

# PART A

## User Stories & On-Chain Requirements

### Project value proposition (refined)

A decentralized, Solana-native community forum purpose-built for Web3 Builders - developers, designers, and entrepreneurs, offering tokenized rewards (via custom tokens), gamified engagement (leaderboards, badges), and a low-cost, high-speed environment powered by Solana. Beyond just incentivizing participation, the platform aims to foster high-quality, sustained collaboration through a thoughtfully designed token economy and builder reputation system. By simplifying onboarding for non-technical users and focusing specifically on the Solana ecosystem's needs, it offers a focused, engaging alternative to both general-purpose Web2 forums and broad Web3 social platforms.

## PART A

### Initial User & Function Mapping




#### 1. Manual User Brainstorming

- Direct Users
  - Solana developers
  - Web3 designers
  - Blockchain enthusiasts
  - Content creators and educators
  
- Indirect Users / Beneficiaries
  - Solana foundation
  - Solana validators
  - Token holders and investors
  - Existing solana projects
  - Web3 service providers
  
- Administrators/Moderators

- Community managers
- Technical moderators
- Content curators
- Event Coordinators
- Stakeholders
- - Solana foundation
  - VC firms
  - Enterprise partners
  - Educational institutions
  - Web2 companies
  - Content creators

## 2. AI-Assisted User Prioritization

### Final prioritized list of User types

- Solana developers
  - Decision 
  - **WHY**
    - They are the primary value drivers, they engage deeply, create technical content, test features, and provide critical feedback
    - A community without devs especially in web3 lacks depth. Their engagement legitimize the forum as a serious place for technical discussions
- Content creators and educators
  - Decision 
  - **WHY**
    - Help translate technical content into digestible knowledge which widens the reach
    - Bring audience from X, YT, Newsletters and useful for acceleration of the platform
- Blockchain enthusiasts
  - Decision 
  - **WHY**

- They are the daily active users who read, react, comment and contribute from a non-technical lens
  - Their curiosity and dedication often lead to long term loyalty and organic content sharing
- Community Managers
  - Decision ⚠
  - **WHY**
    - While valuable, they are multipliers and scalers, not initial traction generators
    - In this, we need raw engagement and authenticity more than professionally managed communities

### 3. Core Function Mapping

Based on my research on Solana development requirements, Web3 content creation needs, blockchain enthusiast engagement patterns, and token economy design principles, I can now provide comprehensive recommendations for key functions and interactions for each prioritized user type in your Solana-native community forum.

### Key Functions and Interactions for Your Solana Community Forum User Types

Based on your value proposition as a decentralized, Solana-native community forum with tokenized rewards and gamified engagement, here are the essential functions and interactions each prioritized user type would need:

#### 1. Solana Developers

##### Core Technical Functions

##### Code Sharing & Repository Integration

- Advanced code snippet editor with Solana/Rust syntax highlighting
- Integration with GitHub for seamless project sharing and version control
- Smart contract deployment testing environment
- Access to Solana development tools like Anchor framework documentation

##### Technical Discussion Forums

- Specialized channels for different Solana development areas (DeFi, NFTs, gaming, infrastructure)
- Advanced search functionality to find solutions to specific technical problems
- Integration with Solana's official documentation and RPC endpoints

##### Development Environment Access

- Direct integration with Solana CLI tools and testnet/devnet access
- Sandboxed environment for testing smart contracts
- Performance optimization guides and compute usage analytics

### Token Economy Interactions

#### **Contribution-Based Rewards**

- Higher token multipliers for solving complex technical challenges
- Reputation-based token earning rates that increase with proven expertise

#### **Technical Validation System**

- Peer review mechanisms where other developers validate technical contributions
  - Community voting on the quality and accuracy of technical content
  - Automated token distribution based on code contribution metrics
- Gamification Elements

#### **Developer Achievement System**

- Badges for different technical milestones (first deployed contract, bug fixes, optimizations)
- Leaderboards for most helpful technical contributors
- Special NFT rewards for significant contributions to the Solana ecosystem

## **2. Content Creators and Educators**

### Content Creation Tools

#### **Multi-Format Content Support**

- Rich text editor for tutorials and educational posts
- Video upload and streaming capabilities for technical tutorials
- Interactive code examples with live execution
- Template systems for consistent educational content formatting

#### **Educational Content Organization**

- Curriculum builders for structured learning paths
- Progress tracking for students following educational content
- Content categorization by difficulty level and topic area

#### **Decentralized Content Ownership**

- IPFS integration for permanent, censorship-resistant content storage
- NFT-based content ownership verification
- Smart contract-based copyright protection with timestamping

### Monetization Functions

#### **Direct Fan Support**

- Cryptocurrency tipping systems for quality educational content
- Subscription models using native forum tokens
- Crowdfunding mechanisms for large educational projects

#### **Token-Gated Premium Content**

- Exclusive content access for token holders
- Tiered access levels based on token ownership or community participation
- Revenue sharing from premium content subscriptions

### Community Building Tools

#### **Audience Engagement**

- Live streaming capabilities for educational events
- Community polls and feedback systems
- Direct messaging with students and followers

- Event hosting and workshop organization tools

### **3. Blockchain Enthusiasts**

#### Discovery and Engagement Functions

##### **Content Consumption Interface**

- Personalized content feeds based on interests and interaction history
- Advanced filtering and search across all content types
- Bookmarking system for saving interesting discussions and resources

##### **Social Interaction Tools**

- Commenting and discussion threading on all content types
- User-to-user messaging and connection building
- Group formation around specific interests or projects

##### **News and Trend Tracking**

- Real-time updates on Solana ecosystem developments
- Curated news feeds from trusted sources
- Price and market data integration for relevant tokens

#### Community Participation

##### **Voting and Governance**

- Community polls on forum features and improvements
- Weighted voting based on reputation and token holdings
- Proposal submission for community initiatives

##### **Curation and Moderation**

- Content quality rating systems
- Community-driven content moderation through reporting
- Crowdsourced fact-checking for news and claims

#### Token Economy Participation

##### **Engagement-Based Rewards**

- Daily login bonuses and streak rewards
- Tokens for quality comments, shares, and community interactions
- Referral bonuses for bringing new quality members

##### **Gamified Learning**

- Quiz systems about Solana and blockchain technology with token rewards
- Achievement badges for different levels of platform engagement
- Seasonal challenges and competitions with NFT prizes.

