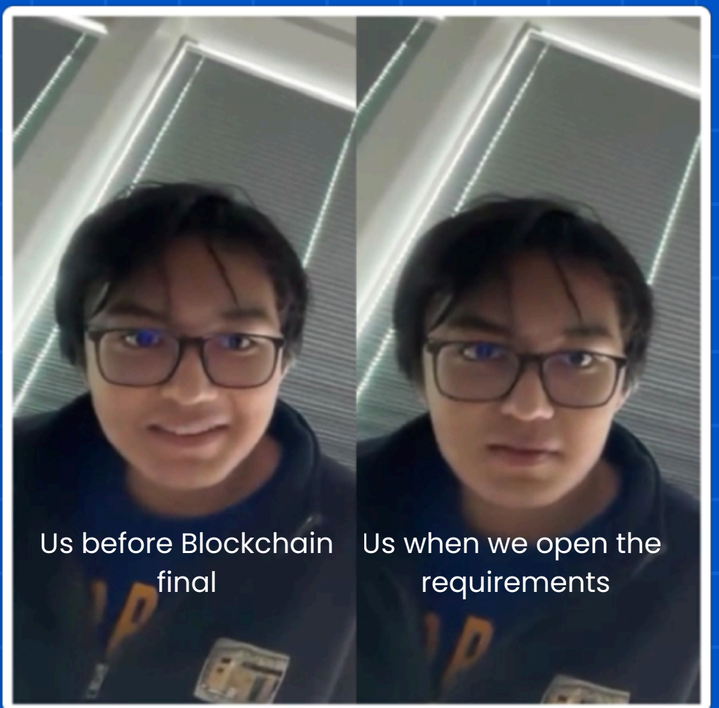
FINAL PROJECT

STUDENTS:
AIDA, TOMI, ARUKA



Purpose & Idea

Build a real decentralized crowdfunding application on an Ethereum test network.

Users create campaigns, contribute test ETH, and receive reward ERC-20 tokens automatically.

Architecture

- Frontend (HTML/JS): UI for creating campaigns and contributing
- MetaMask: wallet connection + transaction signing
- Smart Contracts (Solidity):
  Crowdfunding contract (campaigns + contributions)
  ERC-20 reward token (mint on contribution)
- Ethereum Testnet: blockchain network for deployment and testing
- Flow: User → Frontend → MetaMask → Smart Contract → Blockchain → UI updates

Smart Contract: Crowdfunding Logic

Main features:
Create campaign: title, goal, deadline
Contribute: send test ETH to active campaign
Track contributions per user (mapping)
Finalize after deadline:
If goal reached → funds go to campaign owner
Campaign status becomes finalized

Security checks (require):
cannot contribute after deadline
cannot finalize before deadline
cannot contribute to finalized campaign

ERC-20 Reward Token

Custom ERC-20 token:
minted automatically when user contributes
proportional to contribution amount
educational token (no real value)
Why token is included:
To demonstrate tokenization and ERC-20 standard usage.

MetaMask Integration

What we implemented:
Request account access: eth_requestAccounts
Display connected wallet address
Verify correct test network (Sepolia/Holesky/Local)
Execute transactions through MetaMask:
create campaign
contribute
finalize
Result: real on-chain transactions + user confirmation in MetaMask.

Frontend Features (UI)

User can:
connect wallet
create campaign
contribute test ETH
see transaction result (success/error)
view balances:
test ETH balance
reward token balance
view campaign data (goal, raised amount, deadline)

Deployment & How to Run

Deployed on: Ethereum Testnet (Sepolia/Holesky)

How to run locally:

1. git clone (repo)
2. open frontend (or run with Live Server)
3. connect MetaMask
4. switch to correct testnet
5. use faucet to get test ETH
6. test: create campaign → contribute → finalize

Demo Scenario

Demo steps:

- Connect MetaMask (address appears)
- Network check (testnet validated)
- Create a campaign (title/goal/deadline)
- Contribute test ETH (transaction confirmed)
- Show reward token minted (token balance increases)
- Finalize campaign after deadline

THANKS FOR YOUR ATTENTION