Product Documentation

CodexAI – A Blockchain-based Competitive Coding Platform

1. Introduction

1.1 Purpose

The purpose of this document is to describe the architecture, components, and behavior of *CodexAI*, a decentralized platform for competitive coding that integrates blockchain and AI. This platform aims to incentivize programming through tokenized rewards while ensuring fairness and decentralization.

1.2 Scope

CodexAI merges competitive programming with blockchain technology, allowing users to create, solve, and stake tokens on programming problems. The platform leverages AI for validation, uses smart contracts on the Stellar blockchain, and ensures secure, trustless interactions between users and the platform.

1.3 Acronyms and Abbreviations

- **CDX** – CodexAI Token
- **DSA** – Data Structures and Algorithms
- **AI** – Artificial Intelligence
- **UX** – User Experience
- **dApp** – Decentralized Application
- **SDK** – Software Development Kit

2. Problem Statement

Users can showcase their credibility by solving programming challenges, but existing platforms do not incentivize this with real rewards. CodexAI solves this by enabling **token staking and reward-based contests**, creating a transparent and meritocratic environment for competitive programmers.

3. Technology Stack

3.1 Frameworks

- **Next.js** – Frontend development
- **Stellar CLI** – Blockchain interaction
- **Soroban SDK** – Smart contract development on Stellar

3.2 Languages

- **TypeScript** – Frontend and server logic
- **Rust** – Smart contract development

3.3 Deployment and Infrastructure

- **Frontend** deployed on **Vercel**
- **AI Agent** hosted on **Render**
- **Smart Contracts** deployed on the **Stellar blockchain**

4. System Features

4.1 Contest Participation and Staking

- Users stake CDX tokens to enter contests.
- Each participant's stake is pooled.
- AI Agent scores the solutions and ranks participants.
- Top 5 ranked participants and problem creators receive a portion of the reward pool.

4.2 Problem Submission

- Users can submit custom problems.
- Validator AI Agent checks constraints, test coverage, and logic.
- Questions are auto-tagged and difficulty-labeled.

4.3 Validator AI Agent

Hosted via FastAPI and integrated with **Gemini 2.5 Pro**, this component handles:

- **Problem validation**
- **Solution correctness**
- **Complexity detection**
- **Brute-force flagging**

4.4 Ranking and Reward System

Rankings are based on:

- Accuracy
- Time and space complexity
- Time taken to solve
- Number of attempts
- Partial scores

Reward distribution:

- Percentage to the **problem creator**
- Share among **top 5** participants
- **Platform fee** is deducted
- Withdrawals available post-contest expiration

4.5 Anti-Cheating Measures

- **Time limits** for problem solving
- **AI-based flagging** of suspicious submissions
- **User-based flagging** system enabled 24 hours after rankings are published

5. Blockchain and Smart Contract Architecture

5.1 CDX Token Contract

- Functions like an ERC-20 token on Stellar
- Enables minting, transfers, and staking