## **Cross-User Type Integration Features**

#### Universal Functions Needed by All User Types

##### **Identity and Reputation System**

- Solana wallet-based authentication and identity verification
- Cross-platform reputation aggregation using decentralized identity
- Privacy-preserving reputation scores based on on-chain activity

##### **Token Economy Infrastructure**

- Unified token wallet integration with major Solana wallets

- Transparent tokenomics dashboard showing earning and spending opportunities
- Automated smart contracts for reward distribution

### **Communication and Networking**

- Multi-channel communication (forums, chat, video calls)
- Professional networking features for collaboration
- Event coordination and meetup organization tools

### **Platform Governance**

- DAO-style governance where all user types can participate in platform decisions
- Token-weighted voting on major platform changes
- Community treasury management for platform development funds

## **Implementation Priorities**

Based on your MVP approach, I recommend implementing these functions in order of priority:

### **Phase 1: Core Engagement**

- Basic forum functionality with Solana wallet integration
- Token reward system for quality contributions
- Content creation tools for developers and educators

### **Phase 2: Advanced Features**

- Reputation system and gamification elements
- Premium content and monetization tools
- Advanced search and discovery features

### **Phase 3: Community Scaling**

- DAO governance implementation
- Cross-platform integrations
- Advanced analytics and personalization

This comprehensive function mapping ensures each user type has meaningful ways to engage while creating network effects that benefit the entire community ecosystem.

## **4. Deriving Core POC Requirements**

### **Manual Task**

Q ) Identify the top 1-2 most critical user stories or interaction paths essential for a POC.

1. Solana Developers-discussion threads-validation and reward with peer upvoting.
2. Content Creators and Educators - Rich content posting - Markdown/text support

## AI Task

# Technical Requirements for Your Solana Community Forum Proof-of-Concept

Based on the two critical user interactions — (1) discussion threads with peer validation/rewards for Solana developers, and (2) rich content posting with Markdown support for educators — here are the essential technical requirements for your proof-of-concept.

## Core Infrastructure Requirements

### Blockchain Integration Layer

#### Solana Network Connection

- RPC endpoint integration with Solana mainnet/devnet for real-time blockchain interactions
- Wallet adapter supporting major Solana wallets (Phantom, Solflare, Backpack)
- Smart contract deployment capability for custom token rewards and governance
- Transaction signing and verification mechanisms for user authentication

#### Identity Management

- Solana wallet-based authentication system eliminating traditional username/password
- Public key-based user identification and reputation tracking
- Decentralized identity verification using on-chain transaction history

### Database Architecture

#### Hybrid Storage Solution

- Traditional database (PostgreSQL/MongoDB) for forum metadata, user preferences, and real-time interactions

- IPFS integration for decentralized content storage ensuring censorship resistance
- Smart contract state management for token balances, voting records, and reputation scores

## **User Interaction #1: Discussion Threads with Peer Validation**

### **Forum Infrastructure**

#### **Thread Management System**

- Hierarchical comment threading with unlimited depth
- Real-time updates using WebSocket connections
- Full-text search across all discussions
- Content categorization by Solana dev topics (DeFi, NFTs, infrastructure, etc.)

### **Peer Upvoting Mechanism**

#### **Voting Smart Contract**

- Immutable on-chain upvote/downvote recording
- Weighted voting based on user reputation and community stake
- Anti-spam mechanisms (e.g., time-locks, stake requirements)
- Transparent, publicly verifiable vote tallying

#### **Validation Logic**

- Peer review of technical content
- Quality scoring via votes, engagement, and accuracy
- Auto-flagging low-quality/spam posts

### **Token Reward Distribution**



### **Automated Reward System**

- Smart contract-based token distribution for validated posts
  - Dynamic rewards based on quality, complexity, and engagement
  - Escrow and delay mechanisms for fair distribution
  - Integration with SPL for custom token minting and management
- 

## **User Interaction #2: Rich Content Posting with Markdown Support**

### **Content Creation Engine**

#### **Markdown Processing Infrastructure**

- Real-time GFM/CommonMark-compliant Markdown parser
- WYSIWYG editor with live preview
- Syntax highlighting for Rust, JavaScript, and Solana code
- LaTeX rendering for blockchain/math equations

### **Rich Media Support**

- Image upload with CDN distribution
- Embedded video tutorials
- Interactive code runner for smart contract testing
- File attachment system for dev resources

## **Content Storage and Versioning**

### **Decentralized Content Management**

- IPFS-based storage for permanent archival
- Version history and content diffs
- Content hash verification
- Redundant backup across IPFS nodes

## **Technical Implementation Stack**

### **Frontend Requirements**

#### **User Interface Framework**

- React.js or Next.js frontend
- Tailwind CSS for consistent mobile-first design
- WebSocket integration for real-time updates
- PWA support for mobile use

### **Backend Architecture**

#### **API and Server Infrastructure**

- Node.js/Express.js or Rust for performant APIs
- REST or GraphQL APIs for flexible data handling
- Redis caching for performance
- Rate limiting and DDoS protection

### **Security and Performance**

#### **Security Measures**

- End-to-end encryption
- Input sanitization / XSS protection
- Multi-signature wallet access for admins
- Security audits and penetration testing

### **Scalability Considerations**

- Horizontal scaling support
  - CDN for global content access
  - Sharding for large dataset management
  - Aggressive caching for frequently accessed content
- 

## **Minimum Viable Technical Setup**

### **Phase 1 – Essential Features**

1. Thread creation and commenting
2. Solana wallet login and user profiles
3. Basic on-chain upvoting
4. Markdown editor with live preview
5. Token rewards for validated posts

### **Phase 2 – Enhanced Functionality**

1. Advanced search + categorization
2. Reputation + weighted voting
3. IPFS integration

4. Media uploads and file sharing
5. Real-time notifications and feeds

## PART B

### Adversarial Analysis & Granularity Check

#### 1. Critique & Refine User Stories / Requirements:

### Analysis: User Stories vs. Value Proposition Alignment

Your user stories and technical requirements show strong alignment with your refined value proposition, but there are several gaps and areas that need more granular specification for successful implementation.

---

### What Hits the Mark

#### Strong Value Proposition Alignment

- **Solana-Native Focus**

Your technical requirements correctly prioritize Solana-specific integrations (RPC endpoints, wallet adapters, SPL tokens) that differentiate from generic Web3 platforms.

- **Tokenized Rewards**

The contribution-based reward system and automated token distribution directly support your core value proposition of incentivizing quality participation.

- **Gamified Engagement**

Developer achievement systems, leaderboards, and NFT rewards effectively translate the “gamified engagement” promise into concrete features.

- **High-Quality Collaboration**

Peer review mechanisms and reputation-based systems align with fostering

“sustained collaboration through a thoughtfully designed token economy.”

---

## **Critical Gaps in User Stories**

### **Missing: Simplified Onboarding for Non-Technical Users**

Your value proposition explicitly mentions “simplifying onboarding for non-technical users,” but your user stories lack this crucial element:

#### **Missing Components:**

- Guided wallet setup tutorials
  - Web3 concept explanations for beginners
  - Simplified UI modes for different user types
  - Progressive feature disclosure based on user experience level
  - Onboarding flow that doesn’t assume blockchain knowledge
- 

### **Insufficient Designer User Stories**

Your value proposition targets “developers, designers, and entrepreneurs,” but your user stories heavily skew toward developers.

