Student Startup Booster

FINAL EXAMINATION PROJECT

Olzhas Yelshibay

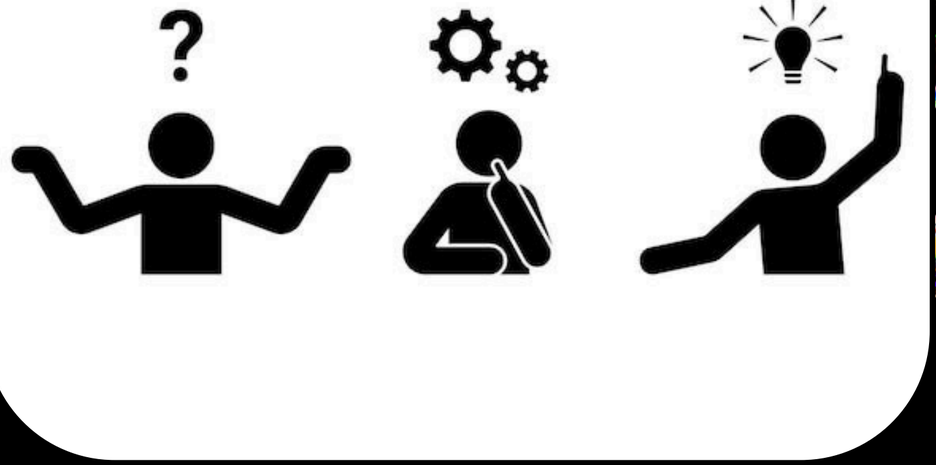
Dias Nygman

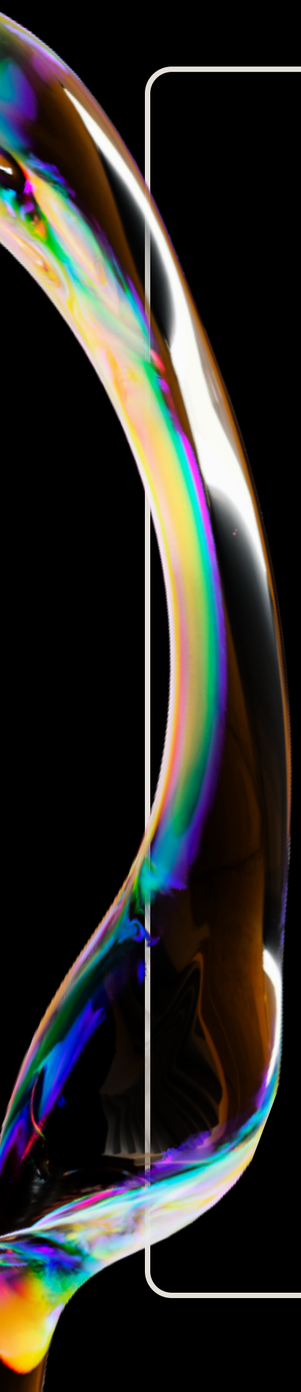
Alisher Sultanov

PROBLEM

THE MAIN PROBLEM WE FOCUS ON IS STUDENT FUNDRAISING.
MANY STUDENTS HAVE GOOD STARTUP IDEAS BUT FACE DIFFICULTIES WHEN TRYING TO COLLECT INITIAL FUNDING.

1. Student startups struggle to raise initial funding
2. Traditional platforms are centralized
3. Low transparency and trust
4. Intermediaries and extra fees