5.2 Transaction Tracker Contract

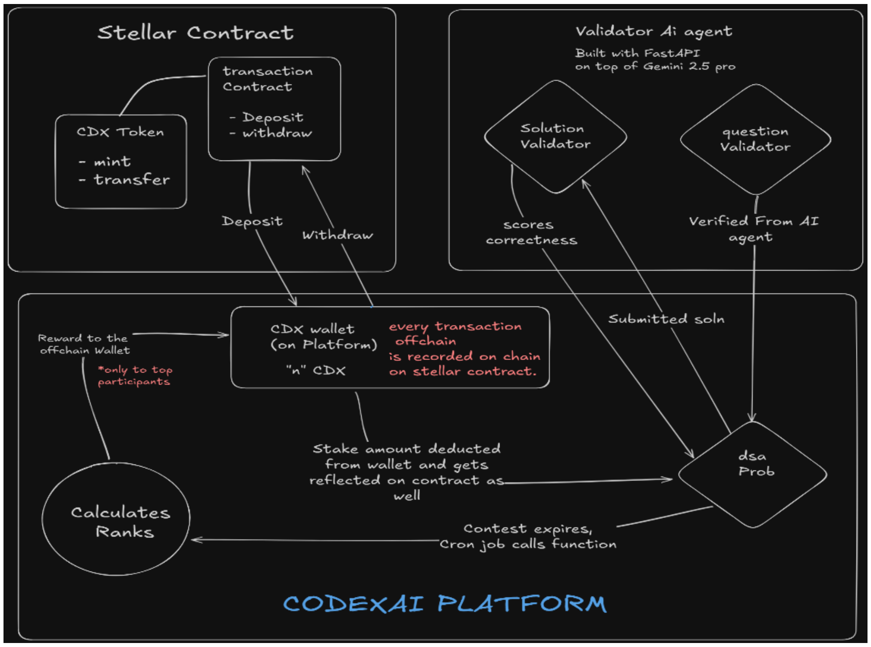
- Maintains a **vault wallet** (CodexAI-owned)
- Maps each deposit to a user's Stellar wallet address
- **Users sign once** to authorize transactions
- **Withdrawals** processed after contest conclusion

5.3 Token Flow

- Tokens staked → Vault Wallet
- Validator checks submission → Ranking generated
- Vault disburses tokens post-ranking

