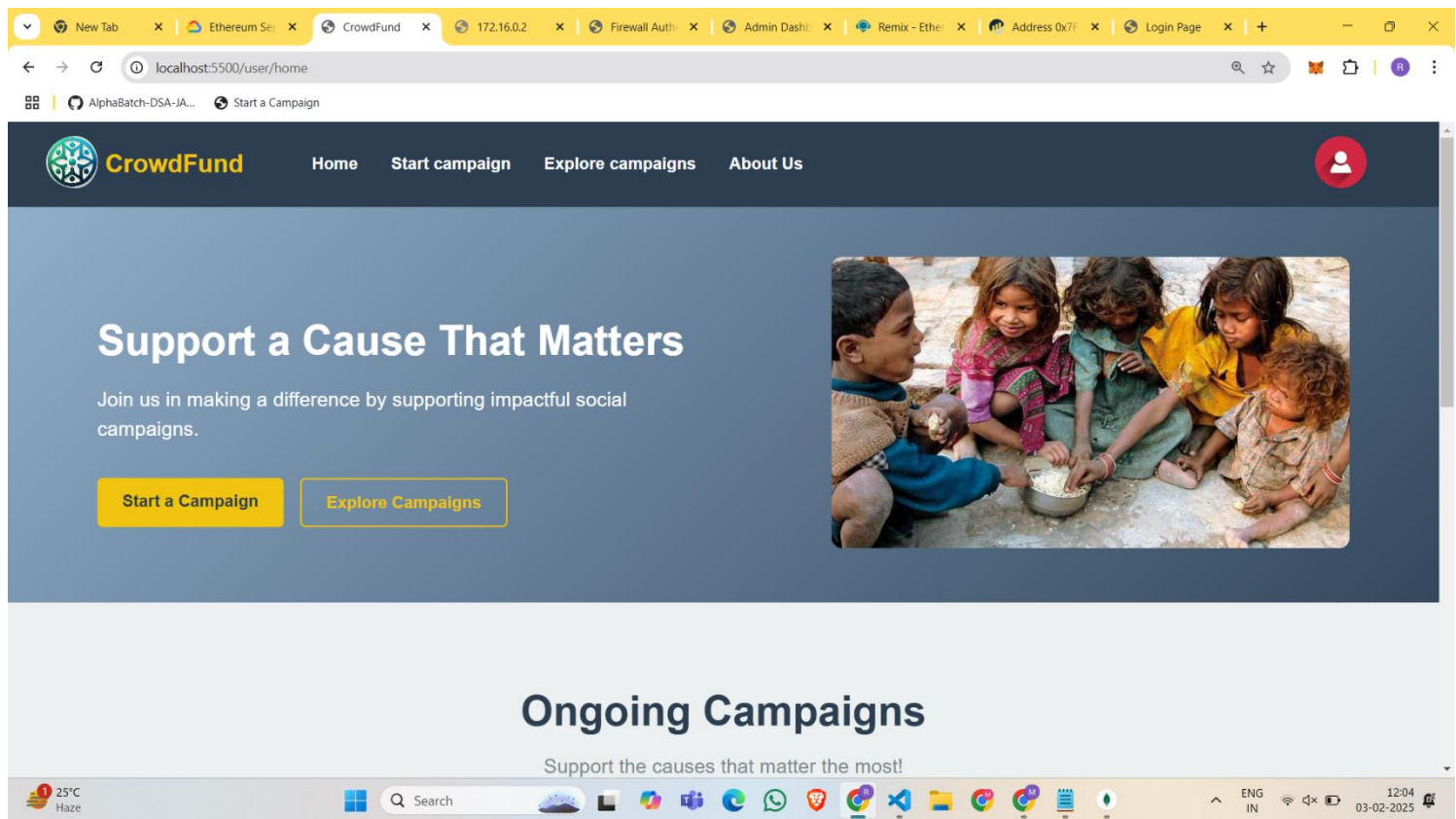


Fig A.3 User Home Page

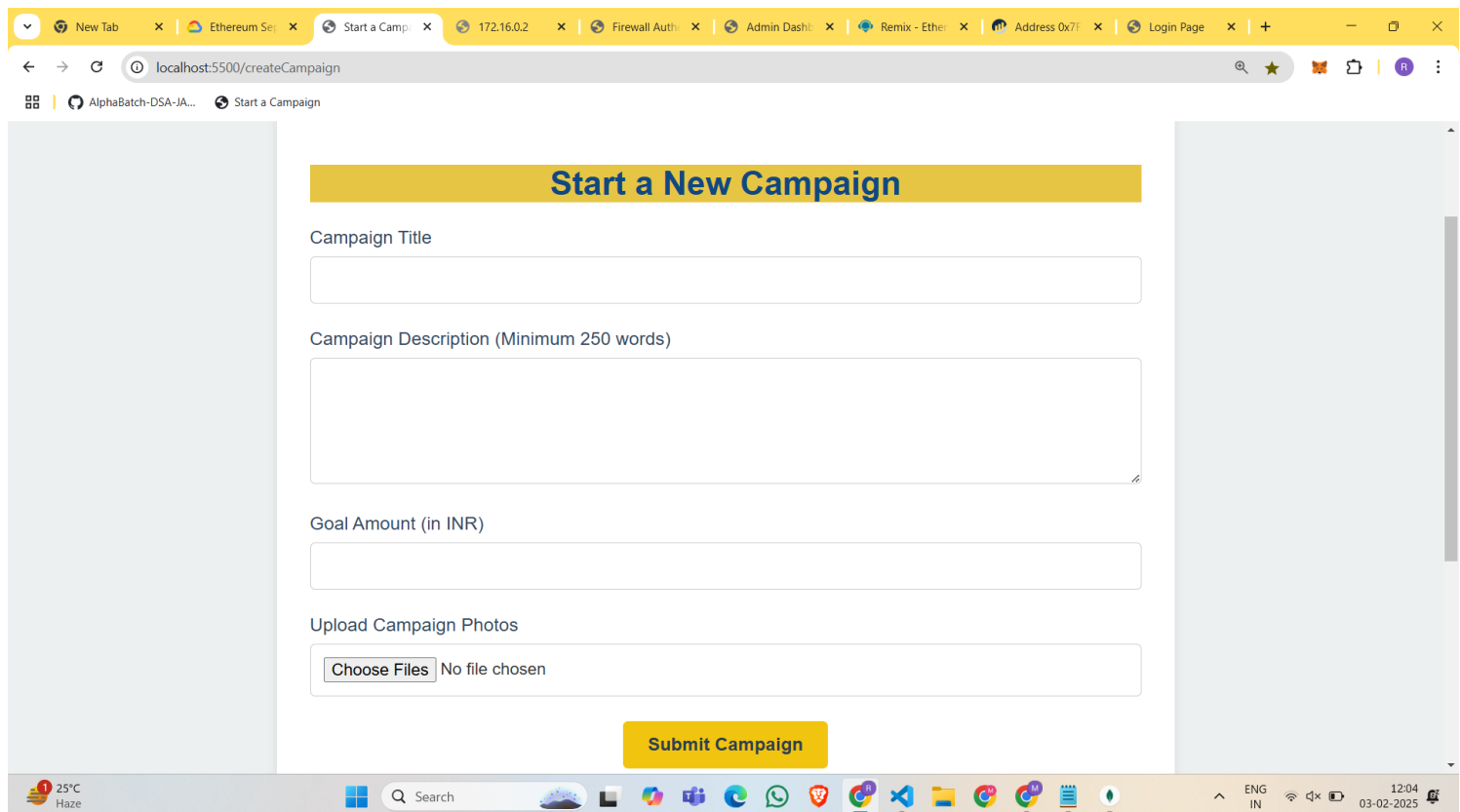


Fig A.4 User creating a new campaign in CFMS

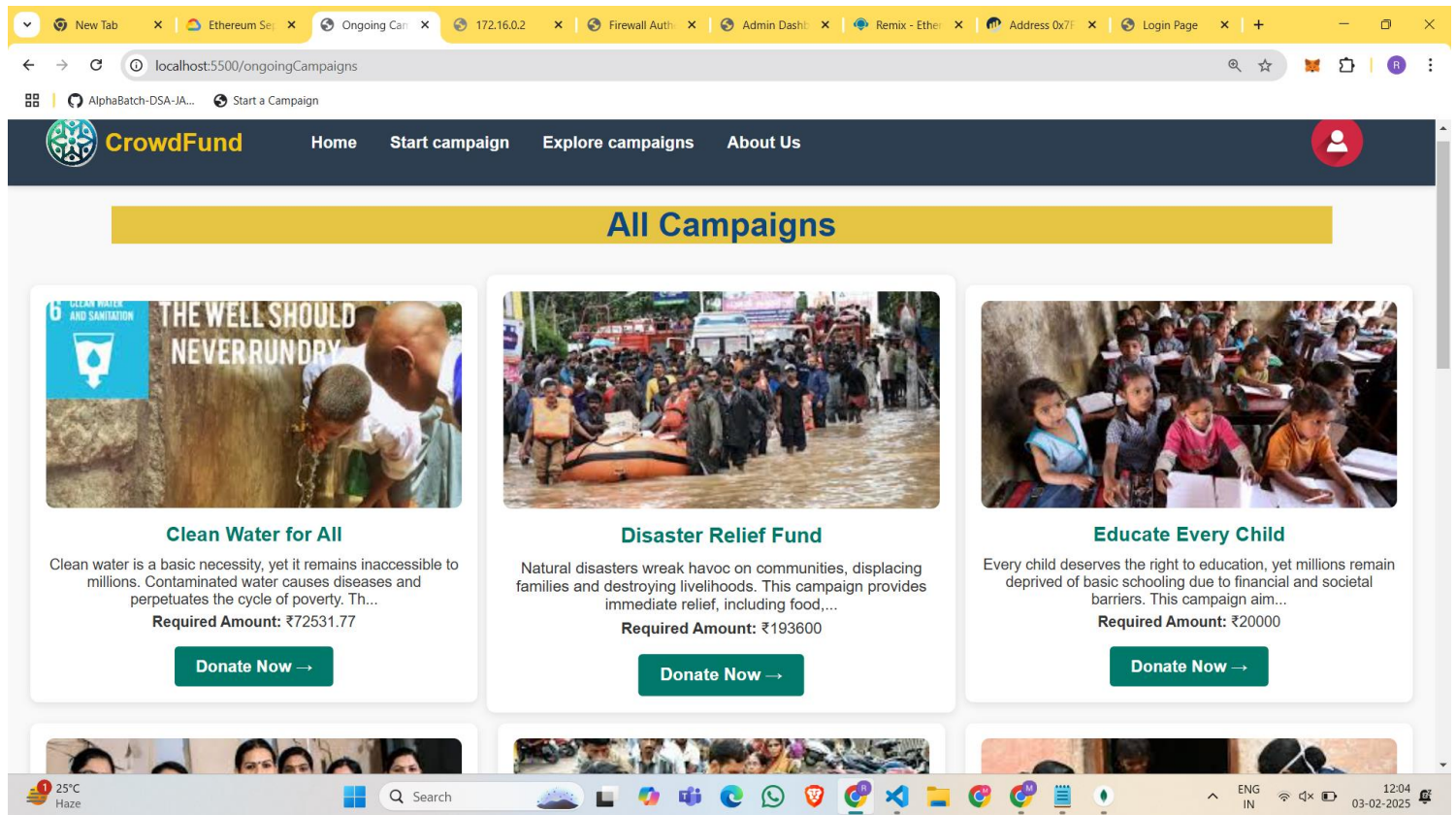


Fig A.5 User dashboard showing all Active Campaigns

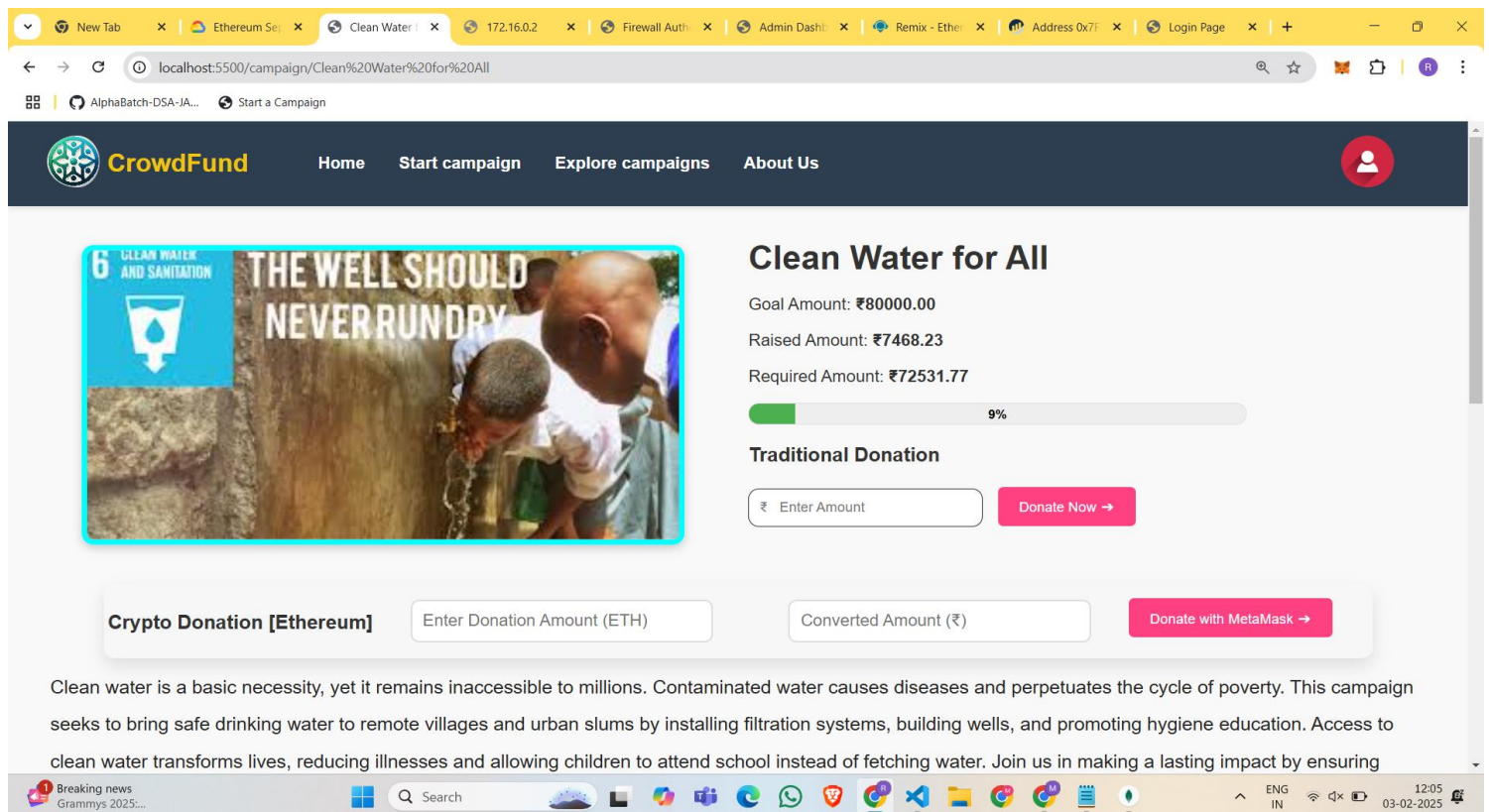


