BitmergeXProduct Requirements Document (PRD)

Product Name: BitMergeX **(Simple Bitcoin Exchange on Clarity)**

1. Objective

The goal is to build a decentralized exchange (DEX) for trading Bitcoin and Bitcoin-pegged assets using Clarity smart contracts on the Stacks blockchain. The exchange will offer a simple, user-friendly platform for users to trade directly without the need for a centralized intermediary, leveraging Bitcoin for settlement and security.

This exchange will allow users to trade Bitcoin and other digital assets, ensuring low fees, high security, and transparent, decentralized order execution.

---

2. Key Features

- **Decentralized Trading Platform:** A marketplace for trading Bitcoin and other Stacks/Bitcoin-pegged assets using Clarity smart contracts.
- **User Wallets:** Users can connect their Stacks wallets to interact with the exchange.
- **Bitcoin Trading Pairs:** Ability to trade Bitcoin against various Stacks-based tokens (e.g., wrapped Bitcoin or stablecoins like USDA).
- **Automated Market Making (AMM):** Optionally integrate a basic AMM mechanism to allow users to swap between assets.
- **No Central Custody:** Users retain full custody of their assets using self-managed wallets.
- **Clarity Smart Contracts:** All trades, transactions, and liquidity management will be governed by auditable Clarity smart contracts.
- **Security Leveraged from Bitcoin:** Settlement and transaction finality are anchored in Bitcoin for enhanced security.

---

3. User Stories

3.1 User Onboarding

- **As a user**, I want to sign in using my decentralized Stacks wallet (e.g., Hiro Wallet) so that I can securely access my funds and interact with the platform.
- **As a new user**, I want guidance on how to set up a wallet and how to trade on the exchange.

3.2 Asset Trading

- **As a user**, I want to trade Bitcoin for a Stacks-based stablecoin (e.g., USDA) or another Stacks token so that I can access more liquidity.
- **As a user**, I want to view current market prices, order books, and my trade history so that I can make informed decisions.

3.3 Smart Contract Interaction

- **As a user**, I want to ensure that all my trades are executed through transparent, auditable smart contracts so that I can verify the fairness and accuracy of trades.

3.4 Wallet Management

- **As a user**, I want to manage my assets directly from my wallet so that I can withdraw or deposit funds without relying on a third party.

3.5 Liquidity Pooling (Optional AMM)

- **As a user**, I want to participate in a liquidity pool to earn rewards by providing liquidity for others to trade between Bitcoin and Stacks-based assets.

3.6 Security and Auditability

- **As a user**, I want to verify that the platform is secure and that no one can manipulate the market or my funds so that I can trust the exchange.

---

4. Technical Requirements

4.1 Backend

- **Clarity Smart Contracts:**
  - Develop core smart contracts for trading, order matching, and liquidity management.
  - Contracts should be Turing-incomplete to ensure predictability and security.
  - Integrate Clarity's `postConditions` feature to verify transactions without any surprises.
- **Bitcoin Integration:**
  - Use the Proof of Transfer (PoX) mechanism to ensure that all transactions and tokenized assets are anchored to Bitcoin's security.
  - Support for Bitcoin transfers between Stacks and Bitcoin using native BTC-pegged assets like Stacks' wrapped BTC (xBTC).

4.2 Frontend

- **User Interface (UI):**
    - Build a simple, responsive web-based UI using React.
    - Integrate with the Stacks Wallet (e.g., Hiro Wallet) for decentralized login and transaction signing.
    - Display current prices, charts, and order books for supported assets.
    - Allow users to initiate and sign trades using their wallets.
- **Order Books and Trade Matching:**
    - Display buy/sell order books.
    - Allow users to place limit orders, market orders, and track open orders.

4.3 Database

- **Market Data Storage:**
    - Store and cache market data such as trading history, open orders, and price movements using MongoDB.
    - Ensure data integrity by anchoring transaction logs on the Stacks blockchain.

4.4 Wallet Integration

- **Stacks Wallet Integration:**
    - Use Hiro Wallet (or any Stacks wallet supporting Clarity contracts) for user authentication and transaction signing.
    - Users should be able to connect wallets, view balances, and interact with Clarity contracts directly.

4.5 API Integration

- **Stacks Blockchain APIs:**
    - Use Stacks blockchain APIs to fetch transaction history, smart contract interactions, and network status.
- **Price Feeds:**
    - Integrate external price feeds for BTC and other assets to provide real-time price information.

4.6 Security

- **Smart Contract Audits:**
    - Perform external audits on all Clarity smart contracts to ensure there are no vulnerabilities.
- **Cold Wallet and Multisig:**
    - Encourage users to interact with multisignature wallets for high-value trades or asset custody.
- **PoX Integration:**
    - Use Stacks' PoX (Proof of Transfer) consensus mechanism to anchor transactions and ensure tamper-proof finality on Bitcoin.

---

5. User Interface Design

- **Simple Dashboard:** Show portfolio overview, asset balances, trading pairs, and market data.
- **Order Placement Screen:** Users can place market or limit orders directly from their wallet.
- **History Page:** Users can view their trade and transaction history with filters for specific time periods or assets.

---

6. Success Metrics

- **User Adoption:**
  - Target of 10,000 users in the first 6 months.
  - Daily active users (DAU) as a key metric for engagement.
- **Transaction Volume:**
  - Achieve $5 million worth of daily trading volume within the first 6 months.
- **Security and Downtime:**
  - Ensure 99.9% uptime with no major security incidents.
  - Regular smart contract audits to maintain platform integrity.
- **Liquidity:**
  - Maintain at least $1 million in liquidity in major trading pairs within the first 3 months.

---

7. Risks and Mitigation

- **Smart Contract Vulnerabilities:**
  - Regular external audits of all Clarity smart contracts to prevent exploits.
- **Low Liquidity:**
  - Bootstrap liquidity by incentivizing early users with rewards or through liquidity mining programs.
- **Regulatory Risks:**
  - Stay compliant with local regulations (AML/KYC) by integrating optional compliance mechanisms or restricting certain jurisdictions.

---

8. Timeline

Phase 1 (0-2 months)

- Finalize requirements, design, and architecture.
- Build Clarity smart contracts for trading functionality.
- Set up frontend and backend infrastructure.

Phase 2 (2-4 months)

- Integrate Stacks wallet functionality.
- Build order book and trading interface.
- Conduct internal testing and user feedback sessions.

Phase 3 (4-6 months)

- Launch Beta platform with limited trading pairs.
- Conduct security audits.
- Onboard early users and liquidity providers.

Phase 4 (6+ months)

- Full public release with all trading pairs.
- Continue improving based on user feedback and market needs.

---

9. Appendices

9.1 Glossary

- **Clarity:** A smart contract language used on the Stacks blockchain, which anchors to Bitcoin for security.
- **PoX (Proof of Transfer):** A consensus mechanism used by the Stacks blockchain to leverage Bitcoin's security.
- **Stacks Wallet:** A decentralized wallet used to interact with the Stacks blockchain.

9.2 References

- Stacks Documentation: https://docs.stacks.co/
- Clarity Language: https://clarity-lang.org/