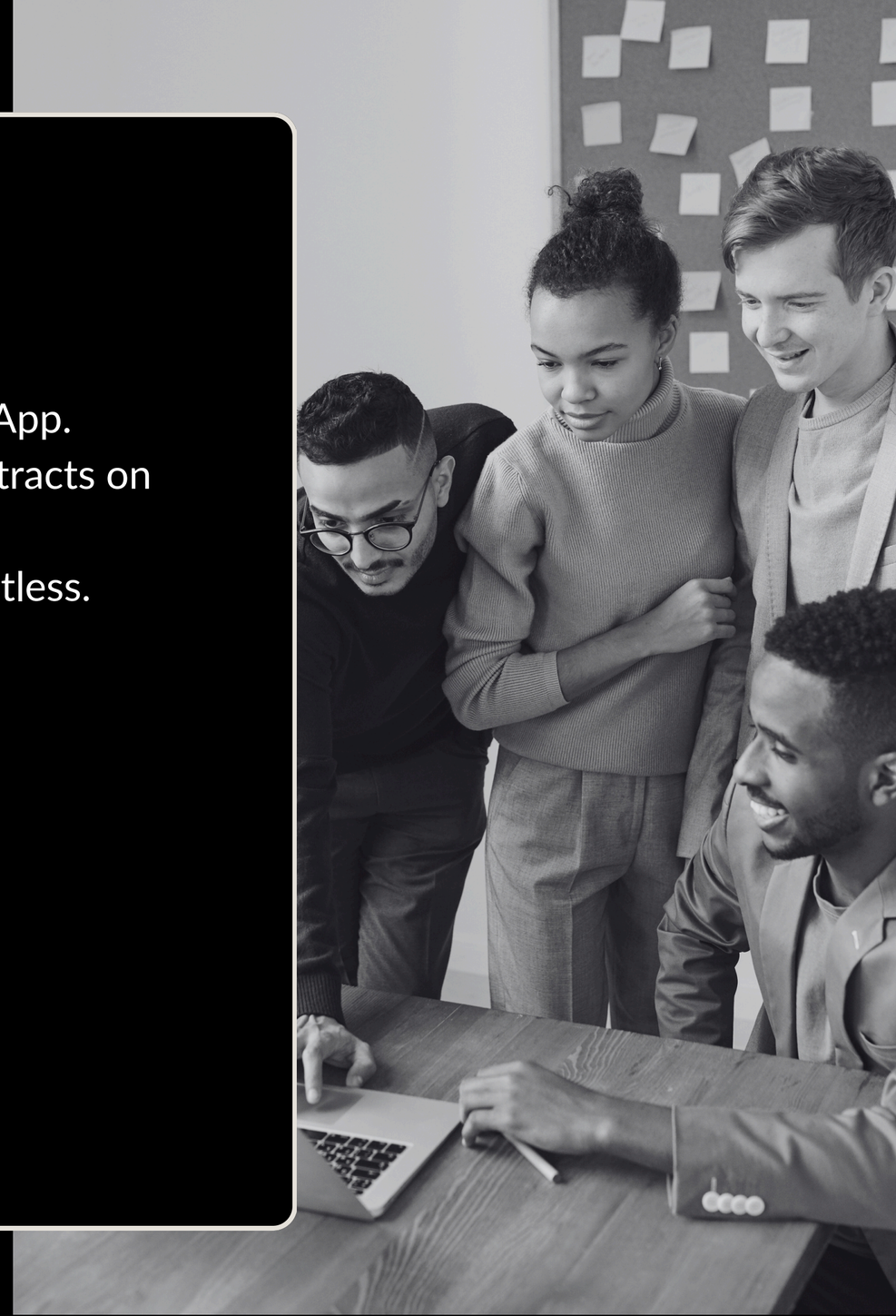


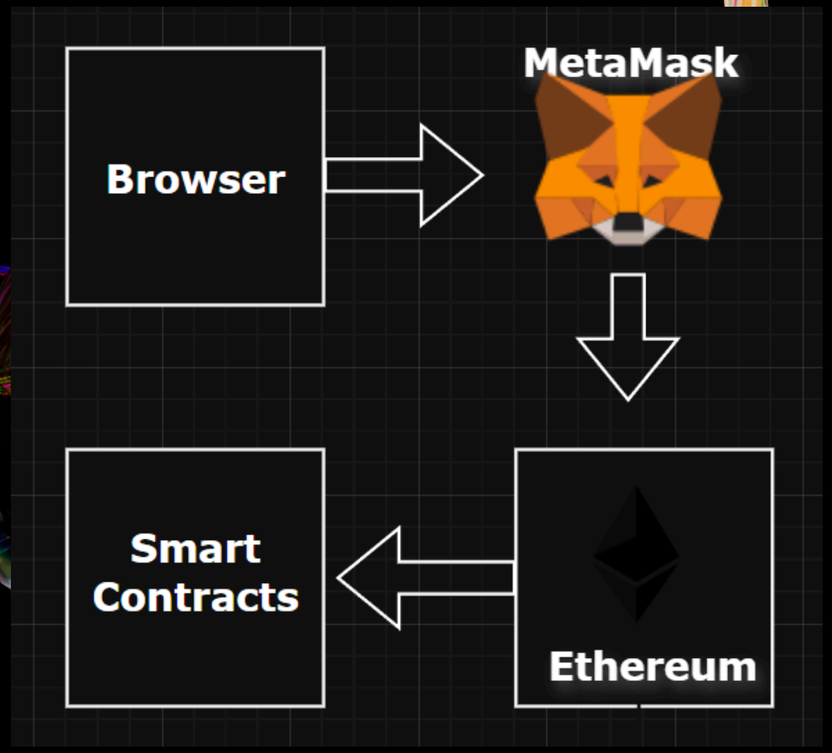


OUR IDEA

To solve this problem, we propose Student Startup Booster.
It is a decentralized crowdfunding platform implemented as a DApp.
Instead of a central authority, all rules are enforced by smart contracts on the blockchain.
This makes the fundraising process transparent, secure, and trustless.

- Built as a DApp on Ethereum
- No intermediaries
- All transactions are transparent and on-chain





- User opens the web application
- Connects MetaMask wallet
- Creates or supports a campaign
- Transaction is signed & sent to Ethereum
- Smart contracts execute the logic

HOW THE DAPP WORKS

The system works in the following way.
First, the user opens the web application and connects their MetaMask wallet. Then, the user can create a crowdfunding campaign or contribute test ETH to an existing one.
MetaMask signs the transaction and sends it to the Ethereum test network. Finally, the smart contracts execute the logic and store all data on-chain.

KEY FEATURES

Our DApp provides several core features.
Users can create crowdfunding campaigns by specifying a title, goal, and deadline.
Other users can contribute test ETH to active campaigns.
All contributions are stored on-chain and can be verified by anyone.
After contributing, users receive ERC-20 reward tokens as proof of participation.

- Create crowdfunding campaigns
- Contribute test ETH
- Track contributions on-chain
- Finalize campaigns after deadline
- Receive ERC-20 reward tokens (SUP)

SMART CONTRACTS

STARTUPBOOSTER.SOL

- Campaign creation
- Contributions & finalization
- Triggers rewards

SUPPORTTOKEN.SOL (ERC-20)

- Reward token (educational)
- Minted after contribution

The backend logic of the system is implemented using two smart contracts.
The first one is StartupBooster, which handles campaign creation, contributions, and finalization.
The second contract is SupportToken, which is an ERC-20 reward token.
This token has no real monetary value and is used only for educational purposes.
After a contribution, the StartupBooster contract mints reward tokens for the contributor.

