BLOCKCHAIN PROJECT

CampusCoins - Campus Token Rewards System

Synopsis

A Blockchain-based token reward system that allows students to earn and redeem tokens using the barcode on their college ID cards. The system will promote and enhance student engagement in extracurricular and academic activities and provide incentives by allowing students to redeem tokens for discounts and offers at the campus canteen, bookstore, and other facilities.

Example Use Cases:

- A student scores above 35 in ISA's → earns **100 tokens**.
- A student attends a hackathon → earns **50 tokens**.
- A student attends a guest lecture → earns **20 tokens**.
- A student donates books to library → earns **25 tokens.**
- A student participates in a campus cleanup drive → earns **30 tokens**.
- A student volunteers for event management → earns **40 tokens**.

Project Features & Workflow

1) Token Generation and Storage

- Tokens are generated and stored on a **blockchain (like Polygon)** for transparency and security.

2) Web Interface

- Event organizers scan the **barcode on student ID cards** to:
  - Credit tokens for participation.
  - Allow students to check their token balance.
  - Redeem tokens at the canteen or bookstore.
  - Deduct tokens from the student's balance after redemption.

3) Token Transactions via Smart Contracts

- Token credits and debits are processed through **Solidity smart contracts** on the blockchain.

4) Bookstore/Canteen Side

- Canteen staff scan the barcode, verify the token balance, and process the redemption.

Slashing Mechanism (Penalty System)

Just as students **earn tokens** for positive actions, they can also **lose tokens** for rule violations. This ensures compliance with campus regulations and maintains a balanced token economy.

Penalty Triggers (Reasons for Token Deduction)

- A student not wearing an ID card on campus → loses **5 tokens**
- A student skipping mandatory events like guest lectures → loses **10 tokens**
- A student cheating in academic tests or plagiarism → loses **20 tokens**
- A student late return of library books → loses **5 tokens**
- A student damages college property or vandalism → loses **15 tokens**

How Slashing Works in Different Situations

1. **Automated Penalties via QR Scan:**
   - Security staff scans the **student's ID barcode** at entry points.
   - If the student is **not wearing an ID**, their **wallet is flagged**, and tokens are deducted.
   - Violation is **recorded on the blockchain**.
2. **Event-Based Penalties:**
   - If a student **skips a mandatory event**, tokens are **automatically deducted** from their balance.
   - The **event attendance list** (QR-based or manual entry) is checked before processing penalties.
3. **Manual Penalties by Admin:**
   - Faculty/Admin can **manually deduct tokens** for misconduct.
   - Admin enters **student ID + violation type + penalty amount** in the dashboard.

UI/UX Enhancements for Penalties

- **Penalty History Page:** Students can view all deductions & reasons.
- **Real-Time Notifications:** ⚠️ `10 tokens deducted for not wearing an ID card.`
- **Appeal Mechanism:** Allow students to dispute penalties with the admin.

Earning Back Slashed Tokens

To counterbalance penalties, students can **earn back lost tokens** by:

- Attending extra events (**bonus tokens**).
- Volunteering for campus services (**library work, event management**).

Functional Requirements

1) Student Side: Students should be able to:
✓ Scan barcode to earn tokens.
✓ View token balance in real-time.
✓ Redeem tokens at the canteen/bookstore.
✓ Receive notifications on successful transactions and penalties.

2) Event Organizer Side: Event organizers should be able to:
✓ Set token rewards for specific events.
✓ Generate and manage event IDs.

3) Canteen/Bookstore Side: Canteen staff should be able to:
✓ Scan barcodes to redeem tokens.
✓ View the token redemption status.
✓ Approve or reject token-based transactions.

4) Admin Side: Admin should be able to:
✓ Manage token supply.
✓ View and approve events.
✓ Monitor transaction history.
✓ Slash tokens for rule violations.

System Requirements

1) FrontEnd Requirements:

- **Framework:** React.js

2) Backend Requirements:

- **Framework:** Node.js + Express.js
- **Database:** MongoDB (for user info, event details; blockchain stores transactions)
- **Blockchain Interaction:** Web3.js

3) Blockchain Requirements:

- **Blockchain:** Polygon Testnet
- **Smart Contract Language:** Solidity
- **Wallet Integration:** MetaMask