

---

## Why This Project Is Cool for NGOs

This system is like turning your FundMe App into a high-tech, trustworthy fundraiser. Donors can:

- See exactly where their money goes (transparency).
- Trust it won't be stolen (security).
- Vote on how it's used (governance).
- Check cool charts to understand the impact (visualization).

For NGOs, it's a way to prove they're honest, attract more donors, and make managing money easier with automatic rules. For high school kids like you, it shows how tech like blockchain can solve real-world problems, like making sure donations help animals or people in need without any funny business!

---

Imagine you're running a FundMe App to raise money for a local animal shelter, but people want to know exactly where their money goes. Now, think of a super-smart, transparent, and secure system that tracks every penny and lets donors decide how it's spent. That's what this project, "Blockchain-Powered Transparency and Governance for Donated Funds in NGOs," is all about! It uses blockchain—a kind of digital ledger that's like a public notebook no one can erase—to make sure donations to non-profits (NGOs) are tracked, secure, and used properly. Let's break down the 15 steps in this project in a way that's easy to understand, like explaining how your FundMe App could use this tech.

---

## Step 1: Setting Up the Workspace (Environment Setup in Google Colab)

**What's Happening:** This is like setting up your kitchen before making FundMe App. You need the right tools, like a juicer and cups. In this project, we use Google Colab, a free online tool that's like a virtual computer for coding. We install special programs (libraries) to work with blockchain, write smart contracts (like digital rulebooks), and create cool graphs to show donation data.

**Why It Matters:** It's an easy, free way for anyone to start building this system without needing a fancy computer. These tools let us talk to the blockchain and show donors how their money is used, like a live scoreboard for your FundMe App donations.

**Example:** Imagine downloading apps to track sales and show customers a graph of how much you've raised. Here, we install:

- **Web3.py:** To talk to the blockchain.
  - **py-solc-x:** To write the rules for donations.
  - **Matplotlib & Plotly:** To draw charts showing who donated what.
-

## Step 2: Connecting to the Ethereum Blockchain

**What's Happening:** This is like connecting your FundMe App to a secure online bank. The Ethereum blockchain is a global, digital ledger where all transactions are stored forever. We use services like Infura or Alchemy (think of them as secure internet bridges) to connect our code to Ethereum without running a full blockchain computer ourselves.

**Why It Matters:** This makes sure every donation is recorded in a place no one can hack or change, so everyone trusts the system. It's like having a tamper-proof receipt book for every dollar donated to your stand.

**Example:** You set up a link to a bank account that everyone can see but no one can mess with. The code uses a special web address (like a URL) to connect to Ethereum's "Mainnet" (the real blockchain).

---

## Step 3: Creating Wallets for Donors and NGOs

**What's Happening:** Everyone needs a digital wallet, like a personal bank account, to send or receive money on the blockchain. We create wallets for donors (people giving money) and NGOs (the groups receiving it) using code. Each wallet has a unique address (like an email) and a secret key (like a password).

**Why It Matters:** These wallets let us track exactly who sent money and who received it, making everything super clear. It's like giving each customer and the animal shelter their own labeled money jar.

**Example:** Imagine giving every donor a special QR code to send money from, and the shelter gets its own code to collect it. The code creates these digital IDs automatically.

---

## Step 4: Writing the Smart Contract (NGOFundLedger)

**What's Happening:** A smart contract is like a robot accountant that follows rules written in code. We write one called "NGOFundLedger" in a language called Solidity. It has three main jobs:

- Accept donations from people.
- Let NGOs take out money (but only NGOs!).
- Keep a record of every donation.

**Why It Matters:** This contract runs on the blockchain, so it's automatic, can't be cheated, and everyone can see it. It's like a vending machine: you put in money, and it only gives out snacks to the right person.

**Example:** For your FundMe App stand, this is like a locked box that only opens for the shelter when they prove it's them, and it writes down every donation in a public notebook.

---

## Step 5: Compiling and Deploying the Smart Contract

**What's Happening:** After writing the smart contract, we turn it into a format the blockchain understands (like translating it into computer language) and “publish” it on Ethereum. Once it's live, it gets a unique address, like a website URL, that everyone can use to interact with it.

**Why It Matters:** This makes the contract a real, working part of the blockchain, ready to handle donations. It's like installing your locked donation box at the FundMe App and telling everyone where it is.

**Example:** You build the donation box, lock it, and place it in a public square where everyone can see it. The code compiles the contract and sends it to Ethereum.

---

## Step 6: Donor Sends a Donation

**What's Happening:** A donor sends money (in Ethereum's currency, ETH) to the smart contract. We create a transaction, sign it with the donor's secret key (like signing a check), and send it to the blockchain. The donation is now recorded forever.

**Why It Matters:** This makes donating secure and transparent. Anyone can check the blockchain to see the donation happened, like checking a public receipt.

