**SecureLoan 🛡️ – No-Collateral, Decentralized Lending on Solana**

Welcome to **SecureLoan**, a decentralized, reputation-based lending platform built on **Solana**. With **SecureLoan**, users can borrow crypto loans without the need for collateral, using reputation scores and real-time income streaming for repayments.

**Table of Contents**

* [Project Overview](#project-overview)
* [Features](#features)
* [Tech Stack](#tech-stack)
* [Setup Instructions](#setup-instructions)
* [RAIR Protocol Integration](#rair-protocol-integration)
* [How It Works](#how-it-works)
* [Team](#team)

**Project Overview**

**SecureLoan** is a decentralized platform that enables no-collateral loans by leveraging on-chain reputation scores and streaming payments on the Solana blockchain. Borrowers can access loans based on their reputation and repay them in real-time, using decentralized methods without needing traditional banks.

**Features**

* **Reputation-Based Lending**: Borrowers are assessed based on their on-chain reputation using decentralized protocols.
* **Income-Based Repayment**: Real-time repayment streams through protocols Solana.
* **No Collateral Required**: The platform is entirely trust- and reputation-based, without traditional collateral.
* **RAIR Protocol Integration**: Tokenized access control for trusted borrowers.
* **User-Friendly Interface**: Simple, easy-to-use interface for managing loans and repayments.

**Tech Stack**

* **Frontend**: React, Solana Web3.js
* **Backend**: Rust, Solana Smart Contracts (Programs)
* **Blockchain**: Solana
* **Reputation System**: On-chain reputation via decentralized protocols
* **Repayment Protocol**: Streaming payments with Solana
* **RAIR Protocol**: Tokenized access control for reputation and loan approval
* **Wallet**: Phantom Wallet or any Solana-compatible Web3 wallet

**Setup Instructions**

Follow these steps to set up the project on your local machine:

**1. Clone the Repository**

git clone https://github.com/your-repo/secureloan.git

cd secureloan

**2. Install Dependencies**

* Install the frontend dependencies:

cd frontend

npm install

* Install the Solana and Rust backend dependencies:

cd ../backend

cargo build

**3. Compile and Deploy Solana Programs**

* Use **Solana CLI** to compile and deploy programs to **Solana Devnet** for testing.

solana-test-validator

cargo build-bpf

solana program deploy ./target/deploy/your\_program.so

**4. Start the Development Server**

* Navigate back to the frontend and start the React application:

cd frontend

npm start

**RAIR Protocol Integration**

**RAIR Protocol** is used to manage tokenized access for trusted borrowers. Here's how it's integrated into SecureLoan:

1. **Tokenized Access Control**: Borrowers with high reputation scores are granted loan access, using RAIR's tokenized system to secure permissions.
2. **Reputation Management**: Borrowers with consistent repayment history are given tokenized access to larger loans.
3. **Seamless Onboarding**: Tokenized access is handled transparently, ensuring a smooth experience for end-users.

**How It Works**

1. **Reputation-Based Loan Application**: Borrowers apply for loans based on their reputation, evaluated through decentralized protocols on Solana.
2. **No Collateral, Trust-Based Loans**: Eligible users can access loans without needing collateral, relying on their on-chain reputation.
3. **Income-Based Repayment**: Repayments are streamed in real-time streaming protocols on Solana.
4. **RAIR Tokenized Access**: Trusted borrowers gain tokenized access to higher loan amounts, ensuring secure and scalable loan issuance.

**Team**

* **Member 1**: Jayesh Laddha
* **Member 2**: Pratyush Kumar
* **Member 3**: Rakshita Jain
* **Member 4**: Mayur Soni