#### **Designers need:**

- Visual portfolio sharing capabilities
  - Design critique and feedback systems
  - UI/UX resource libraries specific to Solana dApps
  - Design challenge competitions with token rewards
- 

### **Limited Entrepreneur-Focused Features**

### Entrepreneurs require:

- Project showcase and funding discovery tools
  - Partnership matching based on skills and interests
  - Business model validation through community feedback
  - Investor connection mechanisms within the Solana ecosystem
- 

## Technical Requirements: Granularity Assessment

### Well-Specified Areas

- **Blockchain Integration**

Your requirements clearly identify specific technologies (SPL tokens, RPC endpoints, wallet adapters) that can be directly implemented.

- **Content Management**

The Markdown processing and IPFS integration requirements are concrete enough for development planning.

---

### Areas Lacking Implementation Detail

#### Database Schema Gaps

Your requirements mention “traditional database” but lack specifics:

#### Missing Schema Elements:

- User profile structure (reputation scores, badge inventories, token balances)
- Thread/post relationship models with voting metadata
- Content categorization taxonomy for specialized Solana topics
- Analytics tracking schema for engagement metrics

---

## API Endpoint Specifications Missing

### Required but Undefined Endpoints:

- `/api/posts/{id}/vote` — Voting mechanism with reputation weighting
- `/api/users/{id}/reputation` — Reputation calculation and display
- `/api/tokens/rewards/calculate` — Dynamic reward computation
- `/api/content/validate` — Peer review submission system

---

## Smart Contract Program Architecture Undefined

### Missing Blockchain Components:

- Voting program structure for immutable upvote/downvote recording
- Token minting program for rewards distribution
- Reputation tracking program linking on-chain activity to forum participation
- Governance program for community decision-making



## Critical Missing Elements



### Token Economy Design Details

While you mention tokenized rewards, the economic model lacks specificity:

#### Undefined Elements:

- Token supply mechanics and inflation control
- Reward calculation algorithms and multiplier systems
- Token utility beyond rewards (governance voting weight, premium features)

- Economic sustainability model preventing token devaluation
- 

## Reputation System Architecture

### Missing Specifications:

- How on-chain transaction history translates to reputation scores
  - Reputation decay mechanisms to maintain relevance
  - Cross-platform reputation aggregation methodology
  - Reputation-based feature access controls
- 

## Community Governance Structure

### Unclear Areas:

- DAO implementation for platform governance decisions
  - Community proposal submission and voting processes
  - Treasury management for platform development funding
  - Dispute resolution mechanisms for content moderation
- 



## Recommendations for Implementation Readiness



### Immediate Actions Required

#### 1. Define Core Data Models

Create detailed database schemas for users, posts, votes, tokens, and reputation before development begins.

#### 2. Specify Smart Contract Architecture



Design the specific Solana programs needed for voting, rewards, and reputation tracking with clear function signatures.

### 3. Map User Journey Flows

Create detailed user flows showing how non-technical users progress from wallet creation to active participation.

---

#### Phase 1 Refinements

- **Simplify Initial Scope**


Your Phase 1 features are appropriate, but add explicit onboarding flow requirements.

- **Add Designer/Entrepreneur Stories**

Include at least 2–3 user stories for each user type to validate your “Web3 Builders” positioning.

- **Define Minimum Viable Token Economy**

Specify the exact reward calculation for your proof-of-concept to enable smart contract development.

 **Conclusion:** Your foundation is solid and aligns well with your value proposition, but implementing these missing elements will be crucial for creating a platform that truly serves the full spectrum of “Web3 Builders” while maintaining the simplified onboarding experience you’ve promised.

## AI output documentation and analysis

### Identified Gaps in analysis

Area	Output	Analysis
<b>Onboarding for Non-Technical Users</b>	Missing from user stories	The forum's growth depends on including beginners; currently, dev-heavy.
<b>Entrepreneur Features</b>	Weak coverage	Entrepreneurs need discovery, funding, and networking options.
<b>Database Schema</b>	Too abstract	Must model users, posts, votes, tokens, and reputation.
<b>API Endpoints</b>	Not defined	Missing for key actions like voting and reward distribution.
<b>Smart Contract Architecture</b>	Undefined	Voting, minting, reputation, and governance contracts are critical.
<b>Token Economy Design</b>	Lacks mechanics	Needs supply logic, multipliers, inflation control, and utility.

### Refined User stories

#### Developer

- As a Solana developer, I want to showcase code snippets with Anchor/Rust highlighting so I can get peer feedback.
- As a developer, I want to earn tokens for helping others debug smart contract issues, so I feel incentivized to contribute.

#### Entrepreneur

- As a Web3 founder, I want to post about my project and find collaborators based on skills/tokens held.
- As a startup founder, I want to validate my idea through community polling and feedback threads before seeking funding.

## Non technical users

- As a new user, I want a step-by-step onboarding wizard that helps me create a wallet and explains Solana basics.
- As a beginner, I want to start with a simplified interface and unlock advanced features as I become more confident.

## Refined Technical requirements

### DB Schema

- Users
  - id, wallet\_address, username, bio, reputation, token\_balance, badges, role, created\_at
- Posts
  - id, author\_id (FK), title, content, category\_id, votes\_up, votes\_down, created\_at, ipfs\_hash
- Votes
  - id, user\_id (FK), post\_id (FK), vote\_type (up/down), weight (reputation-weighted), created\_at
- Tokens
  - id, user\_id (FK), type (reward, governance), amount, transaction\_id
- Categories
  - id, name, description (e.g., NFTs, DeFi, Tooling)

### API Endpoints

**POST** /api/posts/:id/vote

**POST** /api/content/peer-review

**POST** /api/tokens/rewards/calculate

**POST** /api/users/onboard

**GET** /api/discover/projects

### Smart contract overview

1. Voting program
2. Token reward program

### Token economy

#### ***Distribution logic***

- +5 tokens per accepted answer
- +1 token per upvote
- Weekly leadership bonuses

### Reputation System

#### ***Score sources***

- Votes received
- Peer reviewed content approval
- On chain activity participation

#### ***Decay logic***

- Reputation decays 5% monthly unless active

### OnBoarding flow

-> landing -> Role selection -> wallet tut -> User type detected (eg: beginner)-> simple ui enabled -> first token earned(welcome bonus) -> first task: comment or vote -> Guided tour ends there-> collect feedback

## PART C

### Granularity & Clarity Refinement

#### 1. Final Manual Review & Refinement

##### Refined User Stories

##### 1. Solana Developers (De-Jargoned & Atomic)

###### Code Sharing & Discussion

- Story 1: As a developer, I want to write code in a text box that automatically colors different parts of my code (like making keywords blue) so other developers can read it easily.
- Story 2: As a developer, I want to click a button to connect my GitHub account so I can share my existing projects with the community.
- Story 3: As a developer, I want to post a question about my code problem in a specific discussion area (like “DeFi” or “NFTs”) so experts in that area can help me.
- Story 4: As a developer, I want to search for solutions using keywords so I can find answers to technical problems quickly.