**Example:** A customer scans your QR code to send \$5 to the shelter's donation box. The code builds a digital “check” and records it in the blockchain notebook.

---

## Step 7: NGO Releases Funds

**What's Happening:** When the NGO needs to use the donated money, they send a request to the smart contract. The contract checks if it's really the NGO (using their wallet) and releases the funds. Only the NGO can do this.

**Why It Matters:** This keeps the money safe from thieves or misuse. It's like ensuring only the shelter manager can unlock the donation box.

**Example:** The shelter uses their special key to open the box and take the money. The code ensures only their wallet can trigger this.

---

## Step 8: Verifying Transactions

**What's Happening:** Every blockchain transaction gets a unique ID (called a hash). We use code to check this ID and confirm the transaction worked (like checking if a bank transfer went through). This gives us details like who sent it and how much.

**Why It Matters:** Donors can see proof their money reached the NGO, building trust. It's like getting a text from the bank saying, “Your \$5 is with the shelter!”

**Example:** You scan a receipt's barcode to confirm the donation was recorded. The code pulls up the transaction details from Ethereum.

---

### Step 9: Checking Donation History

**What's Happening:** The smart contract has a feature to show the latest donation's details, like who donated and how much. We use code to ask the contract for this info, and it replies instantly.

**Why It Matters:** This lets anyone audit the donations, keeping NGOs honest. It's like having a public list of everyone who donated to your stand and how much they gave.

**Example:** You check the donation box's logbook to see the last person who gave \$5. The code queries the contract for this info.

---

### Step 10: Showing Donation Trends (Visualize Donation Timeline)

**What's Happening:** We use a library called Matplotlib to draw a line graph showing when donations came in and how much. It's like a chart tracking your FundMe App stand's sales over a week.

**Why It Matters:** This makes it easy for donors to see patterns, like if more people donate on weekends. It builds trust by showing activity clearly.

**Example:** You draw a graph showing you got \$10 on Monday, \$15 on Tuesday, etc. The code plots donation amounts over time.

---

### Step 11: Showing How Funds Are Split (Visualize NGO Fund Allocation)

**What's Happening:** We create a pie chart to show how donations are divided among different NGOs or causes (e.g., 50% to animal shelters, 30% to schools). This uses Matplotlib again.

**Why It Matters:** Donors love seeing where their money goes, like knowing their \$5 helped feed dogs or buy books. It makes the system transparent and exciting.

**Example:** Your pie chart shows 60% of donations went to dog food, 40% to cat toys. The code draws this chart based on fund data.

---

### Step 12: Comparing Donations vs. Spending (Compare Donations vs Releases)

**What's Happening:** We use Plotly to make a bar graph comparing how much money was donated vs. how much NGOs spent. This shows if NGOs are using funds quickly or holding onto them.

**Why It Matters:** It helps donors know if their money is being used effectively, like checking if the shelter bought dog food yet. It keeps NGOs accountable.

**Example:** Your graph shows you got \$100 but only spent \$80 so far. The code compares donation totals to withdrawal totals.

---

### Step 13: Letting Donors Vote (DAO Governance Simulation)

**What's Happening:** Donors get “voting tokens” based on how much they donate (e.g., \$10 = 10 tokens). They use these to vote on how funds are spent, like choosing between dog food or vet visits. This is called a DAO (Decentralized Autonomous Organization).

**Why It Matters:** It gives donors a say, making them feel involved. It's like letting customers vote on whether your stand's profits go to dogs or cats.

**Example:** A donor who gave \$20 gets 20 votes to decide if the shelter buys food or medicine. The code assigns tokens based on donations.

---

### Step 14: Checking System Health (Monitor Transaction Status)

**What's Happening:** We create a pie chart showing how many transactions worked vs. how many failed. This helps us spot problems in the system.

**Why It Matters:** It ensures the system is reliable, so donors don't worry about their money getting lost. It's like checking if your donation box's lock ever jams.

**Example:** Your chart shows 95% of donations went through, 5% failed due to errors. The code tracks transaction success rates.

---

### Step 15: Keeping Everything Secure (Security Summary)

**What's Happening:** The blockchain and smart contracts are super secure because:

- Once written, they can't be changed (immutable).
- Everything is public and logged forever.
- No single person controls it (decentralized).

**Why It Matters:** This protects donations from theft, fraud, or government interference, especially in places where trust is low. It's like a donation box that's unbreakable and always watched by everyone.

**Example:** Your FundMe App stand's box can't be stolen, and every dollar is tracked in a notebook anyone can read. The blockchain ensures this for NGOs.

---

For NGOs, it's a way to prove they're honest, attract more donors, and make managing money easier with automatic rules. For high school kids like you, it shows how tech like blockchain can solve real-world problems, like making sure donations help animals or people in need without any funny business!