8/10/2025

BLOCKCHAIN BASED BLUE CARBON REGISTRY AND MRV SYSTEM

PRESENTATION BY:

SAUMY BHARGAVA 2024UCA1877

STUTI JAIN 2024UCA1818

PARINIT SINGH 2024UCA1844

INTRODUCTION

What is Blue Carbon Registry?

- **Blue Carbon: Carbon captured and stored by coastal and marine ecosystems (mangroves, tidal marshes, seagrass).**
- **The Registry: A digital platform to track blue carbon restoration projects, verify carbon sequestration, and manage the issuance, sale, and retirement of carbon credits.**

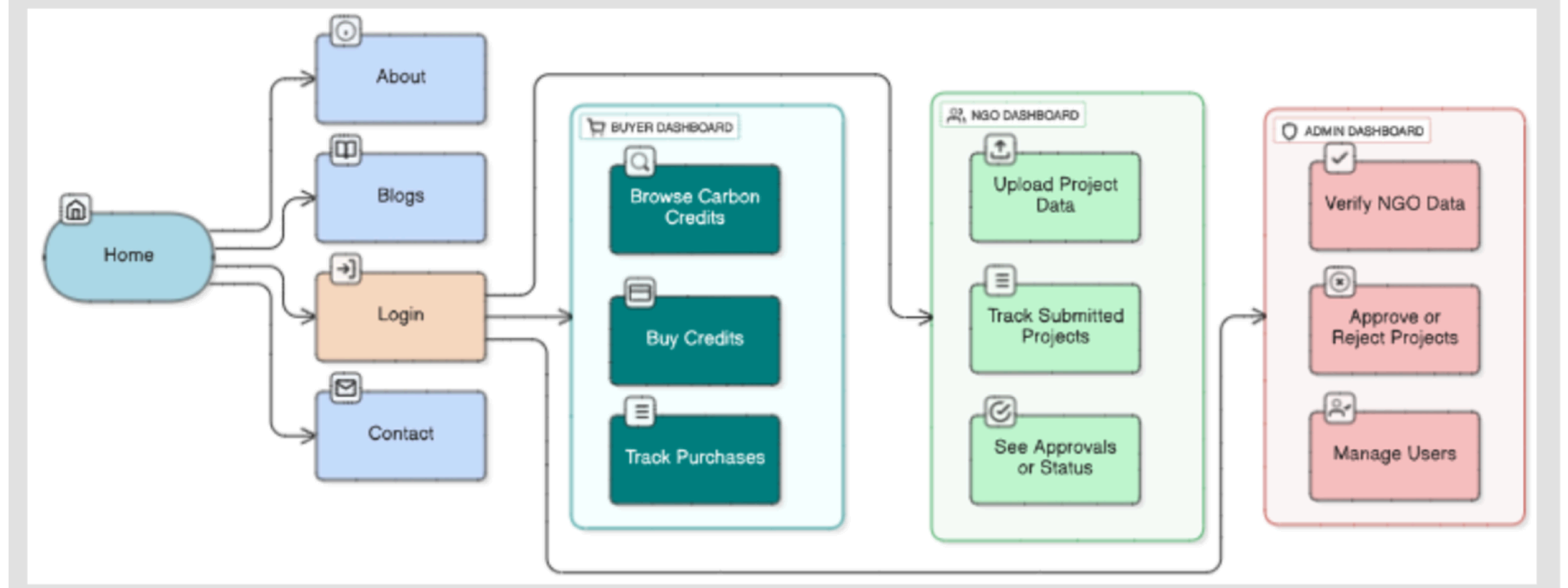
Importance/Relevance

A robust database is the backbone of this system, needed to manage complex data for projects, users, and transactions securely and efficiently

OBJECTIVES/PROBLEM STATEMENTS

- Establish Data Trust (Immutability): Implement a tamper-proof, auditable record for every carbon credit using a secure ledger.
- Ensure Data Accuracy (MRV): Design a decentralized Measurement, Reporting, and Verification (MRV) system to validate project data (e.g., location, restoration metrics).
- Optimize Data Flow (Efficiency): Reduce the credit approval timeline from years to months through automated verification triggers and system workflows.
- Promote Equity: Eliminate data intermediaries to ensure direct, traceable revenue flow to local communities/NGOs

METHODOLOGY/APPROACH



Frontend
React (Vite), TypeScript/JavaScript, HTML, CSS, styled with Tailwind CSS.

