Here is the detailed project plan for the blockchain-based donation system:

---

TABLE OF CONTENT

1.Introduction

1.1 Purpose

The primary goal of this project is to develop a secure, transparent, and decentralized donation platform utilizing blockchain technology. This system will enable organizations to create fundraising campaigns and allow users to contribute donations using cryptocurrencies, ensuring trust and efficiency in the donation process.

1.2 Project Scope

This project encompasses the design, development, and deployment of a web-based application that leverages blockchain for managing donations. Key features include user registration, campaign creation, donation processing, real-time tracking, and comprehensive reporting. The platform aims to provide a seamless experience for both organizations and donors, fostering greater engagement and trust.

---

2. Overall Description

2.1 Product Perspective

The donation system will function as a decentralized application (D-App) built on a blockchain network, such as Ethereum . By leveraging smart contracts, the platform ensures that transactions are secure, transparent, and immutable. This approach minimizes the need for intermediaries, reducing costs and potential points of failure, while enhancing trust among users.

2.2 Product Features

- Campaign Creation: User can initiate fundraising campaigns by specifying details such as the target amount, duration, and beneficiary information. Each campaign will have a unique identifier and a dedicated smart contract to manage donations.

- Donation Processing: Users can browse active campaigns and make donations using supported cryptocurrencies. The system will handle the conversion of donations into the organization's preferred currency, if necessary, and update the campaign's progress in real-time.

- User Registration and Authentication: Donors will need to register on the platform. Registration involves providing essential information and connecting a cryptocurrency wallet, such as MetaMask, to facilitate transactions.

- Real-Time Tracking and Reporting: The platform will offer dashboards for organizations to monitor campaign progress and for donors to track their contributions. Additionally, it will provide features for organizations to withdraw funds once the campaign concludes or the target is reached.

- Security Measures: To ensure the safety of funds and user data, the platform will implement robust security protocols, including data encryption, secure user authentication, and regular security audits.

2.3 User Classes and Characteristics

- Donors: Individuals or entities interested in contributing to campaigns. Donors can browse active campaigns, make donations, and view their donation history and receipts.

Both user groups will interact with the platform through a user-friendly interface designed to facilitate seamless navigation and transaction processing.

2.4 Operating Environment

The platform will be accessible via modern web browsers, including Chrome, Firefox, and Safari, ensuring cross-platform compatibility. The backend will be built using Node.js , with smart contracts deployed on a blockchain network like Ethereum. For data storage, MongoDB will be utilized to manage off-chain information, such as user profiles and campaign details.

2.5 Design and Implementation

The development process will follow best practices in software engineering, including modular design, code reusability, and thorough testing. Key components include:

- Frontend: Developed using Next.js and Tailwind Css provide a responsive and intuitive user interface.

- Backend: Implemented with Node.js and Remix Ide to handle API requests, user authentication, and interaction with the blockchain.

- Blockchain Interaction: Utilization of Web3.js or Ethers.js and Rainbow Kit for smooth interaction with Metamask wallet to facilitate communication between the frontend and the blockchain network, enabling functions like transaction processing and smart contract interactions.

- Smart Contracts: Written in Solidity (for Ethereum) or a similar language, these contracts will manage campaign creation, donation handling, and fund disbursement, ensuring transparency and security.

- Database: MongoDB will store off-chain data, including user information, campaign details, and transaction histories, providing efficient data retrieval and management.

2.6 Assumptions and Dependencies

- User Wallets: It is assumed that users possess a cryptocurrency wallet (e.g., MetaMask) with sufficient funds to make donations. The platform will provide guidance on setting up and using these wallets.

- Blockchain Network: The chosen blockchain network Ethereum is assumed to be operational and reliable. The platform's performance may depend on the network's stability and scalability.

- Regulatory Compliance: Smart contracts can automatically enforce compliance rules when certain conditions are met, which reduces the need for manual intervention.

---

3. System Features

3.1 Functional Requirements

- Campaign Management: Organizations can create, edit, and manage campaigns, specifying details like title, description, target amount, duration, and beneficiary information. Each campaign will have a unique identifier and a dedicated smart contract.

- Donation Handling: Users can select campaigns to support and make donations using cryptocurrencies. The system will process these donations, update the campaign's progress, and provide receipts to donors. Donations will be recorded on the blockchain, ensuring transparency.

- User Authentication: Secure registration and login processes for both organizations and donors, including features like password recovery and two-factor authentication to enhance security.

- Fund Disbursement: Upon reaching the fundraising goal or campaign conclusion, organizations can withdraw the collected funds. The platform will handle the disbursement process, ensuring that funds are transferred securely and in accordance with the campaign's terms.

- Reporting and Analytics: The platform will provide comprehensive reports on campaign performance, donor contributions, and transaction histories. These reports will be accessible to organizations and donors, facilitating transparency and informed decision-making.

---

4. External Interface Requirements

4.1 User Interfaces

- Home Page: A centralized hub displaying active campaigns, featured causes, and navigation links. The home page will provide an overview of the platform's mission and how users can get involved.

- Campaign Creation Page: A dedicated interface for organizations to input campaign details, set fundraising goals, and upload relevant media. This page will guide organizations through the process of launching a new campaign.

- Campaign Details Page: A detailed view of each campaign, showcasing information like the campaign's progress, number of donors, and updates from the organization. Donors can use this page to make informed decisions about their contributions.

-User Dashboard: Personalized dashboards for both organizations and donors, displaying relevant information such as active campaigns, donation history, and account settings. The dashboard will serve as a central point for users to manage their activities on the platform.

- Transaction History Page: A comprehensive list of all transactions, including donations, fund withdrawals, and other financial activities. This page will provide transparency and allow users to track their financial interactions with the platform.

4.2 Hardware Interfaces

The platform is designed to be hardware-agnostic, requiring only a device with internet access and a modern web browser. This includes desktops, laptops, tablets, and smartphones, ensuring accessibility for all users.

4.3 Software Interfaces

- Blockchain Network: The platform will interface with the chosen blockchain network (e.g., Ethereum) to deploy smart contracts, process transactions, and retrieve data

4.4 Communication Interfaces

The communication between the frontend, backend, and blockchain network will be facilitated using APIs and blockchain interaction libraries. Specifically:

- Web3.js or Ethers.js: These libraries will handle communication between the user’s browser and the blockchain. They will allow the application to send transactions, query the blockchain for data (e.g., donation history), and interact with smart contracts.

- HTTP/HTTPS: Standard web protocols (HTTP/HTTPS) will be used for secure data transfer between the frontend, backend, and any off-chain storage (e.g., user data in MongoDB/Firebase).

- WebSocket: Used for real-time updates (e.g., live campaign progress) by establishing a continuous connection between the server and client.

---

This section ensures the platform's communication protocols are clear and secure. Let me know if you need further adjustments!