###### Rewards & Recognition

- Story 5: As a developer, I want to receive community points when other developers vote my answer as helpful so I feel motivated to help others.
- Story 6: As a developer, I want other experienced developers to review my code contributions so I know my solutions are correct.
- Story 7: As a developer, I want to earn a digital badge when I help solve my first complex problem so I can show my expertise to the community.
- Story 8: As a developer, I want to see my name on a leaderboard of most helpful contributors so I can build my reputation in the community.

##### 2. Content Creators & Educators (De-Jargoned & Atomic)

###### Content Creation

- Story 9: As an educator, I want to write tutorials using a rich text editor that shows me exactly how my content will look as I type.
- Story 10: As an educator, I want to upload videos of my tutorials so students can watch and learn visually.
- Story 11: As an educator, I want to create interactive code examples that students can run and modify so they can practice while learning.
- Story 12: As an educator, I want to organize my tutorials into learning courses with multiple lessons so students can follow a structured path.

###### Monetization & Ownership

- Story 13: As an educator, I want students to be able to tip me with digital currency for quality content so I can earn money from my teaching.
- Story 14: As an educator, I want to create premium content that only paying subscribers can access so I can monetize advanced tutorials.
- Story 15: As an educator, I want to store my content permanently on a decentralized network so no one can delete or censor my work.

#### Community Building

- Story 16: As an educator, I want to host live streaming events for my students so I can teach in real-time and answer questions.
- Story 17: As an educator, I want to send direct messages to my students so I can provide personalized guidance.

### 3. Blockchain Enthusiasts (De-Jargoned & Atomic)

#### Content Discovery

- Story 18: As an enthusiast, I want to see a personalized feed of content based on my interests so I don't have to search for relevant discussions.
- Story 19: As an enthusiast, I want to save interesting posts to read later so I can organize content that matters to me.
- Story 20: As an enthusiast, I want to filter discussions by topic and difficulty level so I can find content appropriate for my knowledge level.

#### Social Interaction

- Story 21: As an enthusiast, I want to comment on posts and reply to other comments so I can join conversations that interest me.
- Story 22: As an enthusiast, I want to send private messages to other community members so I can network and build relationships.
- Story 23: As an enthusiast, I want to join groups focused on specific topics so I can connect with like-minded people.

#### Participation & Learning

- Story 24: As an enthusiast, I want to vote on community suggestions for new features so I can help shape the platform.
- Story 25: As an enthusiast, I want to take quizzes about blockchain topics and earn points so I can test and improve my knowledge.
- Story 26: As an enthusiast, I want to report inappropriate content so I can help keep the community safe and high-quality.

### 4. New Users (Addressing Missing Onboarding)

#### Simplified Onboarding

- Story 27: As a new user, I want to see a welcome screen that explains what this platform does in simple terms so I understand its purpose.

- Story 28: As a new user, I want a step-by-step tutorial on creating a digital wallet so I can join the community without technical confusion.
- Story 29: As a new user, I want to choose my experience level (beginner/intermediate/advanced) so the platform can show me appropriate content.
- Story 30: As a new user, I want to start with a simplified interface that only shows basic features so I don't feel overwhelmed.
- Story 31: As a new user, I want to receive welcome points just for joining so I can immediately experience the reward system.

## 5. Designers (Previously Missing User Type)

### Portfolio & Feedback

- Story 32: As a designer, I want to upload images of my UI/UX designs so I can showcase my work to the community.
- Story 33: As a designer, I want other designers to comment on and critique my work so I can improve my designs.
- Story 34: As a designer, I want to access a library of design resources specific to blockchain apps so I can create better user experiences.

## 6. Entrepreneurs (Previously Missing User Type)

### Project Showcase & Networking

- Story 35: As an entrepreneur, I want to create a project profile that describes my startup idea so I can attract collaborators and feedback.
- Story 36: As an entrepreneur, I want to search for potential team members based on their skills and community reputation so I can build my team.
- Story 37: As an entrepreneur, I want to post polls asking the community to validate my business ideas so I can make informed decisions.

## PART D

### Defining Potential On-Chain Requirements

#### 1. Brainstorming On-Chain Requirements for Each User Story

- Solana developers

##### **Story 1: Write code in a text box with automatic coloring**

###### **Potential On-Chain Requirements**

- > No direct on-chain requirements (front-end syntax highlighting)
- > Optional: Store code snippets on IPFS with hash recorded on-chain for permanent archival
- > Optional: Create on-chain metadata linking user wallet to code contribution for reputation tracking

##### **Story 2: Connect GitHub account to share existing projects**

###### **Potential On-Chain Requirements**

- > Create on-chain account linking Solana wallet address to GitHub username
- > Store GitHub integration status and permissions on-chain for decentralized verification
- > Record GitHub repository links in user's on-chain profile for portfolio verification

##### **Story 3: Post questions in specific discussion areas (DeFi, NFTs, etc.)**

###### **Potential On-Chain Requirements**

- > Create on-chain post account with category metadata (DeFi, NFT, Gaming, etc.)
- > Store post creator's wallet address for ownership verification
- > Initialize post with zero votes and zero replies counter
- > Record timestamp for chronological ordering
- > Store IPFS hash of post content for decentralized storage

##### **Story 4: Search for solutions using keywords**

###### **Potential On-Chain Requirements**



- > No direct on-chain requirements (search handled by indexing service)
- > *Optional:* Store search analytics on-chain for decentralized platform analytics
- > *Optional:* Record popular search terms for community-driven feature development

### **Story 5: Receive community points when others vote answer as helpful**

#### **Potential On-Chain Requirements**

- > Create SPL token mint for community points with controlled supply
- > Implement voting smart contract that records upvotes/downvotes immutably
- > Create token transfer function that rewards points based on vote weight
- > Store voter's wallet address and vote timestamp to prevent double-voting
- > Update recipient's token balance automatically when vote threshold is met

### **Story 6: Have experienced developers review code contributions**

#### **Potential On-Chain Requirements**

- > Create reviewer assignment system based on on-chain reputation scores
- > Store review results (approved / rejected / needs-changes) on-chain for transparency
- > Record reviewer's wallet address and review timestamp for accountability
- > Update author's reputation score based on review outcomes
- > Create review token rewards for reviewers who provide quality feedback