Backend
Node.js with Express.js (REST API), PostgreSQL (db).

Blockchain
Ethereum (Solidity smart contracts) with Hardhat for dev/testing.

Tools and DevOps
Git/GitHub , Thunder Client (API testing), pgAdmin & SQLTools (db management).

TOOLS AND TECHNOLOGIES USED

Relational Database (Centralized): PostgreSQL
Purpose: Stores dynamic data, user roles, project application metadata, dashboard metrics, and unverified data (pre-approval).

Distributed Ledger (Immutable Registry): Ethereum (Solidity Smart Contracts)
Purpose: Acts as the final, immutable registry for carbon credits, ensuring credit ownership and retirement status cannot be altered.

Backend/API: Node.js with Express.js (REST API)
Function: Manages the secure transfer of data between the UI, PostgreSQL, and the Ethereum Blockchain.

MAIN CONTENT/FINDINGS

Data Integrity and Immutability

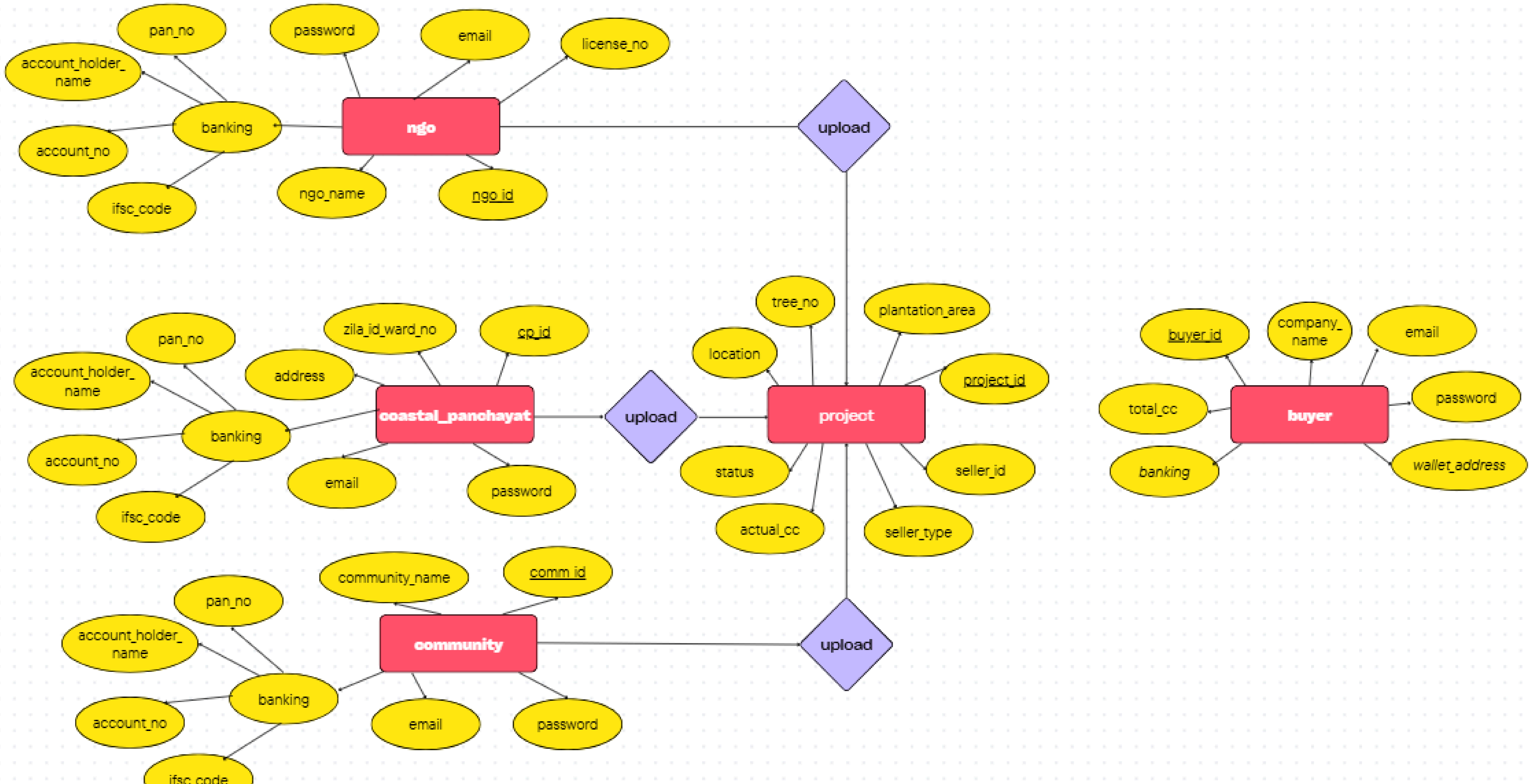
- Database Division: We segment data storage based on function:
    - PostgreSQL: Handles CRUD operations (Create, Read, Update, Delete) for non-critical, frequently changing data (e.g., draft project forms, user profiles).
    - Ethereum Blockchain: Handles Append-Only operations for critical data. Once a credit is issued or retired, that state is permanently recorded.
- Preventing Double-Counting:
    - The Smart Contract acts as the ledger logic. Each issued carbon credit is a unique, non-fungible token (NFT or similar digital asset).
    - When a buyer Retires a credit, the Smart Contract immediately updates its status to "Retired" on the Blockchain, making it technically impossible to sell or use again.

