





SpiralOS Integration - Complete Implementation


This pull request introduces the complete SpiralOS integration with ScarCoin contract functionality, witness system, and enhanced UI components.

Features Implemented


Core Infrastructure

-  **Complete Next.js Application Structure** with TypeScript support
-  **Polygon Amoy Testnet Configuration** (Chain ID: 80002)
-  **Production-ready build setup** with Tailwind CSS, ESLint, and PostCSS
-  **Environment configuration** with example file for easy deployment








Smart Contract Integration

-  **ScarCoin ERC-20 Contract Integration**
- Full token functionality (transfer, balance, allowance)
- Contract ABI and deployment configuration
- Type-safe contract interactions with ethers.js




API Endpoints (/api/witness)

-  **Witness System Endpoints**
- `POST /api/witness` - Create witness entries with signature verification
- `GET /api/witness/nonce` - Get nonce for signature generation
- `GET /api/witness/stats` - Retrieve witness statistics
- Full signature verification and security measures



Enhanced UI Components

-  **Dashboard Component** - Main application interface
-  **WalletConnect Component** - MetaMask integration and connection
-  **TokenInfo Component** - Real-time ScarCoin balance and info display
-  **TransferForm Component** - Token transfer functionality
-  **LogAche Component** - Transaction and activity logging
-  **WitnessEntries Component** - Display witness data
-  **WitnessStats Component** - Statistics and analytics

Database Integration

-  **Supabase Integration** with service role key
-  **Database setup endpoint** (`/api/setup-db`) for witness table creation
-  **Type-safe database operations** with proper error handling

Custom Hooks

-  **useScarCoin Hook** - ScarCoin contract interactions
-  **useWallet Hook** - Wallet connection and management

-  **useWitness Hook** - Witness system operations

Technical Stack






- **Frontend:** Next.js 14, React 18, TypeScript
- **Styling:** Tailwind CSS with responsive design
- **Blockchain:** ethers.js v6, MetaMask integration
- **Database:** Supabase with PostgreSQL
- **Network:** Polygon Amoy Testnet
- **Build Tools:** ESLint, PostCSS, Vercel deployment ready

Configuration





Environment Variables Required

```
NEXT_PUBLIC_CHAIN_ID=80002
NEXT_PUBLIC_RPC_URL=https://rpc-amoy.polygon.technology
NEXT_PUBLIC_NETWORK_NAME=Amoy
NEXT_PUBLIC_CURRENCY_SYMBOL=MATIC
NEXT_PUBLIC_BLOCK_EXPLORER=https://amoy.polygonscan.com
NEXT_PUBLIC_SCARCOIN_CONTRACT=<deployed_contract_address>
NEXT_PUBLIC_SUPABASE_URL=https://dnhagjbsruihntzbckm.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=<anon_key>
SUPABASE_SERVICE_ROLE_KEY=<service_role_key>
```

Testing & Quality

-  **TypeScript strict mode** enabled
-  **ESLint configuration** for code quality
-  **Error handling** throughout the application
-  **Responsive design** for mobile and desktop
-  **Production-ready** build configuration

Deployment

-  **Vercel configuration** included
-  **CI/CD pipeline** ready
-  **Environment setup** documented
-  **Database initialization** endpoint available

Database Setup

After deployment, initialize the database by calling:

```
curl -X POST https://your-domain.com/api/setup-db \
-H "Authorization: Bearer $SUPABASE_SERVICE_ROLE_KEY"
```

Files Changed/Added

Core Application

- `pages/index.tsx` - Main dashboard page
- `pages/_app.tsx` - App configuration with providers
- `pages/_document.tsx` - Document structure

API Routes

- `pages/api/witness/index.ts` - Main witness endpoint
- `pages/api/witness/nonce.ts` - Nonce generation
- `pages/api/witness/stats.ts` - Statistics endpoint
- `pages/api/setup-db.ts` - Database initialization

Components

- `components/Dashboard.tsx` - Main dashboard
- `components/WalletConnect.tsx` - Wallet connection
- `components/TokenInfo.tsx` - Token information display
- `components/TransferForm.tsx` - Token transfer form
- `components/LogAche.tsx` - Activity logging
- `components/WitnessEntries.tsx` - Witness data display
- `components/WitnessStats.tsx` - Statistics display

Hooks & Utilities

- `hooks/useScarCoin.ts` - ScarCoin contract hook
- `hooks/useWallet.ts` - Wallet management hook
- `hooks/useWitness.ts` - Witness system hook
- `lib/ethers.ts` - Ethers.js configuration
- `lib/supabase.ts` - Supabase client setup

Configuration

- `contracts/ScarCoin.json` - Contract ABI
- `types/ethereum.d.ts` - TypeScript definitions
- Configuration files (`next.config.js`, `tailwind.config.js`, etc.)

Ready for Production

This implementation is production-ready with:

- Comprehensive error handling
- Type safety throughout
- Responsive UI design
- Secure API endpoints
- Proper environment configuration
- Database integration
- Smart contract integration

The application provides a complete ScarCoin wallet interface with witness system functionality, ready for deployment on Polygon Amoy testnet.