### **Story 7: Earn digital badge for solving first complex problem**

#### **Potential On-Chain Requirements**

- > Create NFT collection for achievement badges with different rarity levels
- > Implement milestone tracking system that monitors user contributions on-chain
- > Mint unique NFT badge to user's wallet when criteria are met
- > Store achievement metadata (date earned, problem complexity level) in NFT
- > Prevent duplicate badge minting with on-chain verification logic

## **Story 8: See name on leaderboard of most helpful contributors**

### **Potential On-Chain Requirements**

- > Aggregate on-chain voting data and reputation scores in real-time
- > Store leaderboard rankings on-chain for decentralized verification
- > Update rankings automatically when new votes or contributions are recorded
- > Create time-based leaderboards (weekly, monthly, all-time) with separate on-chain accounts
- > Reward top contributors with bonus tokens through automated smart contract distribution

## **- Content Creators & Educators**

## **Story 9: Write tutorials using rich text editor with live preview**

### **Potential On-Chain Requirements**

- > Store tutorial content hash on IPFS with on-chain reference for permanence
- > Create tutorial account with creator's wallet address for ownership verification
- > Record tutorial metadata (creation date, last modified) on-chain
- > Initialize engagement metrics (views, likes, shares) to zero

## **Story 10: Upload videos for visual learning**

### **Potential On-Chain Requirements**

- > Store video file hash on IPFS with on-chain metadata record
- > Create video account linking content to creator's wallet address
- > Record video metadata (duration, file size, upload timestamp) on-chain
- > Initialize view counter and engagement metrics on-chain

## **Story 11: Create interactive code examples students can run and modify**

### **Potential On-Chain Requirements**

- > Store code execution environment configuration on-chain
- > Record student interactions and code modifications for analytics
- > Create sandbox account permissions linked to tutorial creator's wallet
- > Track code execution statistics and success rates on-chain for quality metrics

## **Story 12: Organize tutorials into learning courses with multiple lessons**

### **Potential On-Chain Requirements**

- > Create course account with structured lesson progression stored on-chain
- > Link individual tutorial accounts to course account through on-chain references
- > Store student enrollment data and progress tracking on-chain
- > Record course completion certificates as NFTs for student achievement verification

## **Story 13: Students tip with digital currency for quality content**

### **Potential On-Chain Requirements**

- > Create tipping smart contract that accepts SOL or SPL token transfers
- > Record tip amounts and sender wallet addresses for transparency
- > Transfer funds directly to educator's wallet with minimal fees
- > Store tipping history and total received amounts on educator's profile

## **Story 14: Create premium content for paying subscribers only**

### **Potential On-Chain Requirements**

- > Create subscription smart contract with time-based access control
- > Store subscriber wallet addresses and subscription expiry dates on-chain
- > Gate content access through on-chain verification of active subscription status

- > Handle automatic subscription renewals and payment processing through smart contracts

### **Story 15: Store content permanently on decentralized network**

#### **Potential On-Chain Requirements**

- > Upload all content to IPFS and store content hashes on Solana blockchain
- > Create content ownership records linking IPFS hashes to creator's wallet
- > Implement content versioning system with on-chain history tracking
- > Ensure content immutability through cryptographic hash verification

### **Story 16: Host live streaming events for real-time teaching**

#### **Potential On-Chain Requirements**

- > Create event account with scheduled time, duration, and participant limits
- > Store attendee wallet addresses for access control and attendance verification
- > Record event completion and attendance rates for educator reputation
- > Issue attendance NFTs or tokens as proof of participation

### **Story 17: Send direct messages to students for personalized guidance**

#### **Potential On-Chain Requirements**

- > Create encrypted messaging system with on-chain message metadata
- > Store message sender/receiver wallet addresses for identity verification
- > Record message timestamps and delivery status on-chain
- > Optional: Implement message tokens for premium messaging features

### **Story 18: See personalized feed based on interests**

### **Potential On-Chain Requirements**

- > Store user interest preferences and interaction history on-chain
- > Record post engagement data (views, likes, comments) for algorithm training
- > Create recommendation algorithm that uses on-chain behavioral data
- > Update feed preferences automatically based on on-chain activity patterns

### **Story 19: Save interesting posts to read later**

#### **Potential On-Chain Requirements**

- > Create bookmark account linking user wallet to saved post accounts
- > Store bookmark timestamp and categorization metadata on-chain
- > Allow bookmark sharing and collaboration through on-chain permissions
- > Track bookmark usage analytics for content quality metrics

### **Story 20: Filter discussions by topic and difficulty level**

#### **Potential On-Chain Requirements**

- > Store discussion metadata (topic, difficulty, tags) on-chain for decentralized filtering
- > Create content categorization system with community-verified tags
- > Record user filter preferences on-chain for personalized experience
- > Track popular filter combinations for platform optimization

### **Story 21: Comment on posts and reply to other comments**

#### **Potential On-Chain Requirements**

- > Create comment account linked to parent post account
- > Store commenter's wallet address and comment timestamp on-chain

- > Implement hierarchical reply system with on-chain thread tracking
- > Initialize comment voting system with zero likes/dislikes

### **Story 22: Send private messages to other community members**

#### **Potential On-Chain Requirements**

- > Create private messaging accounts between two wallet addresses
- > Implement end-to-end encryption with public keys stored on-chain
- > Record message delivery status and read receipts on-chain
- > Store message thread history with chronological ordering

### **Story 23: Join groups focused on specific topics**

#### **Potential On-Chain Requirements**

- > Create group account with member wallet addresses stored on-chain
- > Implement group permissions and moderation roles through smart contracts
- > Store group metadata (topic, rules, creation date) on-chain
- > Track group activity and engagement metrics for discovery algorithms

### **Story 24: Vote on community suggestions for new features**

#### **Potential On-Chain Requirements**

- > Create governance proposal accounts with voting metadata
- > Implement weighted voting based on token holdings and reputation
- > Store all votes immutably with voter wallet addresses and timestamps
- > Execute proposal outcomes automatically through smart contract logic

### **Story 25: Take quizzes about blockchain topics and earn points**

### **Potential On-Chain Requirements**

- > Store quiz questions and correct answers with cryptographic verification
- > Create quiz attempt accounts tracking user responses and scores
- > Distribute reward tokens automatically based on quiz performance
- > Prevent cheating through on-chain answer verification and timing controls

### **Story 26: Report inappropriate content to keep community safe**

#### **Potential On-Chain Requirements**

- > Create content reporting system with reporter wallet address verification
- > Store report details and content references on-chain for transparency
- > Implement community-based moderation with weighted voting on reports
- > Execute content moderation actions through automated smart contracts

### **Story 27: See welcome screen explaining platform purpose**

#### **Potential On-Chain Requirements**

- > Record user's first visit timestamp for onboarding analytics
- > Store onboarding completion status on-chain for progress tracking
- > Optional: Issue welcome NFT or tokens for completing platform introduction

