VeryTontine — MVP (Flutter + Sui Move)

🗒️ Project Overview

VeryTontine is a mobile-first decentralized savings circle platform built on the Sui blockchain. It digitizes traditional community savings groups (Tontines / Ikirimba / Chama / Susu) by replacing physical cash boxes and human treasurers with secure on-chain smart contracts.

The MVP focuses on solving the core problems of:

Cash theft

Damaged bank notes

Treasurer errors

Lack of transparency

No financial reputation history

The system uses Sui Move smart contracts as a secure digital vault and Flutter as the mobile frontend to ensure accessibility and fast onboarding.

🎯 MVP Goal

Build a working mobile MVP where a small savings group can:

Create a savings circle

Join a circle

Contribute funds

Store funds securely in a Sui Move contract

Automatically determine payout rotation

View payment status of members

Execute payouts by smart contract logic

No advanced features — only core value.

👥 Target Users

Primary Users:

Community savings groups

Local rotating savings circles

Informal financial groups

User Characteristics:

Mobile-first

Low crypto experience

Needs simple onboarding

Needs strong trust & transparency

🔥 Core MVP Features (ONLY)

1️⃣ Simple User Onboarding

Capability
User opens Flutter app

Signs in with:

zkLogin (preferred) OR

Simple wallet connect

MVP Requirement
No seed phrase handling in MVP UI

Store wallet session locally

2️⃣ Create Savings Circle

Capability
User can create a circle with:

Circle name

Contribution amount

Number of members

Contribution frequency (demo = manual trigger)

Member wallet list (for MVP)

Smart Contract Action
Creates a Circle Object on Sui with:

Circle {

id
creator
members
contribution_amount
round_index
vault_balance
payout_order
}

3 Join Circle

Capability
User joins existing circle

Added to member list

Visible in UI member roster

MVP Simplification
Join via circle ID or invite code

No complex permissions yet

4 Contribute Funds

Capability
User presses:

"Contribute Now"

App executes:

Transaction to Sui Move contract

Transfers contribution amount into vault

Smart Contract Logic
Verify user is member

Verify not already paid this round

Accept deposit

Mark paid status

5 Digital Vault (Move Smart Contract)

Capability
Move contract holds:

All contributions

Locked funds

Round tracking

Member payment status

Rules
Funds cannot be withdrawn manually

Only payout function releases funds

Only contract controls distribution

6️⃣ Automated Rotation Payout
Capability
When all members contribute:

Contract automatically determines next beneficiary

Vault releases total pool to that member

Round increments

MVP Rule
Simple rotation:

beneficiary = members[round_index % members.length]

7️⃣ Circle Dashboard Screen (Flutter)
Show:
Circle name

Contribution amount

Members list

Paid / Pending status

Current round

Next beneficiary

Vault balance

8️⃣ Basic Trust Score (Simple Version)
Capability
Track:

On-time contributions

Missed contributions

MVP Implementation
Stored in Move:

```
TrustScore {
  user
  score
}
```

Rules:

+5 on payment

−10 on missed round

Displayed in UI profile

❌ Not Included in MVP
Do NOT build these yet:

DAO governance

Voting systems

Cross-chain payments

Advanced analytics

Chat system

Loans

Credit scoring engine

Notifications

AI features

🧱 Technical Stack
📱 Frontend — Flutter
Use:

Flutter

Riverpod or Provider

HTTP client for Sui RPC

Wallet / zkLogin SDK

Material UI

Screens:

Login

Create Circle

Join Circle

Circle Dashboard

Contribute Screen

Profile

⛓️ Blockchain — Sui
Smart Contracts
Language:

Sui Move
Modules:

circle.move
vault.move
trust_score.move
Objects:

Circle

Vault

ContributionRecord

TrustScore

🔗 Sui Integration
Use:

Sui TypeScript SDK (via backend helper OR direct)

JSON RPC

Devnet/Testnet

📂 Suggested Repo Structure

```
verytontine_mvp/

  flutter_app/
    lib/
      screens/
      widgets/
      services/
      models/

  sui_move/
    sources/
      circle.move
      vault.move
      trust_score.move

  docs/
    architecture.md
    contract_design.md
```