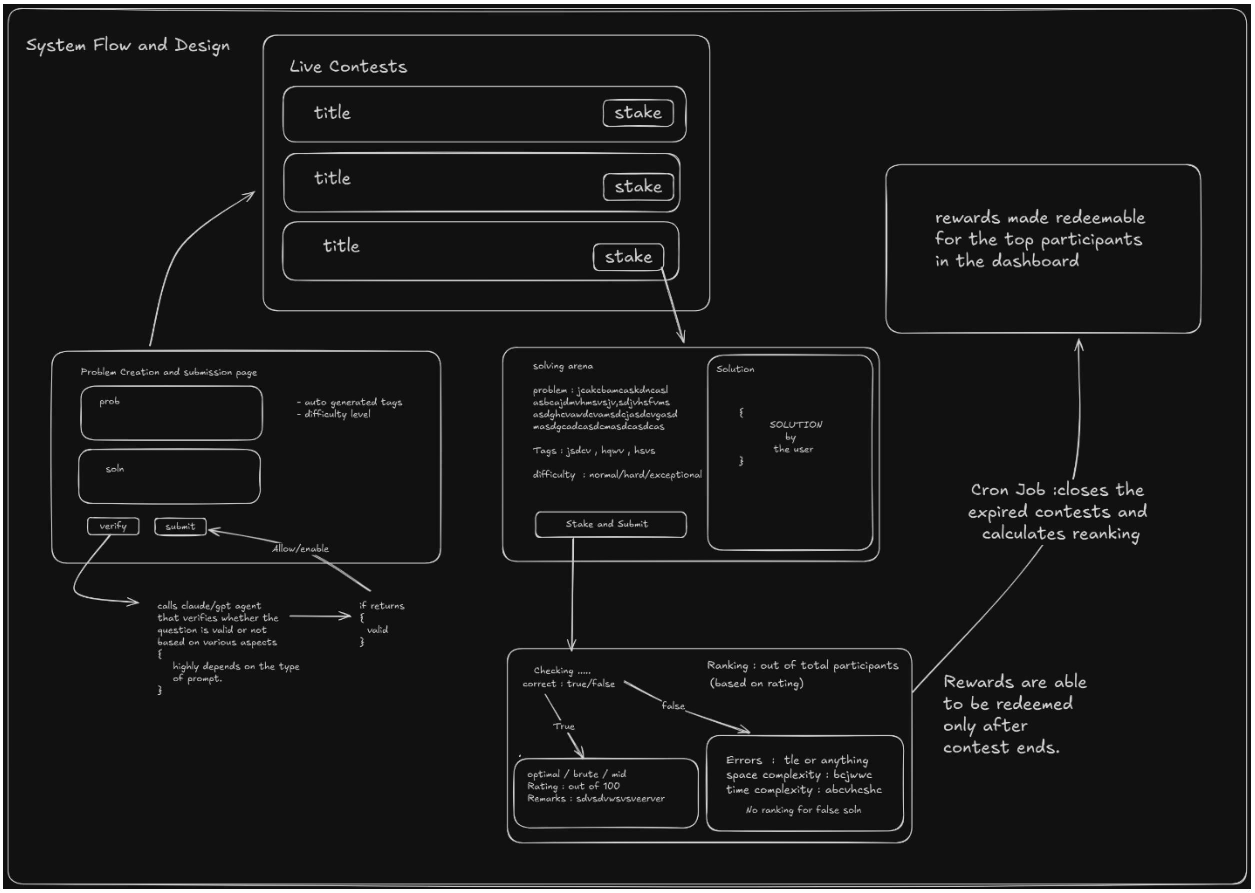
---

6. Architecture Diagram



This diagram outlines how the system's components—including the frontend, smart contracts, AI agent, and blockchain—interact to facilitate contest participation and reward distribution.

7. System Flow


System Flow and Design
Live Contests
title
stake
title
stake
title
stake
rewards made redeemable for the top participants in the dashboard
Problem Creation and submission page
prob
- auto generated tags
- difficulty level
soln
verify
submit
Allow/enable
calls claude/gpt agent that verifies whether the question is valid or not based on various aspects
{
highly depends on the type of prompt.
}
if returns
{
valid
}
solving arena
problem : jcakcbamcaskdncasl asbcajdmvhmsvsjv,sdjvhsfvms asdghcvawdcvamsdcjasdcvgasd masdgcadcasdcmasdcasdcas
Tags : jsdcv , hqwv , hsvs
difficulty : normal/hard/exceptional
Stake and Submit
Solution
{
SOLUTION
by
the user
}
Cron Job :closes the expired contests and calculates reanking
Checking .....
correct : true/false
Ranking : out of total participants (based on rating)
True
false
optimal / brute / mid
Rating : out of 100
Remarks : sdvsdvwsvsveerver
Errors : tle or anything
space complexity : bcjwwc
time complexity : abcvhcshc
No ranking for false soln
Rewards are able to be redeemed only after contest ends.

This flow diagram captures the lifecycle of a contest, from question submission to AI validation, staking, solution review, ranking, and reward distribution.

8. Future Improvements

8.1 Blind Contest Pooling

In future updates, users will **stake on anonymous questions** based on visible tags. After staking, they receive a **random question** with that tag, reducing bias and promoting fairness.

8.2 AI-Powered Practice Arena

A gamified practice environment where:

- AI dynamically generates personalized DSA questions based on user's history
- Real-time difficulty scaling
- Progress reports and XP-based leveling system
- AI-guided hints and feedback to simulate coaching

8.3 Enhanced Cheating Prevention

- **Time-restricted solving** with visible timers
- **Validator AI** detects brute-force or plagiarized code
- **User flagging** mechanism becomes active 24 hours after results are declared

---

9. Non-Functional Requirements

9.1 Performance

- Low-latency contest participation and submission flow
- Efficient token transaction processing via Stellar

9.2 Security

- One-time wallet signing for simplicity and safety
- Centralized vault system reduces risk of contract exploits

9.3 Usability

- Simple UI/UX using Next.js
- Minimal blockchain knowledge required to participate

---

10. Conclusion

CodexAI creates a unique synergy between AI, blockchain, and competitive programming. It rewards talent through transparent token economics and fosters fair competition with AI validation. With future updates like anonymous contests and AI-driven practice arenas, it aims to reshape how developers improve and get recognized.

Links

- **CodexAI github : mohakchakraborty2004/codexAI**
- **CodexAI Agent :** mohakchakraborty2004/Validator-agent: AI agent validater for codexAI
- **CodexAi Stellar Contract :** mohakchakraborty2004/codexAI-Vault-Contract
- **codexCoin CDX Contract :** mohakchakraborty2004/codexcoin