Fig A.6 Particular Campaign Page Overview with donation sections

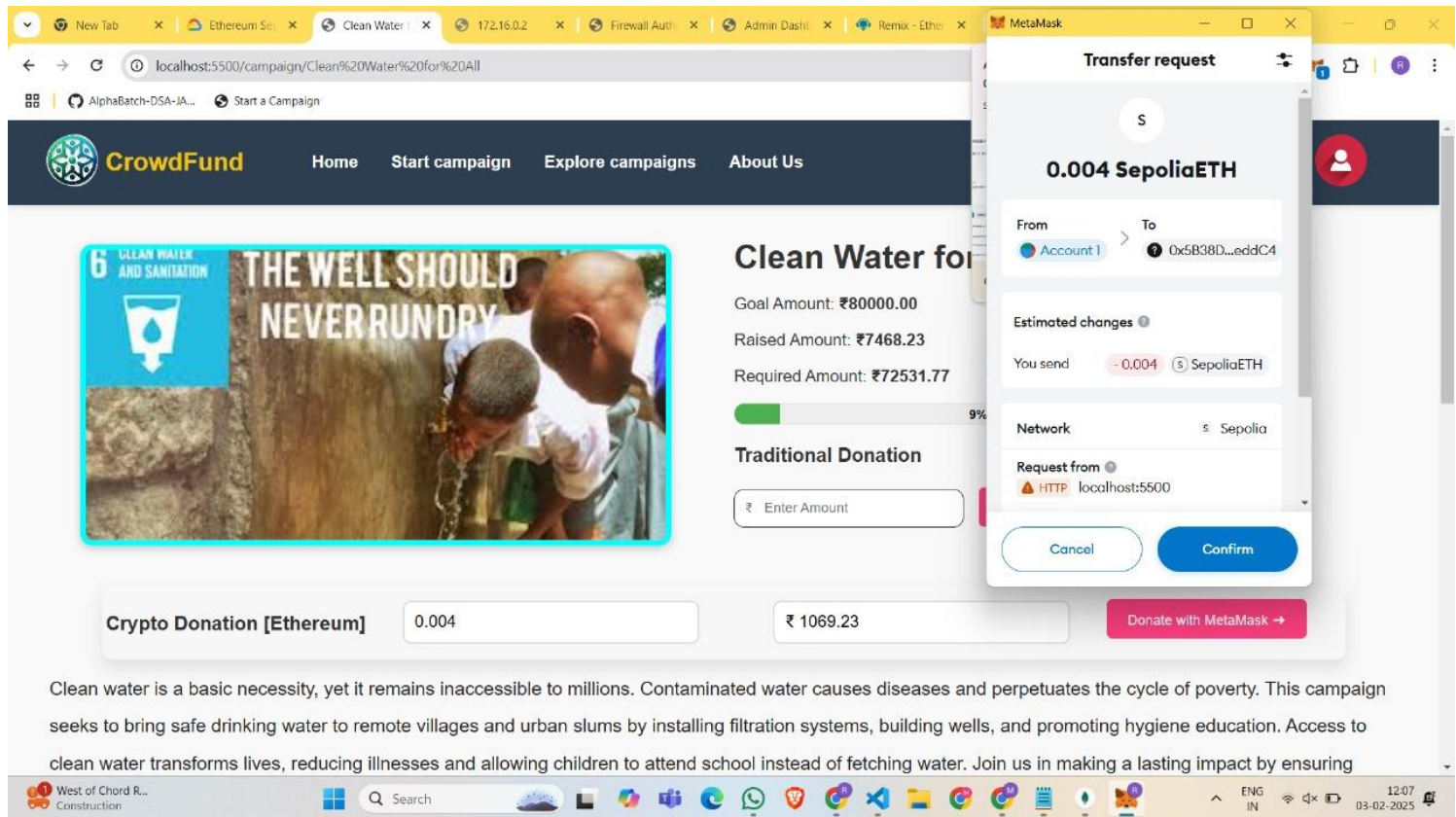


Fig A.7 Crypto Donation Process with Mata mask wallet

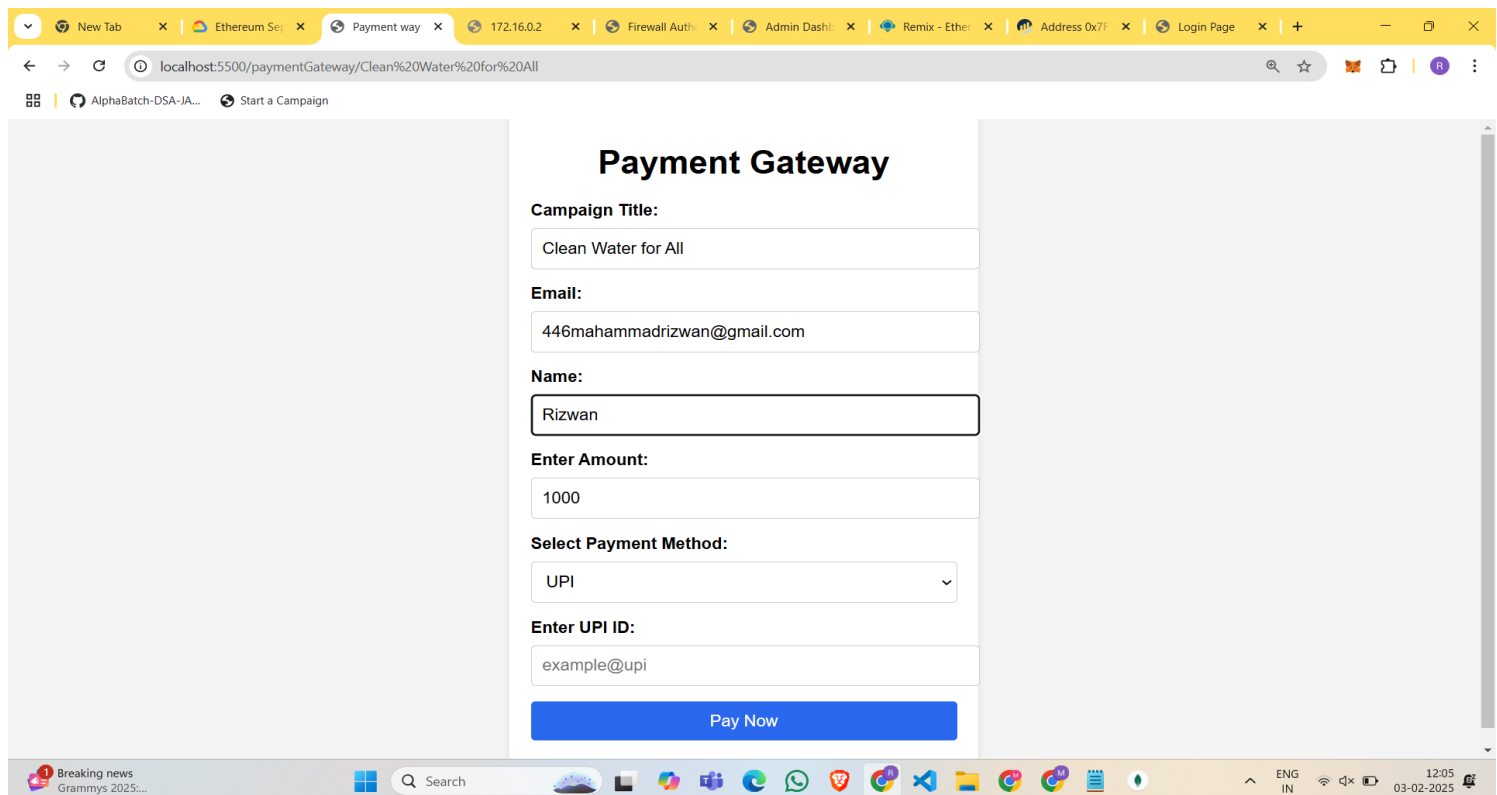


Fig A.8 Traditional Donation Process with payment gateway

Personal Information

Name: mahammad rizwan
Email: 446mahammadrizwan@gmail.com
Address: Deogiri, Sandur(T), Bellary(D), Karnataka - 583112
Phone: 09483384972

Your Donations

Transaction ID	Campaign Title	Donated with Name	Amount (₹)	Payment Method	Timestamp
0d79e06c-5c5f-4f4c-98fe-a7b4b43a628a	Feed the Hungry	Mehraj	₹100.00	qr	24/1/2025, 3:37:21 pm
16f6140b-7cda-4656-ab3b-b414fc8b353e	Feed the Hungry	Faizal	₹600.00	qr	24/1/2025, 10:18:26 am
1ae63f1e-5a47-416b-b4df-489baad35c46	Clean Water for All	Aleem	₹1000.00	upi	2/2/2025, 9:07:52 pm

