OrganMatch+ – Blockchain-Based Secure Organ Donation Platform

---

Team Members

Name	Course	Specialization	Semester	IAR No
Disha Girase	B.Tech CE	AI	5	14099
Meet Mochi	B.Tech CE	AI	5	14502

Guide / Mentor

- Dr.Maitri Patel

Abstract

OrganMatch+ is a blockchain-powered decentralized application (DApp) designed to ensure **transparency, security, and trust** in organ donation and transplantation.
The system leverages **smart contracts** to automate donor-recipient matching, prevent data tampering, and provide a **decentralized, tamper-proof ledger** for hospitals and authorities.

Keywords: Blockchain, Smart Contracts, Organ Donation, DApp, Security, Transparency

---

1. Introduction

Problem Statement:
Traditional organ donation systems face challenges like fraud, manual record errors, lack of transparency, and unauthorized access.

Purpose:
OrganMatch+ provides a secure, decentralized solution using

blockchain technology to manage donor-recipient records ethically and efficiently.

Scope:

- Hospitals and authorized administrators
- Donors and recipients
- Real-time tracking and verification of organ donation

---

2. Objectives

- Ensure authenticity and transparency of donor and recipient data
- Prevent data manipulation and unauthorized access
- Automate donor-recipient matching with smart contracts
- Provide a decentralized platform for hospitals and authorities
- Promote ethical organ donation practices

---

3. System Analysis

Existing System:

- Centralized manual record keeping
- Prone to data tampering and unauthorized access
- No automated matching system

Limitations:

- Slow and error-prone
- Lack of transparency

Proposed System:

- Blockchain-based decentralized ledger
- Automated donor-recipient matching using smart contracts
- Secure and transparent transaction logs

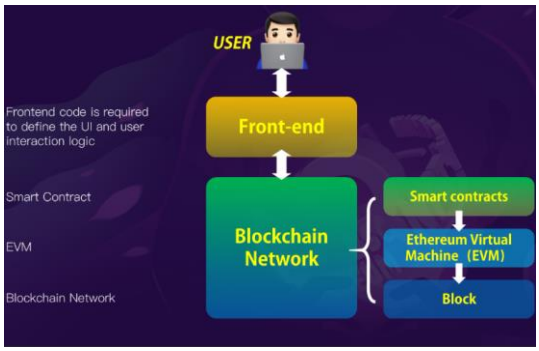
Advantages:

- Tamper-proof data storage
- Ethical and transparent organ allocation
- Easy verification by authorized hospitals

---

4. System Design

4.1 Architecture Diagram



4.2 Data Flow Diagram (DFD)

- Level 0: High-level process flow
- Level 1: Detailed flow of donor/recipient registration and matching

4.3 UML Diagrams

- **Use Case Diagram** – Shows donor, recipient, hospital, and admin interactions
- **Class Diagram** – Classes for smart contracts, users, and transactions
- **Sequence Diagram** – Flow of organ donation request, verification, and confirmation

---

5. Technology Used

Layer	Technology
Blockchain Framework	Ethereum / Hardhat

Smart Contract Language	Solidity
Frontend	React + Vite
Backend	Python Flask / Node.js
Storage	IPFS (for decentralized files)
Wallet Integration	MetaMask
Tools	Remix IDE, Hardhat, VS Code

---

6. Implementation

Smart Contract Overview:

- OrganDonation.sol – handles registration, verification, and matching
- Deployment via Hardhat scripts

Backend:

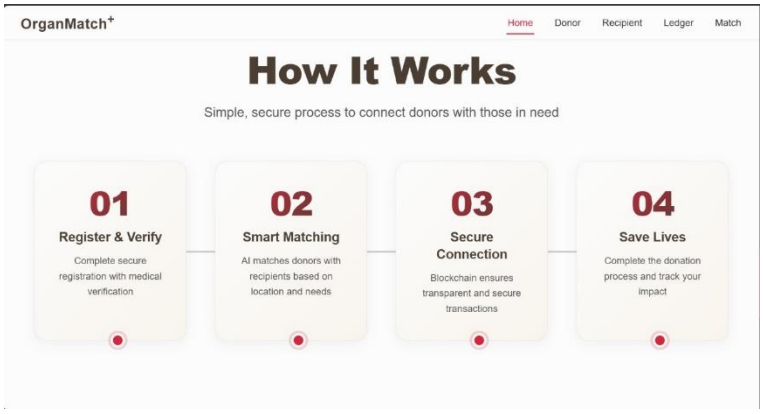
- Flask API connects frontend with blockchain