### **Story 28: Step-by-step tutorial on creating digital wallet**

#### **Potential On-Chain Requirements**

- > Record tutorial completion milestones for user progress tracking
- > Verify successful wallet creation through on-chain transaction history

- > Issue completion certificate or badge as NFT for educational achievement

### **Story 29: Choose experience level for appropriate content**

#### **Potential On-Chain Requirements**

- > Store user experience level preference on-chain for decentralized personalization
- > Update experience level automatically based on on-chain activity and learning progress
- > Create content difficulty matching algorithm using on-chain user data

### **Story 30: Start with simplified interface showing basic features**

#### **Potential On-Chain Requirements**

- > Store UI preference settings on-chain for cross-device consistency
- > Track feature usage to automatically suggest interface upgrades
- > Record user progression from basic to advanced interface

### **Story 31: Receive welcome points for joining**

#### **Potential On-Chain Requirements**

- > Create welcome bonus smart contract that mints tokens for new users
- > Verify unique wallet address to prevent duplicate welcome bonuses
- > Record welcome bonus distribution timestamp and amount
- > Link welcome bonus to completion of onboarding requirements

### **Story 32: Upload UI/UX design images for community showcase**

#### **Potential On-Chain Requirements**

- > Store design file hashes on IPFS with on-chain ownership records



- > Create design portfolio account linked to designer's wallet address
- > Record upload timestamp and design metadata on-chain
- > Initialize design voting and feedback collection systems

### **Story 33: Receive comments and critiques on design work**

#### **Potential On-Chain Requirements**

- > Create design feedback accounts linked to original design posts
- > Store critic's wallet address and feedback timestamp for credibility
- > Implement design rating system with weighted votes based on expertise
- > Reward quality design feedback with reputation points or tokens

### **Story 34: Access library of design resources for blockchain apps**

#### **Potential On-Chain Requirements**

- > Store resource metadata and IPFS hashes on-chain for permanent access
- > Create resource contribution and curation system with community verification
- > Track resource usage and popularity through on-chain analytics
- > Reward resource contributors with tokens based on community usage

### **Story 35: Create project profile describing startup idea**

#### **Potential On-Chain Requirements**

- > Create project account with entrepreneur's wallet address as owner
- > Store project metadata (description, category, stage) on-chain
- > Initialize project metrics (views, interest, collaboration requests) to zero
- > Link project to entrepreneur's reputation and track record on-chain

### **Story 36: Search for team members based on skills and reputation**

#### **Potential On-Chain Requirements**

- > Store user skill profiles and expertise verification on-chain
- > Create team matching algorithm using on-chain reputation and activity data
- > Record collaboration requests and responses for transparency
- > Track successful team formations and project outcomes

### **Story 37: Post polls for business idea validation**

#### **Potential On-Chain Requirements**

- > Create poll account with voting options and metadata stored on-chain
- > Implement weighted voting based on voter expertise and reputation
- > Store all poll responses immutably with voter wallet addresses
- > Calculate and display poll results automatically through smart contract logic
- > Reward poll participation with community engagement points

# PART B

## Process Appendix

### PART A (Logs)

#### AI PROMPT: (part a - 2)

*My project's value proposition is [ A decentralized, Solana-native community forum purpose-built for Web3*

*Builders - developers, designers, and entrepreneurs, offering tokenized rewards (via custom tokens), gamified engagement (leaderboards, badges), and a low-cost, high-speed environment powered by Solana. Beyond just incentivizing participation, the platform aims to foster high-quality, sustained collaboration through a thoughtfully designed token economy and builder reputation system. By simplifying onboarding for non-technical users and focusing specifically on the Solana ecosystem's needs, it offers a focused, engaging alternative to both general-purpose Web2 forums and broad Web3 social platforms.*

*]. Here is a brainstormed list of all potential user types: [Direct Users*

*Solana developers*

*Web3 designers*

*Blockchain enthusiasts*

*Content creators and educators*

#### *Indirect Users / Beneficiaries*

*Solana foundation*

*Solana validators*

*Token holders and investors*

*Existing solana projects*

*Web3 service providers*

*Administrators/Moderators*

*Community managers*

*Technical moderators*

*Content curators*

*Event Coordinators*

*Stakeholders*

*Solana foundation*

*VC firms*

*Enterprise partners  
Educational institutions  
Web2 companies  
Content creators*

*J. Based on the value proposition, which 2-5 of these user types are the most critical to focus on for an initial Proof-of-Concept? For each user you recommend, provide a brief rationale explaining why they are essential for proving the project's core value.*

### **AI OUTPUT:**

*Based on my research findings on Solana development community needs, Web3 user acquisition strategies, and proof-of-concept user priorities, I can now provide recommendations for the most critical user types for your decentralized Solana-native community forum.*

#### *Critical User Types for Your Solana Community Forum Proof-of-Concept*

*Based on your value proposition as a decentralized, Solana-native forum with tokenized rewards and gamified engagement, here are the 4 most critical user types to focus on for your initial Proof-of-Concept:*

##### *1. Solana Developers*

*Rationale: Solana developers are the absolute cornerstone of your PoC success for several compelling reasons:*

- Active Pain Points: Research shows Solana developers face significant frustration with learning resources and community fragmentation. Your platform directly addresses these gaps by providing a focused, purpose-built environment.*
- High Engagement Potential: Developers naturally gravitate toward problem-solving discussions, code sharing, and technical collaboration - core activities that drive forum engagement.*
- Content Creation: They generate valuable technical content, tutorials, and solutions that attract other community members and establish platform credibility.*
- Token Economy Adoption: Developers understand blockchain mechanics and are more likely to engage with your tokenized reward system meaningfully rather than just speculatively.*
- Network Effects: Successful developer adoption creates a magnetic pull for other user segments who seek technical guidance and solutions.*

##### *2. Content Creators and Educators*

*Rationale: This group is essential for creating the educational ecosystem that differentiates your platform:*

- Value Alignment: Web3 content creators are actively seeking platforms that offer better monetization and ownership than Web2 alternatives. Your tokenized rewards directly address this need.*
- Community Building: They excel at creating engaging, educational content that helps onboard newcomers to the Solana ecosystem.*
- Early Adopter Mindset: Content creators in Web3 are inherently early adopters who embrace new platforms and help establish social proof.*

- Cross-Platform Promotion: They bring existing audiences from other platforms, accelerating user acquisition.
- Sustainable Content Flow: Unlike purely technical discussions, educational content appeals to broader audiences and maintains long-term engagement.

### 3. Community Managers

Rationale: Community managers are force multipliers who can make or break your PoC:

- Professional Expertise: They understand community dynamics, engagement strategies, and how to foster healthy discussions.
- Bridge Building: Community managers excel at connecting different user types and facilitating meaningful interactions between developers, creators, and enthusiasts.
- Platform Advocacy: They become natural evangelists who promote your platform within their professional networks.
- Feedback Quality: Their experience provides sophisticated insights into platform improvements and community needs.
- Scalability: They can help manage and moderate discussions as your community grows, essential for maintaining quality.