TECHNOLOGY STACK



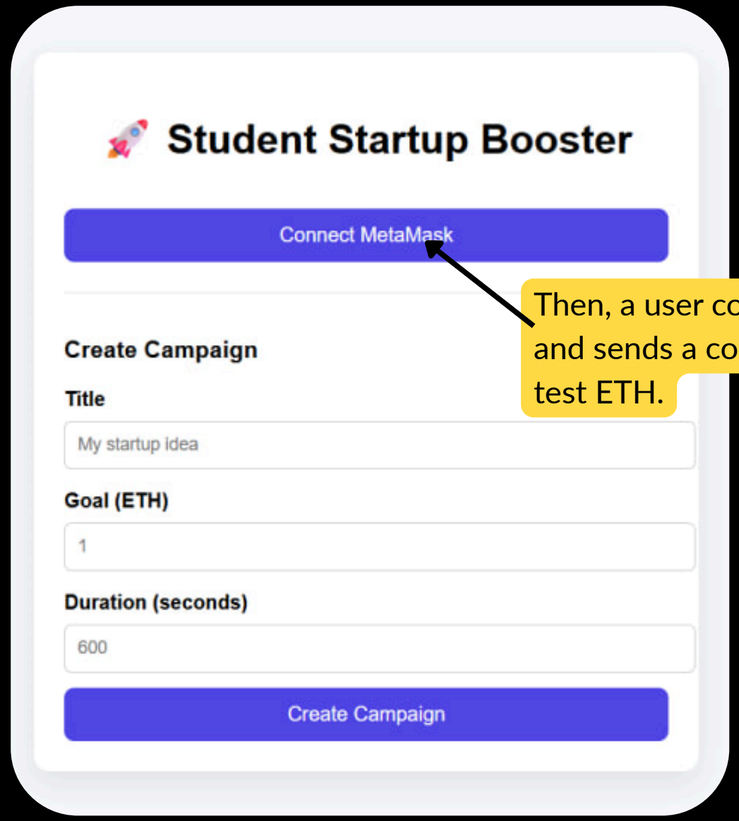

- Blockchain: Ethereum (Testnet)
- Smart Contracts: Solidity, OpenZeppelin
- Development: Hardhat
- Frontend: HTML, JavaScript, ethers.js
- Wallet: MetaMask
- Collaboration: GitHub

For smart contract development, we used Solidity and OpenZeppelin libraries for security.
Hardhat was used for compilation, deployment, and local testing.
The frontend is written in HTML and JavaScript and interacts with the blockchain using ethers.js.
MetaMask is used for wallet connection and transaction signing, and GitHub was used for team collaboration.

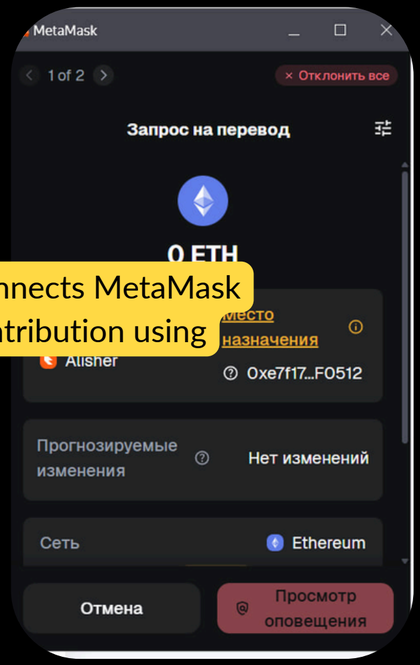
DEPLOYMENT & TESTING

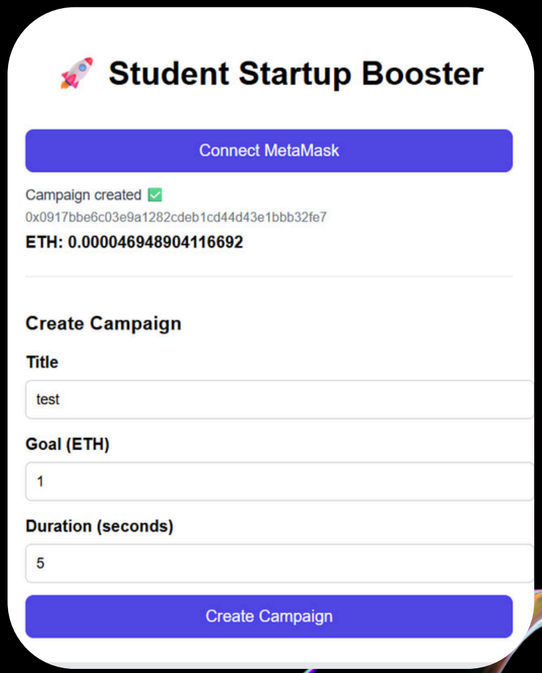
On this slide, we demonstrate how the application works in practice.

