DeFi Protocol Project Guide

Build Decentralized Finance Primitives

BSc Blockchain, Crypto Economy & NFTs

FS2026

Learning Objectives

By completing this project, you will:

1. Understand constant product AMM mechanics
2. Implement collateralized lending
3. Create flash loan functionality
4. Integrate price oracles

Related Lessons: L33, L34, L35, L37, L38, L42

Problem: How do we create financial services without trusted intermediaries?

DeFi Building Blocks

Automated Market Makers

- Liquidity pools instead of order books
- Constant product formula: $x \cdot y = k$
- Anyone can provide liquidity
- Fees go to liquidity providers

Lending Protocols

- Overcollateralized loans
- Interest rates by supply/demand
- Liquidation mechanisms
- No credit checks

Key Innovation: Composability - DeFi protocols can be combined like LEGO blocks

DeFi replaces banks with smart contracts

Constant Product AMM

The Formula: $x \cdot y = k$

- x = Reserve of Token A
- y = Reserve of Token B
- k = Constant (invariant)

Price Calculation:

$$\text{Price of A} = \frac{y}{x}$$

Swap Output (after fee):

$$\Delta y = \frac{y \cdot \Delta x \cdot (1 - \text{fee})}{x + \Delta x \cdot (1 - \text{fee})}$$

Properties:

- Price increases as you buy (slippage)
- Infinite liquidity (but infinite slippage at extremes)
- Fees accumulate in k

Uniswap V2 uses this exact formula

AMM Price Curve

Lending Protocol Mechanics

Collateralized Lending Flow:

1. Deposit collateral (e.g., ETH)
2. Borrow up to X% of collateral value
3. Pay interest over time
4. Repay debt + interest to withdraw collateral

Key Parameters:

Parameter	Typical	Purpose
Collateral Factor	75%	Max borrow per collateral
Liquidation Threshold	80%	When liquidation triggers
Liquidation Bonus	5%	Incentive for liquidators

Health Factor:

$$\text{HF} = \frac{\text{Collateral Value} \times \text{Liq. Threshold}}{\text{Debt Value}}$$

Liquidation occurs when $\text{HF} < 1$

Overcollateralization eliminates credit risk

Flash Loans

What Makes Them Special:

- Borrow any amount with zero collateral
- Must repay + fee in same transaction
- If not repaid, entire transaction reverts
- Enables atomic arbitrage

Use Cases:

Legitimate

- DEX arbitrage
- Collateral swaps
- Self-liquidation

Attack Vectors

- Oracle manipulation
- Governance attacks
- Price exploitation

Fee: Typically 0.09% (9 basis points)

Flash loans democratize access to large capital

Oracle Integration

Why Oracles?

- Smart contracts can't access external data
- Prices, weather, sports - all need oracles
- Security depends on oracle reliability

Chainlink Price Feeds:

```
(, int256 price,,,) = priceFeed.latestRoundData();
// ETH/USD price with 8 decimals
```

Oracle Risks:

- Stale prices (check timestamp)
- Flash loan manipulation (use TWAPs)
- Single point of failure (use multiple sources)

Oracle security is critical for DeFi protocols

Risk Analysis

DeFi Risk Categories:

Risk	Description	Mitigation
Smart Contract	Bugs, exploits, reentrancy	Audits, bug bounties
Oracle	Price manipulation	TWAPs, multiple sources
Liquidation	Cascade liquidations	Conservative parameters
Governance	Malicious proposals	Time locks, quorum
Economic	Token depegs, bank runs	Insurance, reserves

Historical Examples:

- The DAO (2016): Reentrancy exploit
- bZx (2020): Flash loan attack
- Terra/Luna (2022): Economic failure

Understanding risks is essential for building secure protocols

Apply the 6 Questions:

Question	DeFi Protocol Answer
PROBLEM	Financial services without banks
INCENTIVES	LPs earn fees, liquidators earn bonus
BENEFITS/COSTS	Permissionless access; gas costs, risks
FAILURE MODE	Oracle manipulation, flash loan attacks
DESIGN CHOICES	Collateral ratios, fee structures
ALTERNATIVES	Centralized exchanges, traditional finance

DeFi trades counterparty risk for smart contract risk

Implementation Checklist

Phase 1: Simple AMM

- ☐ Deploy two test tokens
- ☐ Create liquidity pool
- ☐ Implement swap function
- ☐ Test slippage calculations

Phase 2: Lending Pool

- ☐ Implement deposit/withdraw
- ☐ Add borrow/repay functions
- ☐ Create liquidation mechanism

Phase 3: Flash Loans

- ☐ Add flash loan function
- ☐ Create receiver interface
- ☐ Test arbitrage scenario

Phase 4: Oracle Integration

- ☐ Connect Chainlink price feeds
- ☐ Update lending with real prices
- ☐ Add staleness checks

Resources

Protocols to Study:

- Uniswap V2: `docs.uniswap.org`
- Aave V3: `docs.aave.com`
- Compound: `compound.finance/docs`

Tools:

- Chainlink: `docs.chain.link`
- DefiLlama: `defillama.com`
- Dune Analytics: `dune.com`

Project Materials:

- Notebook: `projects/notebooks/03_defi_protocol.ipynb`
- Web: `.../projects/defi-protocol/`