

CarbonPay

Public Report I Module 1

Project Overview

CarbonPay is a decentralized platform built on the Solana blockchain to enable the tokenization, traceability, and retirement of certified carbon credits. It addresses persistent inefficiencies in the carbon market—such as double counting, lack of auditability, and low liquidity—by integrating smart contracts, decentralized storage, and regulatory-aligned processes.

The platform supports project developers, corporate offsetters, ESG professionals, and auditors through a unified interface. It facilitates project browsing, on-chain purchases, credit retirement, and compliance reporting, all while maintaining full data integrity via IPFS and Solana.

Module 1 Scope and Achievements

Module 1 focused on defining and delivering CarbonPay's technical foundations over a ten-week development cycle. Key achievements include:

- Initial deployment of a WebApp (Next.js + wallet integration) for credit visualization and onboarding
- Smart contract logic for credit minting and retirement implemented using Anchor on Solana Devnet
- Admin interface for project issuers with IPFS file storage
- Backend developed in Rust, integrated with PostgreSQL and IPFS
- Modular architecture designed for future integration with credit registries (e.g., Verra, Gold Standard)
- Visual prototypes tested with users and iterated for accessibility and clarity

Technical Architecture and Testing

CarbonPay's architecture is modular, layered, and aligned with ISO/IEC 25010 quality standards. It comprises five primary domains: frontend, backend, blockchain, data layer, and external services. The use of IPFS ensures decentralized document retention, while Solana provides low-cost, high-throughput blockchain operations.

Testing strategy includes:

- Unit testing (Vitest, Anchor) for both frontend components and smart contracts;
- Integration testing using Cypress and backend Rust modules;
- Static analysis of contracts to ensure security and correctness;
- Performance benchmarks targeting $\leq 1200\text{ms}$ response time and blockchain finality under 3 seconds;
- Test coverage $\geq 80\%$, with 99.5% uptime target and disaster recovery within 10 minutes.