MAIN CONTENT/FINDINGS

Decentralized MRV (Measurement, Reporting, Verification)

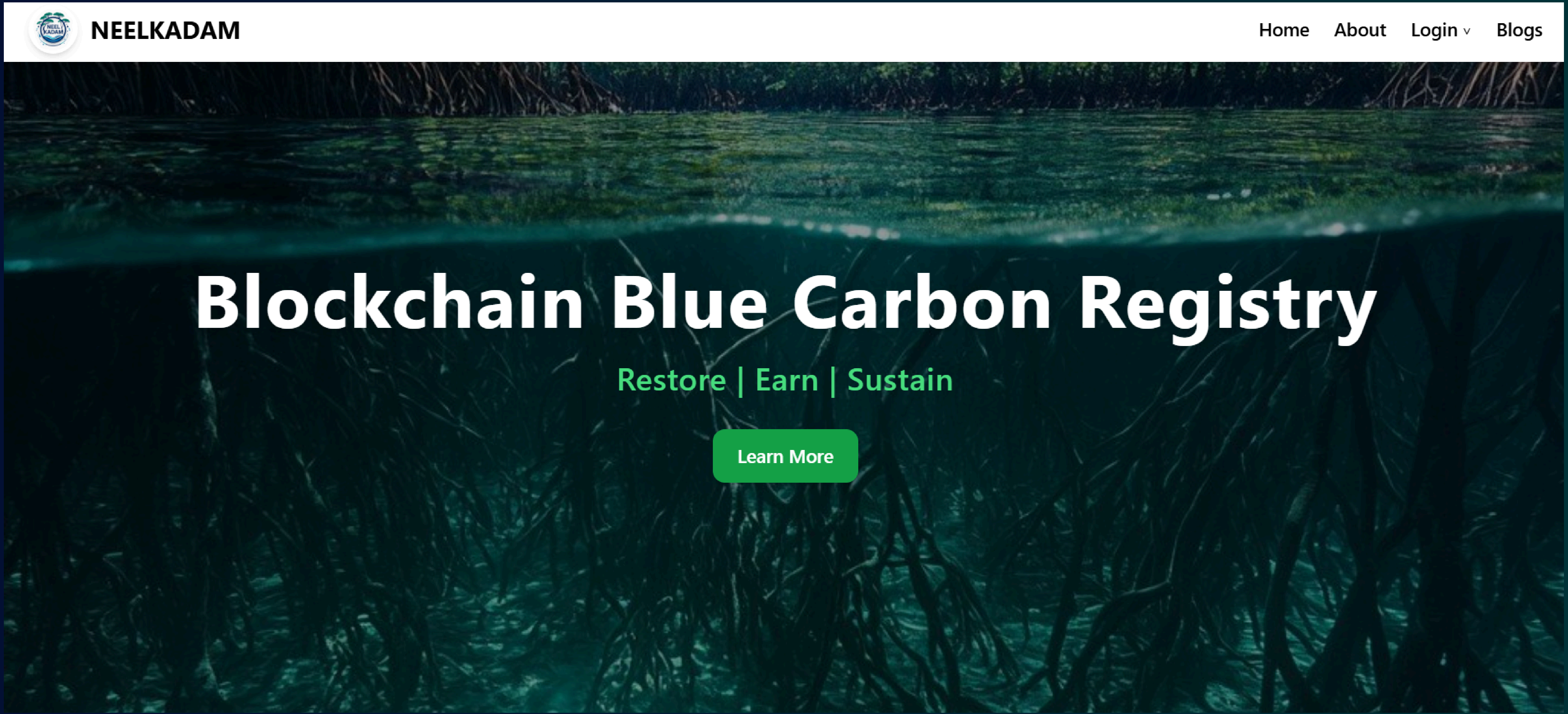
- Multi-Layer Verification Strategy: Data accuracy is maintained through redundant checks:
    a. Digital Proof: Geo-tagged photos and location data (Google Maps integration).
    b. External Validation: Government Verifiers (NCCR/MOES) manually review project data.
    c. Audit Trail: Every verification step is timestamped and logged, providing a transparent audit trail.