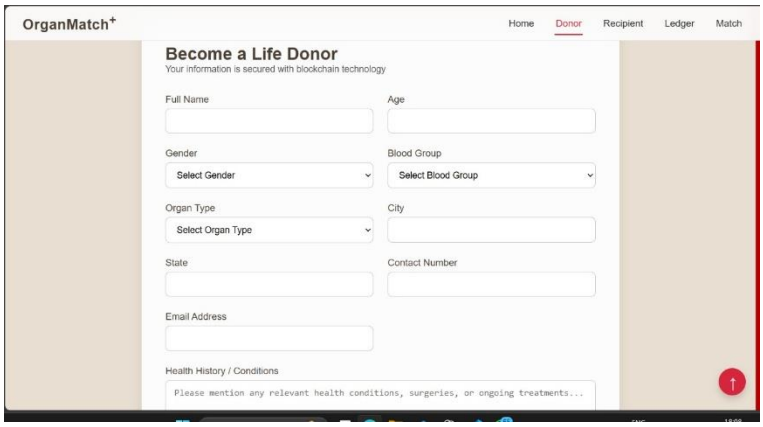
Frontend:

- React DApp allows donors and recipients to register and view matches

Screenshots:

- Donor Registration Form
- Recipient Approval Screen
- Transaction Confirmation on Blockchain

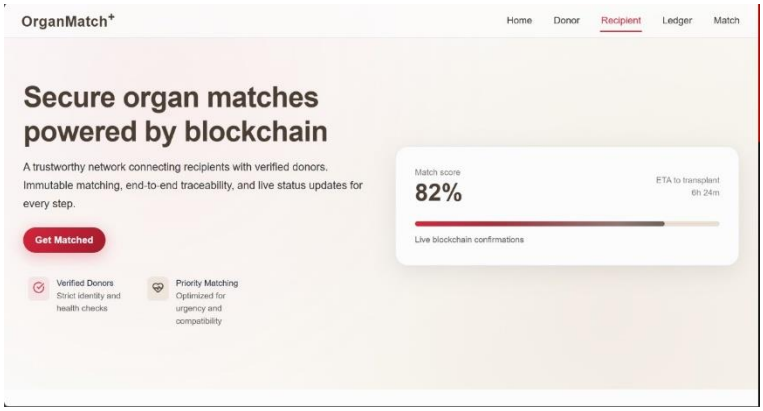


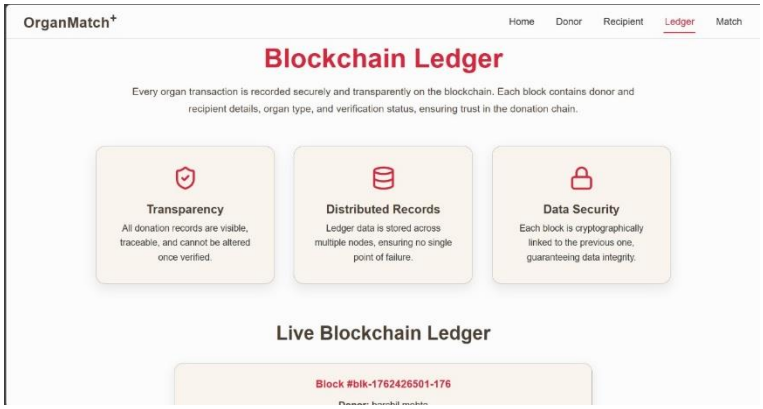


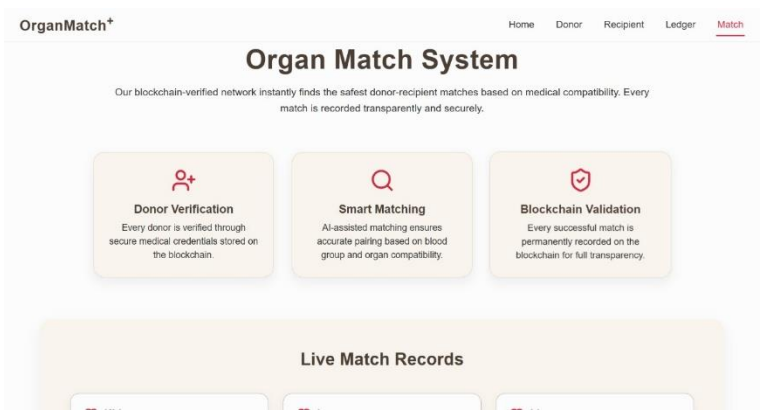
---

7. Testing & Results

- Unit tests executed using Hardhat
- Test cases include registration, verification, and successful matching
- Blockchain logs captured for each transaction
- Screenshots of test execution







---

8. Conclusion & Future Scope

Conclusion:
OrganMatch+ successfully provides a **secure, transparent, and decentralized organ donation system**. It minimizes fraud, automates matching, and ensures ethical management of donor-recipient data.

Future Scope:

- Integration with government health databases
- AI-based donor-recipient matching
- Multi-hospital collaboration
- Storage of medical files using IPFS

---

9. References / Bibliography

1. Ethereum.org – Ethereum Documentation
2. Solidity Docs – Solidity Language Reference

3. Hardhat.org – Ethereum Development Environment
4. Flask Documentation – Python Web Framework
5. ReactJS Docs – Frontend Library

---

10. Acknowledgments

- Special thanks to our project guide for guidance and support
- References: Hardhat, Remix IDE, ReactJS, MetaMask