First, a campaign is created through the frontend interface.



Then, a user connects MetaMask and sends a contribution using test ETH.



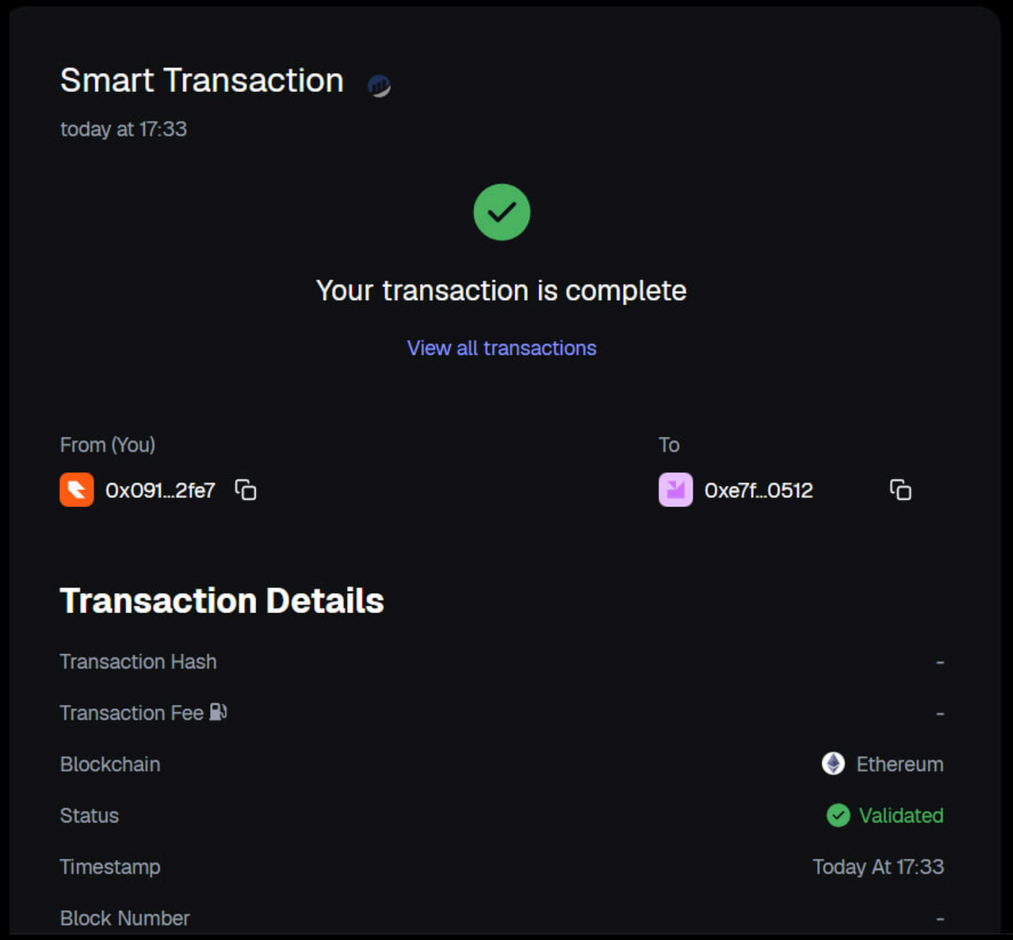


MetaMask requests confirmation, the transaction is signed, and after that it is permanently recorded on the blockchain.
This shows that the application is not just theoretical — it performs real blockchain transactions.

TRANSPARENCY AND TRUST

Transparency and On-Chain Verification

- All transactions are public
- No need to trust the platform
- Blockchain acts as a public ledger



TEAMWORK, RESULTS

Thank you for your time! Reach out to us for questions.

- Team-based development with GitHub
- Fully working decentralized application
- Clear learning outcomes

OLZHAS YELSHIBAY

Smart contracts, deployment

DIAS NYGMAN

ERC-20 token logic

ALISHER SULTANOV

Frontend, MetaMask integration, design

THANK YOU

for your time and attention