Sprint 1: Project Planning & Initial Design

Goal: Define project scope, set up repository, and establish architecture. No coding beyond setting up environments.

Tasks

Administrative:

- Schedule six sprint meetings with the TA.
- Finalize Lean Canvas and paper prototype designs.
- Establish GitLab repository and invite all team members.
- Create product backlog with all required features in GitLab Issues.

Architecture & Design (Backend 1, Backend 2, Blockchain, Frontend, Web Design)

- Define project architecture based on:
  - Frontend: Next.js (deployed on Vercel).
  - Backend: Flask (Python-based REST API).
  - Database: SQLite (local for development, Sepolia for blockchain transactions).
  - Blockchain: Ethereum Sepolia (via Hardhat).
- Document database schema (Users, Investments, Loans, Voting, Collateral).
- Define API specifications based on project requirements (RESTful API structure).
- Plan testing approach (unit testing, integration testing, security testing).

Deliverables

- Lean Canvas & Paper Prototypes.
- GitLab repository with team members, backlog, and issues.
- Architecture & Database Design Document.
- Initial Sprint Status Report.

Sprint 2: Core Infrastructure & Authentication

Goal: Build the foundation for authentication and initial API structure.

Tasks

Backend (Backend 1 & Backend 2)

- Implement Flask-based authentication API:
  - `POST /api/auth/register`
  - `POST /api/auth/login` (JWT authentication).
- Set up Prisma ORM with SQLite for user storage.
- Deploy backend on VCM (Duke's Virtual Computing Machine).

Frontend (Frontend Developer, Web Designer)

- Implement Next.js user authentication pages.
- Set up API integration for login/registration.

Blockchain (Blockchain Developer)

- Set up Hardhat environment for smart contract deployment.
- Implement basic fund escrow contract and deploy on Sepolia.

Testing & Security

- Write unit tests for authentication API (pytest).
- Implement basic security tests (JWT token validation).

Deliverables

- Working authentication system (API + frontend).
- Deployed smart contract escrow (test version).
- Database integration with SQLite + Prisma.

Sprint 3: Investment & Proposal System

Goal: Enable investors to browse and invest in loan proposals.

Tasks

Backend (Backend 1 & Backend 2)

- Implement investment proposal API:
  - `GET /api/proposals` (Retrieve all loan proposals).
  - `POST /api/proposals` (Create a new loan proposal).
- Implement investment API:
  - `POST /api/investments` (Allow users to invest in a project).
  - `GET /api/investments/{userId}` (Retrieve user's investments).

Frontend (Frontend Developer, Web Designer)

- Implement investment proposal form (for borrowers).
- Build investor dashboard (list of projects & investment options).
- Create investment transaction page (user selects amount to invest).

Blockchain (Blockchain Developer)

- Extend fund escrow contract to hold investor contributions.
- Deploy updated smart contract on Sepolia.
- Implement fund lock mechanism until project goal is met.

Testing & Security

- Unit tests for investment proposal & investment APIs.
- Smart contract tests for fund storage and retrieval.

Deliverables

- Working investment system (backend + UI).
- Smart contract supporting fund storage & investments.
- Deployed APIs for investment tracking.

Sprint 4: Borrower Fund Withdrawal & Voting Mechanism

Goal: Enable borrowers to withdraw funds with investor approval voting.

Tasks

Backend (Backend 1 & Backend 2)

- Implement fund withdrawal API:
  - `POST /api/withdrawals` (Request fund withdrawal).
- Implement voting API:
  - `POST /api/vote` (Investors approve/deny fund withdrawal).
- Extend investment tracking API to include voting results.

Frontend (Frontend Developer, Web Designer)

- Implement borrower dashboard (request withdrawals).
- Build voting system for investors.
- Display withdrawal request status.

Blockchain (Blockchain Developer)

- Extend smart contract to support withdrawal requests.
- Implement majority-rule voting mechanism.
- Deploy updated contract on Sepolia.

Testing & Security

- Unit tests for fund withdrawal API & voting system.
- Smart contract tests for fund approval logic.

Deliverables

- Fund withdrawal & voting system (backend + frontend).
- Smart contract handling investor voting & approvals.
- Fully tested end-to-end investment flow.

Sprint 5: Interest Distribution & Repayments

Goal: Automate interest distribution and enable borrower repayments.

Tasks

Backend (Backend 1 & Backend 2)

- Implement loan repayment API:
  - `POST /api/repayments` (Borrowers repay funds).
- Implement interest distribution logic (calculates and distributes earnings).

Frontend (Frontend Developer, Web Designer)

- Implement loan repayment page for borrowers.
- Build investor earnings dashboard.

Blockchain (Blockchain Developer)

- Modify smart contract to handle interest payments.
- Implement auto-distribution logic for investors.
- Deploy updated contract on Sepolia.

Testing & Security

- Smart contract repayment & interest distribution tests.
- API integration tests for repayment system.

Deliverables

- Automated investor interest distribution via smart contract.
- Loan repayment system with UI & backend integration.
- Fully tested end-to-end lending cycle.

Sprint 6: Security, KYC, and Deployment

Goal: Final testing, security audits, and live deployment.

Tasks

Backend (Backend 1 & Backend 2)

- Implement KYC/AML verification API (Onfido or free alternative).
- Deploy final backend version on VCM.

Frontend (Frontend Developer, Web Designer)

- Implement KYC upload page.
- Perform final UI/UX optimizations.

Blockchain (Blockchain Developer)

- Conduct security audit on smart contracts.
- Deploy final contract on Sepolia.

Testing & Deployment

- Full end-to-end testing (investment, voting, repayment).
- Deploy MVP to Vercel (frontend) + VCM (backend).

Deliverables

- Fully functional MVP.
- Live deployment on Vercel & VCM.
- Final sprint status report.