### 4. Blockchain Enthusiasts

Rationale: This group provides the critical mass and engagement foundation:

- High Activity Levels: Blockchain enthusiasts are typically power users who engage frequently and extensively with community platforms.
- Knowledge Sharing: They bridge the gap between developers and newcomers, helping to explain complex concepts in accessible ways.
- Token Economy Participation: They understand and actively engage with tokenized systems, helping validate your reward mechanisms.
- Community Loyalty: Once engaged, blockchain enthusiasts tend to become long-term, dedicated community members.
- Diverse Backgrounds: This group includes various professionals (traders, investors, researchers) who bring different perspectives and use cases.

Why These Four User Types Work Together

This combination creates a powerful ecosystem where:

- Developers provide technical depth and credibility
- Content creators make complex topics accessible and engaging
- Community managers facilitate connections and maintain platform health
- Blockchain enthusiasts provide sustained engagement and feedback

Together, they form a self-reinforcing community where each group's strengths complement the others' needs, creating the network effects essential for PoC success.

### **AI PROMPT (Part a -3)**

*For a project with this value proposition [A decentralized, Solana-native community forum purpose-built for Web3*

*Builders - developers, designers, and entrepreneurs, offering tokenized rewards (via custom tokens), gamified engagement (leaderboards, badges), and a low-cost, high-speed environment powered by Solana. Beyond just incentivizing participation, the platform aims to foster high-quality, sustained collaboration through a thoughtfully designed token economy and builder reputation system. By simplifying onboarding for non-technical users and focusing specifically on the Solana ecosystem's needs, it offers a*

*focused, engaging alternative to both general-purpose Web2 forums and broad Web3 social platforms.*

*] and focusing on these prioritized user types [Final prioritized list of User types*

*Solana developers*

*Decision* 

*WHY*

*They are the primary value drivers, they engage deeply, create technical content, test features, and provide critical feedback*

*A community without devs especially in web3 lacks depth. Their engagement legitimize the forum as a serious place for technical discussions*

*Content creators and educators*

*Decision* 

*WHY*

*Help translate technical content into digestible knowledge which widens the reach*

*Bring audience from X, YT, Newsletters and useful for acceleration of the platform*

*Blockchain enthusiasts*

*Decision* 

*WHY*

*They are the daily active users who read, react, comment and contribute from a non-technical lens*

*Their curiosity and dedication often lead to long term loyalty and organic content sharing*

*Community Managers*

*Decision* 

*WHY*

*While valuable, they are multipliers and scalers, not initial traction generators*

*In this, we need raw engagement and authenticity more than professionally managed communities*

*], help map out the key functions or interactions each user would need to perform.*

### **AI PROMPT (part a - 4- AI task)**

*Based on these top 1-2 critical user interactions [Solana Developers-discussion threads-validation and reward with peer upvoting,*

*Content Creators and Educators - Rich content posting - Markdown/text support*

*], what are the key technical requirements needed to build a proof-of-concept?*

## **PART B (Logs)**

### **AI PROMPT: (part b)**

Review my core user functions/stories [1. Solana Developers

Core Technical Functions

Code Sharing & Repository Integration

- Advanced code snippet editor with Solana/Rust syntax highlighting
- Integration with GitHub for seamless project sharing and version control
- Smart contract deployment testing environment
- Access to Solana development tools like Anchor framework documentation

Technical Discussion Forums

- Specialized channels for different Solana development areas (DeFi, NFTs, gaming, infrastructure)
- Advanced search functionality to find solutions to specific technical problems
  - Integration with Solana's official documentation and RPC endpoints

Development Environment Access

- Direct integration with Solana CLI tools and testnet/devnet access
- Sandboxed environment for testing smart contracts
- Performance optimization guides and compute usage analytics

Token Economy Interactions

Contribution-Based Rewards

- Higher token multipliers for solving complex technical challenges
- Reputation-based token earning rates that increase with proven expertise

Technical Validation System

- Peer review mechanisms where other developers validate technical contributions
  - Community voting on the quality and accuracy of technical content
- Automated token distribution based on code contribution metrics

Gamification Elements

Developer Achievement System

- Badges for different technical milestones (first deployed contract, bug fixes, optimizations)
- Leaderboards for most helpful technical contributors
- Special NFT rewards for significant contributions to the Solana ecosystem

## 2. Content Creators and Educators

### Content Creation Tools

#### Multi-Format Content Support

- Rich text editor for tutorials and educational posts
- Video upload and streaming capabilities for technical tutorials
- Interactive code examples with live execution
- Template systems for consistent educational content formatting

#### Educational Content Organization

- Curriculum builders for structured learning paths
- Progress tracking for students following educational content
- Content categorization by difficulty level and topic area

#### Decentralized Content Ownership

- IPFS integration for permanent, censorship-resistant content storage
- NFT-based content ownership verification
- Smart contract-based copyright protection with timestamping

### Monetization Functions

#### Direct Fan Support

- Cryptocurrency tipping systems for quality educational content
- Subscription models using native forum tokens
- Crowdfunding mechanisms for large educational projects

#### Token-Gated Premium Content

- Exclusive content access for token holders
- Tiered access levels based on token ownership or community participation
- Revenue sharing from premium content subscriptions

### Community Building Tools

#### Audience Engagement

- Live streaming capabilities for educational events
- Community polls and feedback systems
- Direct messaging with students and followers
- Event hosting and workshop organization tools

## 3. Blockchain Enthusiasts

### Discovery and Engagement Functions

#### Content Consumption Interface

- Personalized content feeds based on interests and interaction history
- Advanced filtering and search across all content types
- Bookmarking system for saving interesting discussions and resources

#### Social Interaction Tools

- Commenting and discussion threading on all content types
- User-to-user messaging and connection building
- Group formation around specific interests or projects

#### News and Trend Tracking

- Real-time updates on Solana ecosystem developments
- Curated news feeds from trusted sources
- Price and market data integration for relevant tokens



## Community Participation

### Voting and Governance

- Community polls on forum features and improvements
- Weighted voting based on reputation and token holdings
- Proposal submission for community initiatives

### Curation and Moderation

- Content quality rating systems
- Community-driven content moderation through reporting
- Crowdsourced fact-checking for news and claims

### Token Economy Participation

#### Engagement-Based Rewards

- Daily login bonuses and streak rewards
- Tokens for quality comments, shares, and community interactions
- Referral bonuses for bringing new quality members

#### Gamified Learning

- Quiz systems about Solana and blockchain technology with token rewards
- Achievement badges for different levels of platform engagement
- Seasonal challenges and competitions with NFT prizes.