Fig A.9 My Profile Page with user and his donation details

Admin Dashboard

Waiting List Campaigns
View and approve campaigns that are awaiting approval.
[View Campaigns](#)

Edit Ongoing Campaigns
Manage and edit ongoing campaigns.
[Edit Campaigns](#)

User Management
View and manage users. Delete if necessary.
[Manage Users](#)

Recent Traditional Donations
View the most recent donations made to campaigns.
[View Donations](#)

Fig A.10 Admin Dashboard

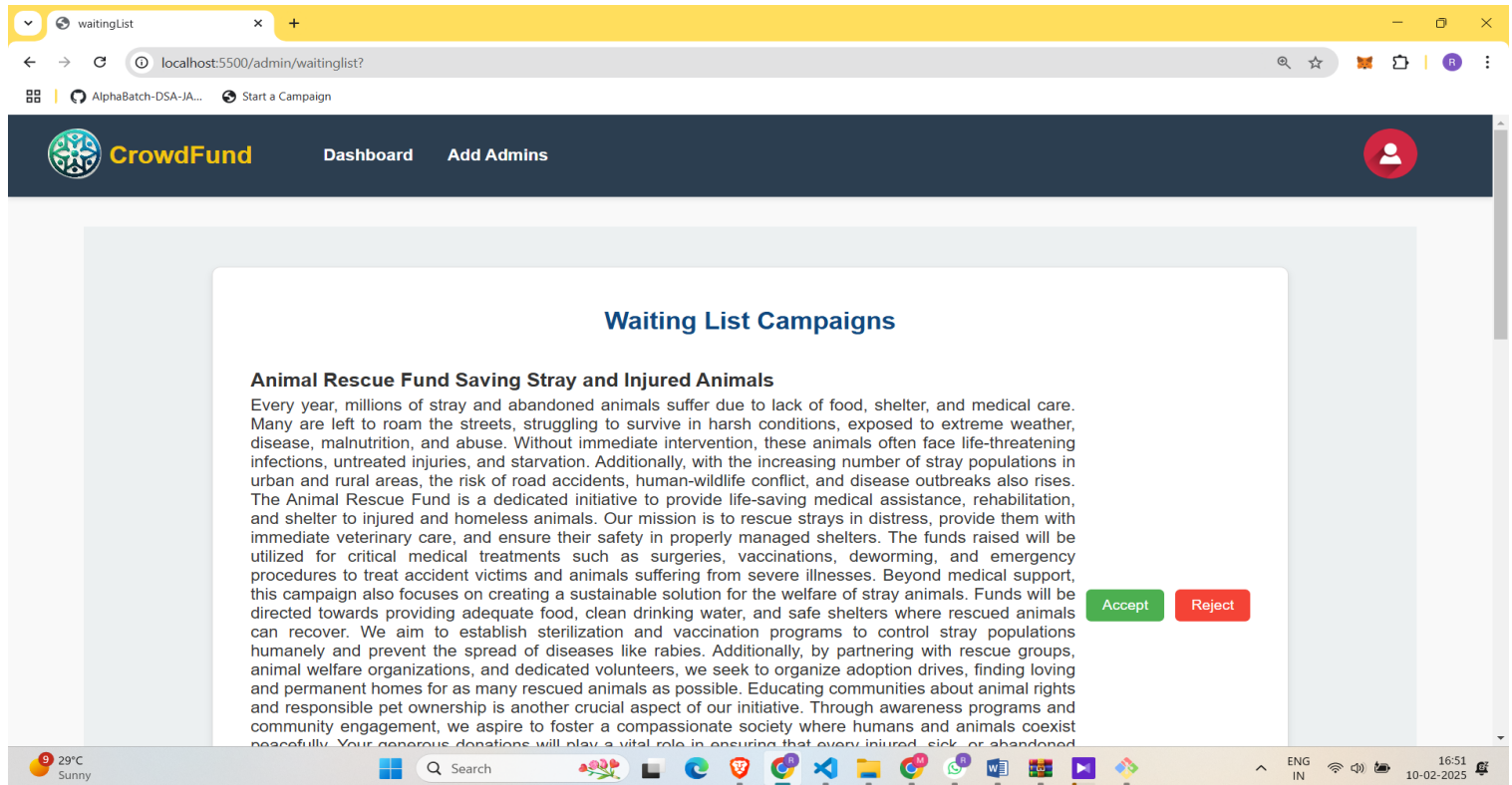


Fig A.11 Campaign Approval/Rejection Section

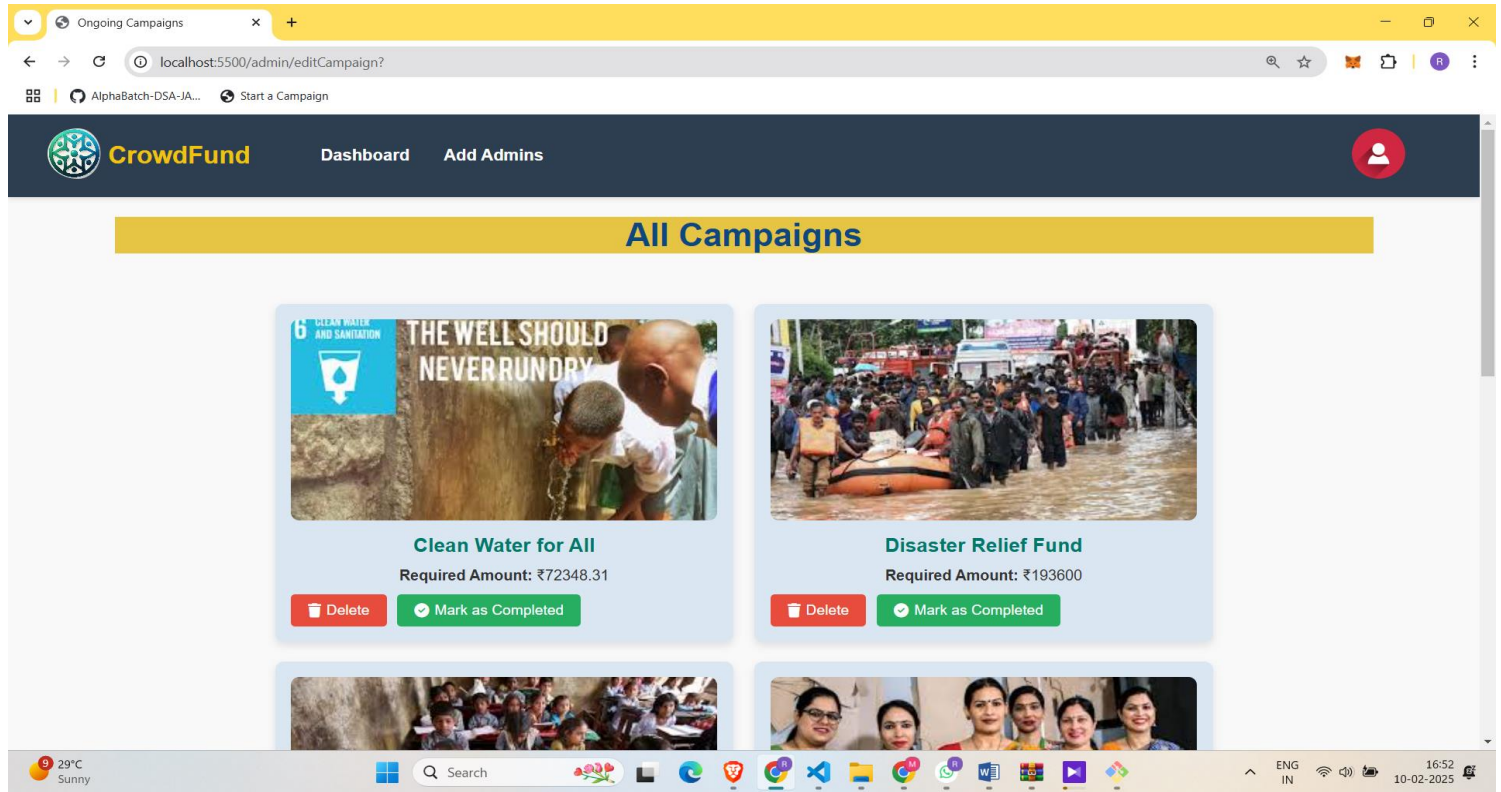
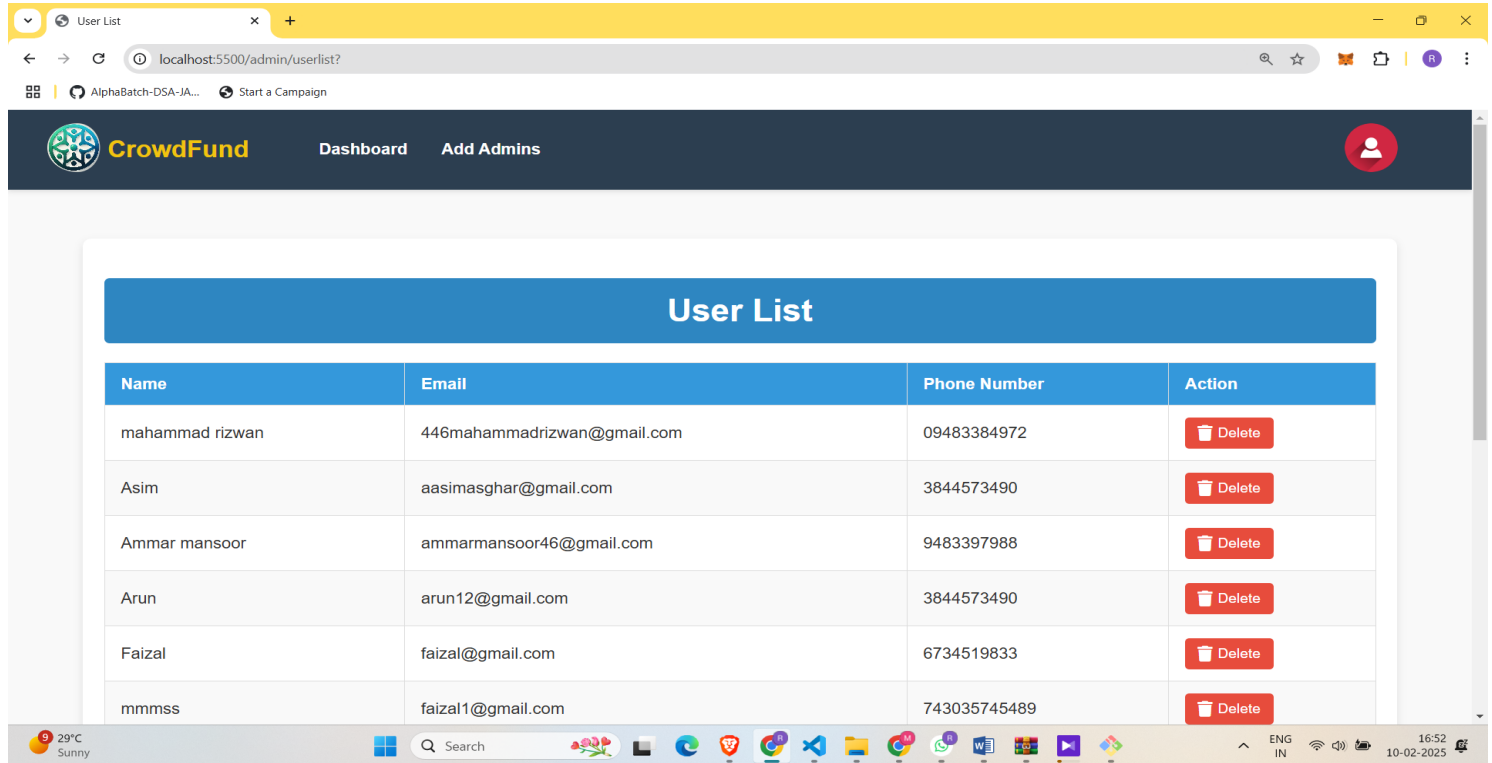
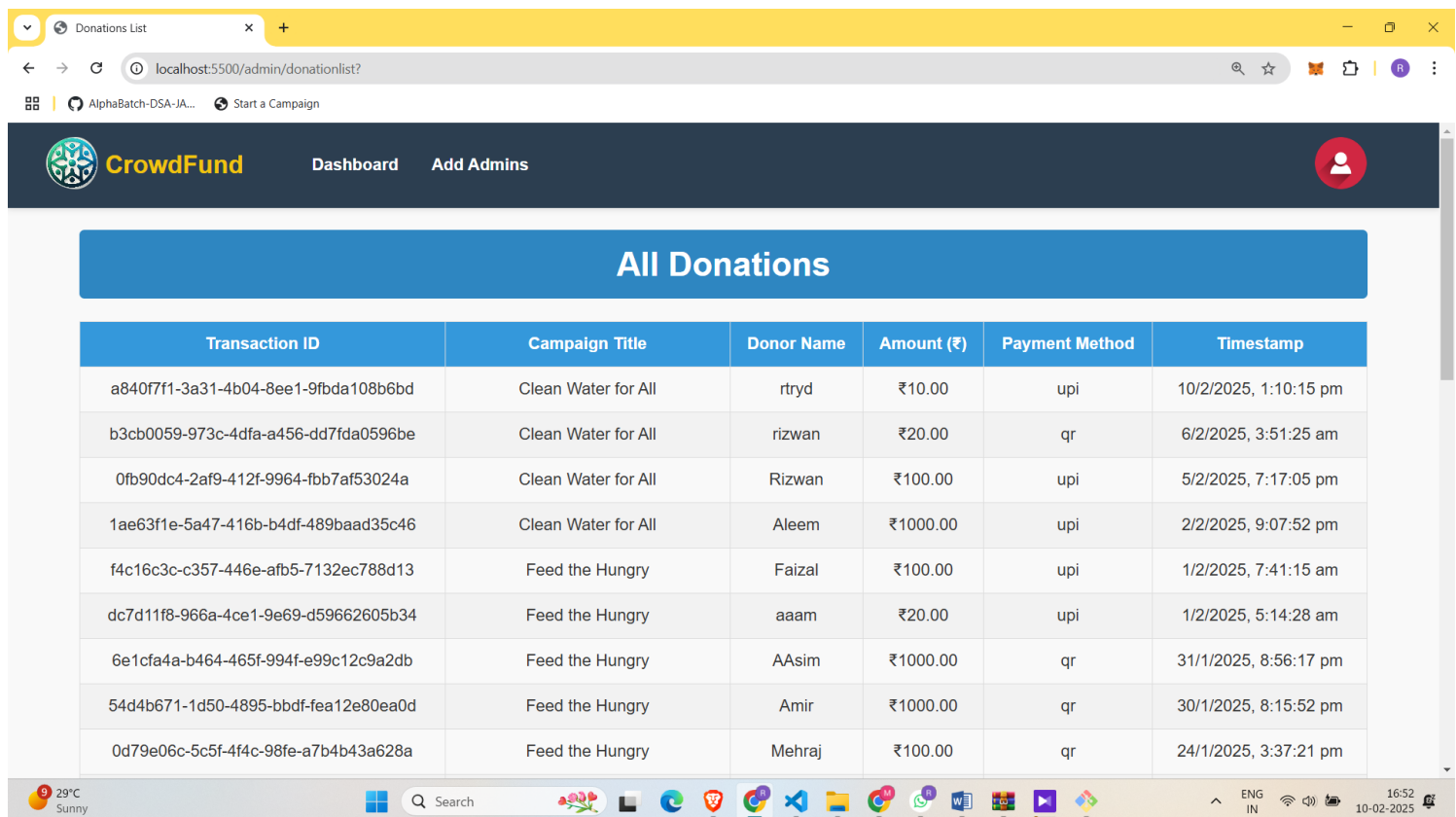