ER DIAGRAM



pan_no
password
email
license_no
account_holder_name
banking
ngo
upload
account_no
ifsc_code
ngo_name
ngo_id
tree_no
plantation_area
zila_id_ward_no
cp_id
pan_no
location
buyer_id
company_name
email
address
project_id
account_holder_name
total_cc
password
coastal_panchayat
upload
project
buyer
banking
account_no
status
seller_id
banking
wallet_address
email
password
ifsc_code
actual_cc
seller_type
community_name
comm_id
pan_no
account_holder_name
community
upload
banking
account_no
email
password
ifsc_code

APPLICATION/ STIMULATION



WEBSITE LINK: https://neel-kadam.vercel.app/
VIDEO EXPLANATION: https://youtu.be/kpIlNnXLj7k?si=nMmF97QFlrN43dkD

ANALYSIS/DISCUSSION

Advantages of Our DBMS Design

- High Trust & Auditability: The Blockchain component guarantees that the most critical data (credit status) is permanently immutable and publicly auditable.
- Operational Efficiency: The PostgreSQL and API layers allow for fast, scalable data interaction and dashboard reporting, which is impossible with a pure blockchain system.
- Scalability: The architecture is scalable, allowing easy expansion of PostgreSQL tables to track new ecological parameters without altering the core blockchain registry logic.

Key Disadvantages / Challenges

- Initial Data Uncertainty: The system verifies the authenticity of the uploaded proof (e.g., geo-tagged photos), but cannot guarantee the original accuracy of the raw field data (e.g., carbon stock measurements).
- Increased System Complexity: Managing a hybrid architecture (PostgreSQL and Blockchain) requires specialized expertise and complex synchronization logic for data consistency.
- Cost and Latency: Blockchain transactions require gas fees, and syncing data between the relational DB and the distributed ledger can introduce latency and variable costs.
- Data Error Identification: Difficulty in identifying and correcting fundamental errors in the initial field data uploaded by the NGO before it reaches the immutable ledger.

CONCLUSION

Key Takeaways

1.Hybrid DBMS Success: Successfully integrated PostgreSQL (Scalability) with Blockchain (Trust), establishing a robust, two-tiered data architecture.

2.Data Immutability Achieved: Ensured data integrity for all critical records, definitively solving the market problem of credit double-counting.

3.Streamlined Data Lifecycle: Reduced the entire project data pipeline, from NGO upload to credit retirement, into an efficient, automated workflow.

4.Enhanced Auditability: Created a public, transparent audit trail for all credit transactions, building high trust for corporate buyers.

5.Economic Empowerment: Successfully designed a system that directs verifiable revenue and financial data straight to local communities, bypassing intermediaries.

FUTURE SCOPES/RECOMMENDATIONS

- Mandatory Geo-Verification (Google Maps APIs): Integrate Google Maps APIs for mandatory geo-tagging of all project data. This ensures the location data stored in PostgreSQL is spatially indexed and cross-validated against satellite imagery, enhancing trust in MRV data submissions.

- AI-Driven Data Integrity and Estimation: Implement AI/Machine Learning models to analyze data patterns in PostgreSQL for fraud detection (flagging anomalies in reported growth/location data) and provide an automated, data-driven credit estimation forecast.

- International Interoperability and Scaling: Develop standardized Smart Contract interfaces that adhere to global registry formats (ee.g., Verra, Gold Standard). This allows verified credit data to be instantly recognized and traded in international carbon markets to scale the project beyond India.