] and requirements [Technical Requirements for Your Solana Community Forum

### Proof-of-Concept

Based on the two critical user interactions — (1) discussion threads with peer validation/rewards for Solana developers, and (2) rich content posting with Markdown support for educators — here are the essential technical requirements for your proof-of-concept.

#### Core Infrastructure Requirements

##### Blockchain Integration Layer

##### Solana Network Connection

RPC endpoint integration with Solana mainnet/devnet for real-time blockchain interactions

Wallet adapter supporting major Solana wallets (Phantom, Solflare, Backpack)

Smart contract deployment capability for custom token rewards and governance

Transaction signing and verification mechanisms for user authentication

#### Identity Management

Solana wallet-based authentication system eliminating traditional username/password

Public key-based user identification and reputation tracking

Decentralized identity verification using on-chain transaction history

Database Architecture

Hybrid Storage Solution

Traditional database (PostgreSQL/MongoDB) for forum metadata, user preferences, and real-time interactions

IPFS integration for decentralized content storage ensuring censorship resistance

Smart contract state management for token balances, voting records, and reputation scores

User Interaction #1: Discussion Threads with Peer Validation

Forum Infrastructure

Thread Management System

Hierarchical comment threading with unlimited depth

Real-time updates using WebSocket connections

Full-text search across all discussions

Content categorization by Solana dev topics (DeFi, NFTs, infrastructure, etc.)

Peer Upvoting Mechanism

Voting Smart Contract

Immutable on-chain upvote/downvote recording

Weighted voting based on user reputation and community stake

Anti-spam mechanisms (e.g., time-locks, stake requirements)

Transparent, publicly verifiable vote tallying

Validation Logic

Peer review of technical content

Quality scoring via votes, engagement, and accuracy

Auto-flagging low-quality/spam posts

Token Reward Distribution

Automated Reward System

Smart contract-based token distribution for validated posts

Dynamic rewards based on quality, complexity, and engagement

Escrow and delay mechanisms for fair distribution

Integration with SPL for custom token minting and management

User Interaction #2: Rich Content Posting with Markdown Support

Content Creation Engine

Markdown Processing Infrastructure

Real-time GFM/CommonMark-compliant Markdown parser

WYSIWYG editor with live preview

Syntax highlighting for Rust, JavaScript, and Solana code

LaTeX rendering for blockchain/math equations

Rich Media Support

Image upload with CDN distribution

Embedded video tutorials

Interactive code runner for smart contract testing

File attachment system for dev resources

Content Storage and Versioning  
Decentralized Content Management  
IPFS-based storage for permanent archival

Version history and content diffs

Content hash verification

Redundant backup across IPFS nodes

Technical Implementation Stack  
Frontend Requirements  
User Interface Framework  
React.js or Next.js frontend

Tailwind CSS for consistent mobile-first design

WebSocket integration for real-time updates

PWA support for mobile use

Backend Architecture  
API and Server Infrastructure  
Node.js/Express.js or Rust for performant APIs

REST or GraphQL APIs for flexible data handling

Redis caching for performance

Rate limiting and DDoS protection

Security and Performance

Security Measures

End-to-end encryption

Input sanitization / XSS protection

Multi-signature wallet access for admins

Security audits and penetration testing

Scalability Considerations

Horizontal scaling support

CDN for global content access

Sharding for large dataset management

Aggressive caching for frequently accessed content

Minimum Viable Technical Setup

Phase 1 – Essential Features

Thread creation and commenting

Solana wallet login and user profiles

Basic on-chain upvoting

Markdown editor with live preview

Token rewards for validated posts

Phase 2 – Enhanced Functionality  
Advanced search + categorization

Reputation + weighted voting

IPFS integration

Media uploads and file sharing

Real-time notifications and feeds

]. Considering my project's refined value proposition [A decentralized, Solana-native community forum purpose-built for Web3

Builders - developers, designers, and entrepreneurs, offering tokenized rewards (via custom tokens), gamified engagement (leaderboards, badges), and a low-cost, high-speed environment powered by Solana. Beyond just incentivizing participation, the platform aims to foster high-quality, sustained collaboration through a thoughtfully designed token economy and builder reputation system. By simplifying onboarding for non-technical users and focusing specifically on the Solana ecosystem's needs, it offers a focused, engaging alternative to both general-purpose Web2 forums and broad Web3 social platforms.

], do these stories truly hit the mark? Are the requirements granular enough to map to specific technical components (e.g., database schemas, API endpoints, specific blockchain programs)? What's missing or unclear?

## **PART C ( Refinement Log )**

### **De-Jargoning Changes**

Before: "Advanced code snippet editor with Solana/Rust syntax highlighting"

After: "Write code in a text box that automatically colors different parts of my code"

Rationale: Removed technical terms like "syntax highlighting" and "code snippet editor"

Before: "Integration with GitHub for seamless project sharing"

After: "Click a button to connect my GitHub account so I can share existing projects"

Rationale: Removed "integration" jargon and made the action concrete

Before: "RPC endpoint integration with Solana mainnet/devnet"

After: Removed from user stories - this is technical implementation detail

Rationale: Users don't care about RPC endpoints, they care about the outcome

### **Atomicity Improvements**

Before: "User signs up and creates a profile with wallet integration"

After:

- Story 27: "See a welcome screen that explains what this platform does"
- Story 28: "Step-by-step tutorial on creating a digital wallet"
- Story 29: "Choose my experience level"

Rationale: Split complex onboarding into discrete, testable actions

Before: "Peer review mechanisms where developers validate technical contributions and vote"

After:

- Story 6: "Other experienced developers review my code contributions"
- Story 5: "Receive community points when other developers vote my answer as helpful"

Rationale: Separated review process from voting mechanism

### **Clarity Improvements**

Before: "Tokenized rewards via custom tokens"

After: "Receive community points when other developers vote my answer as helpful"

Rationale: Made the reward mechanism and trigger condition explicit

Before: "Community-driven content moderation through reporting"

After: "Report inappropriate content so I can help keep the community safe"

Rationale: Made the user's motivation and outcome clear

### **Redundancy Elimination**

Before: Had separate stories for "upvoting" and "community voting on quality"

After: Combined into Story 5 about receiving points for helpful answers

Rationale: Both described the same core voting mechanism

Before: Multiple stories about "advanced search functionality"

After: Story 4: "Search for solutions using keywords"

Rationale: Consolidated various search features into one clear user need

### **Missing User Types Added**

Added: Stories 32-34 for Designers

Rationale: Value proposition mentions "designers" but original stories were developer-heavy

Added: Stories 35-37 for Entrepreneurs

Rationale: Value proposition targets "entrepreneurs" but original stories lacked business-focused features

Added: Stories 27-31 for New Users

Rationale: Value proposition promises "simplified onboarding for non-technical users" but original stories assumed technical knowledge