Fig A.12 Admin Ongoing Campaign Management Page Overview



User List

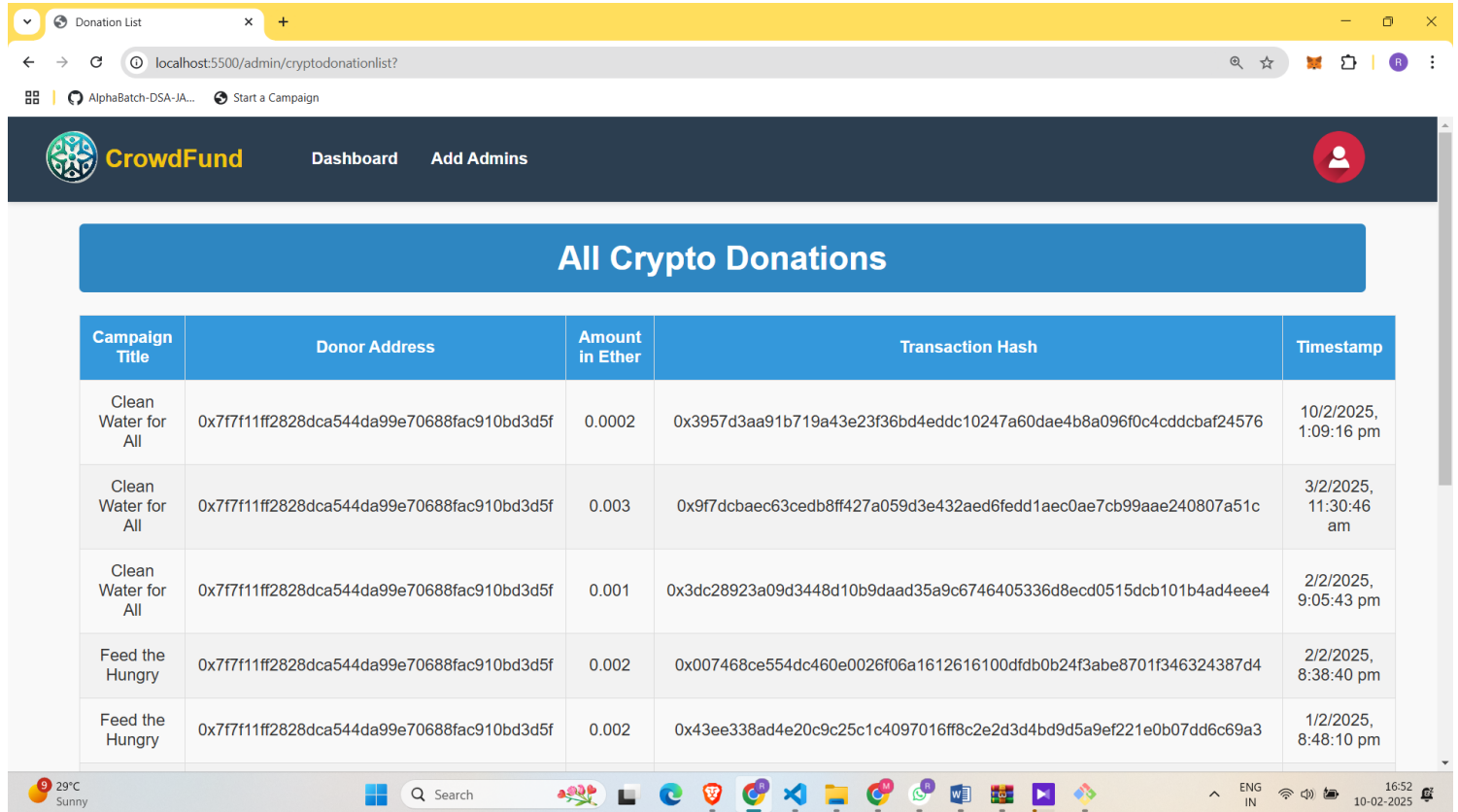
Name	Email	Phone Number	Action
mahammad rizwan	446mahammadrizwan@gmail.com	09483384972	Delete
Asim	aasimasghar@gmail.com	3844573490	Delete
Ammar mansoor	ammarmansoor46@gmail.com	9483397988	Delete
Arun	arun12@gmail.com	3844573490	Delete
Faizal	faizal@gmail.com	6734519833	Delete
mmmss	faizal1@gmail.com	743035745489	Delete

Fig A.13 Admin User Management

All Donations

Transaction ID	Campaign Title	Donor Name	Amount (₹)	Payment Method	Timestamp
a840f7f1-3a31-4b04-8ee1-9fbda108b6bd	Clean Water for All	rtryd	₹10.00	upi	10/2/2025, 1:10:15 pm
b3cb0059-973c-4dfa-a456-dd7fda0596be	Clean Water for All	rizwan	₹20.00	qr	6/2/2025, 3:51:25 am
0fb90dc4-2af9-412f-9964-fbb7af53024a	Clean Water for All	Rizwan	₹100.00	upi	5/2/2025, 7:17:05 pm
1ae63f1e-5a47-416b-b4df-489baad35c46	Clean Water for All	Aleem	₹1000.00	upi	2/2/2025, 9:07:52 pm
f4c16c3c-c357-446e-afb5-7132ec788d13	Feed the Hungry	Faizal	₹100.00	upi	1/2/2025, 7:41:15 am
dc7d11f8-966a-4ce1-9e69-d59662605b34	Feed the Hungry	aaam	₹20.00	upi	1/2/2025, 5:14:28 am
6e1cfa4a-b464-465f-994f-e99c12c9a2db	Feed the Hungry	AAsim	₹1000.00	qr	31/1/2025, 8:56:17 pm
54d4b671-1d50-4895-bbdf-fea12e80ea0d	Feed the Hungry	Amir	₹1000.00	qr	30/1/2025, 8:15:52 pm
0d79e06c-5c5f-4f4c-98fe-a7b4b43a628a	Feed the Hungry	Mehraj	₹100.00	qr	24/1/2025, 3:37:21 pm

Fig A.14 Admin Traditional Donations Management Page Overview



The screenshot displays the 'All Crypto Donations' management page. The page header includes the 'CrowdFund' logo and navigation links for 'Dashboard' and 'Add Admins'. A user profile icon is visible in the top right corner. The main content area features a table with the following data:

Campaign Title	Donor Address	Amount in Ether	Transaction Hash	Timestamp
Clean Water for All	0x7f7f11ff2828dca544da99e70688fac910bd3d5f	0.0002	0x3957d3aa91b719a43e23f36bd4eddc10247a60dae4b8a096f0c4cddcbaf24576	10/2/2025, 1:09:16 pm
Clean Water for All	0x7f7f11ff2828dca544da99e70688fac910bd3d5f	0.003	0x9f7dcbac63cedb8ff427a059d3e432aed6fedd1aec0ae7cb99aae240807a51c	3/2/2025, 11:30:46 am
Clean Water for All	0x7f7f11ff2828dca544da99e70688fac910bd3d5f	0.001	0x3dc28923a09d3448d10b9daad35a9c6746405336d8ecd0515dcb101b4ad4eee4	2/2/2025, 9:05:43 pm
Feed the Hungry	0x7f7f11ff2828dca544da99e70688fac910bd3d5f	0.002	0x007468ce554dc460e0026f06a1612616100dfdb0b24f3abe8701f346324387d4	2/2/2025, 8:38:40 pm
Feed the Hungry	0x7f7f11ff2828dca544da99e70688fac910bd3d5f	0.002	0x43ee338ad4e20c9c25c1c4097016ff8c2e2d3d4bd9d5a9ef221e0b07dd6c69a3	1/2/2025, 8:48:10 pm

The bottom of the screenshot shows a Windows taskbar with various application icons, a search bar, and system status indicators including temperature (29°C), weather (Sunny), and time (16:52, 10-02-2025).

Fig A.15 Admin Crypto Donations Management Page Overview

