Hawamoni Project Overview:

Project Summary
Hawamoni is a blockchain-based treasury management system for student groups, campus societies, project teams, and small co-founder groups, leveraging Solana blockchain and Solana Pay for secure, transparent financial governance.

Core Team Structure

Frontend Developer
- Implement Next.js frontend with Solana wallet integration
- Build responsive UI components for group management, deposits, withdrawal requests
- Create QR code generation for Solana Pay deposits
- Implement notification center and approval interfaces

Backend Developer
- Develop Spring Boot REST API
- Implement database schema and ORM models
- Create authentication system using wallet signatures
- Build SMS notification service integration
- Implement webhook handling for transaction confirmations

Blockchain Developer
- Design and implement Solana Anchor program for treasury management
- Develop on-chain governance mechanism with 80% approval threshold
- Create secure transaction handling for deposits and withdrawals
- Implement Solana Pay integration for QR-based deposits
- Set up and manage blockchain infrastructure and RPC connections

AI Engineer
- Develop fraud detection system
- Implement transaction categorization and analytics
- Create spending insights generation
- Build vector database for financial pattern recognition
- Design and implement API endpoints for AI services

Revised Implementation Plan

Phase 1: Setup & Foundation (Day 1 and 2)
1. **Repository Setup**
   - Create GitHub repository with proper branching strategy
   - Set up project board with issues organized by component
   - Configure GitHub Actions for CI/CD

2. **Environment Configuration**
   - Set up development environments for all team members
   - Configure cloud services and development databases
   - Set up Solana development environment

3. **Core Architecture**
   - Define API contracts between services
   - Create database schema
   - Design Solana program account structure

Phase 2: Core Functionality (Day 3 and 4 and 5)
1. **User Authentication & Group Management**
   - Frontend: User interface for auth and group creation
   - Backend: Authentication endpoints and group management APIs
   - Blockchain: On-chain group representation
   - AI: Initial data collection for model training

2. **Transaction System**
   - Frontend: UI for deposits and withdrawal requests
   - Backend: Transaction APIs and database interactions
   - Blockchain: Smart contract for treasury and approval logic
   - AI: Setup monitoring for transaction patterns

Phase 3: Integration & Features (Day 6 and 7)
1. **Notification & Approval System**
   - Frontend: Notification center and approval interface
   - Backend: SMS service integration
   - Blockchain: Execute transactions based on approval threshold
   - AI: Anomaly detection for suspicious transactions

2. **Analytics & Insights**
   - Frontend: Dashboard for financial insights
   - Backend: Data aggregation endpoints
   - Blockchain: On-chain analytics extraction
   - AI: Spending pattern analysis and recommendations

Phase 4 :
Pitch deck, video demo and presentation preparation

Launching an production stage:
Phase 5: Testing & Refinement
1. **Security & Performance**
   - Frontend: Security audit and performance optimization
   - Backend: Load testing and security hardening
   - Blockchain: Smart contract audit and optimization
   - AI: Model validation and tuning

2. **User Testing**
   - Conduct usability testing with target users
   - Refine UX based on feedback
   - Fix critical issues

Phase 6: Launch Preparation
1. **Documentation & Deployment**
   - Frontend: Deployment to production hosting
   - Backend: Production environment setup
   - Blockchain: Mainnet deployment planning
   - AI: Production model deployment

2. **Launch**
   - Final QA and user acceptance testing
   - Public release
   - Monitoring and support