**BASE STRUCTURE**

**Project Name:** Maya Exchange

**Core Features:**

1. KYC/AML Verification (Know-Your-Coustomer / Anti Money Laundering)
2. Integrated Banking Services (IBS)
3. UPI Payments (Unified Payment Service)
4. Cross-Border Transactions (CBT)
5. Local Currency to Local Currency (LC2LC)
6. Crypto to Crypto (C2C)
7. Crypto to Local Currency (CLC)
8. Local Currency To Crypto (LC2C)
9. Multi-Currency Support (MCS)
10. Local Internal Currency Transactions (LICT)
11. Inbuild UPI system like phonepe and gpay and cross app transactions.

**User Application Design For:**

1. Web Application (Responsive Design)
2. Mobile Application (iOS and Android)
3. API for Integrations (Application Programming Interface)

**User Flow:**

**Scenario 1: User 1 sends local currency to User 2 (LC2LC)**

1. User 1 logs in and selects "Send" option.
2. Chooses recipient (User 2).
3. Enters amount and confirms transaction.
4. Maya Exchange Transfers Local currency to user 2 Via UPI Infrastructure.

**Scenario 2: User 1 sends crypto to User 2 (C2C)**

1. User 1 logs in and selects "Send" option.
2. Chooses recipient. (User 2).
3. Selects cryptocurrency. (BTC).
4. Chooses desired cryptocurrency for User 2 (ETH)
5. Enters amount and confirms transaction.
6. Maya Exchange executes crypto-to-crypto transaction.

**Scenario 3: User 1 Sends crypto and wants local currency (C2LC)**

1. User 1 logs in and selects "Send" option.
2. Chooses recipient (User 2).
3. Selects cryptocurrency (BTC).
4. Chooses local currency (INR).
5. Maya Exchange converts crypto to local currency it’ll transfer and credits to User 2 Bank Account that linked to the app and selected as primary.

**Scenario 4: User 1 Sends Local Currency and wants Crypto(C2LC)**

1. User 1 logs in and selects "Send" option.
2. Chooses recipient (User 2).
3. Selects cryptocurrency (INR).
4. Chooses local currency (BTC).
5. Maya Exchange converts local currency to crypto it’ll transfer and credits to User 2 Web3, Crypto Hot wallet in the app.

**Tech Stack:**

1. **Frontend:** HTML, CSS, React.tsx, tsx, java, kotlin
2. **Backend & Security and encryption:** C#, Perl, Ruby, Python
3. **Encryption algorithms:** CRYSTALS-Kyber, FALCON, Bcrypt
4. **Hashing Algorithms:** SHA-256, Skein, Grøstl, Whirlpool, Streebog
5. **Database:** MongoDB, MySQL, PostgreSQL, Amazon RDS, Amazon DynamoDB, Amazon Redshift, Apache Cassandra
6. **Integration:** Amazon DMS, AWS RDBS, AWS API Gateway, AWS Lambda
7. **Cryptocurrency Integration:** APIs (cryptoapi.io, tradinview), Web3.js, own liquidity pool
8. **Banking Integration APIs & SDKs:** debit, credit, rupay, visa, master card, amex, UPI SDKs
9. **Security**: OpenSSL, SSL/TLS, 2FA, JWS, Oauth Token, Encryption, access controls
10. **AWS:** Amazon EC2, Amazon RDS, Amazon S3, Amazon CloudWatch
11. **Orchestration & Containerization & Deployment:** Podman& Kubernetes
12. **(CI/CD):** GitLab CI/CD & Jenkins

**API Endpoints:**

1. **/users** - User management
2. **/transactions** - Transaction history
3. **/send** - Send local currency or crypto
4. **/receive** - Receive crypto or local currency
5. **/convert** - Convert currency (LCC, CLC, CC)
6. **/kyc** - KYC/AML verification

**Database Schema:**

1. Users table: id, name, email, password, KYC status
2. Transactions table: id, user\_id, transaction\_type, amount, currency
3. Currencies table: id, name, symbol, exchange\_rate

**Security Considerations:**

1. Encryption for sensitive data
2. Secure password storage (bcrypt)
3. Two-Factor Authentication (Google Authenticator)
4. Regular security audits and penetration testing

**Development Roadmap:**

**Phase 1: Research and Planning (2 weeks)**

1. Market research
2. Technical feasibility study
3. Team assembly

**Phase 2: Frontend Development (8 weeks)**

1. User interface design
2. Web and mobile application development

**Phase 3: Backend Development (12 weeks)**

1. API development
2. Database design and integration
3. Cryptocurrency and banking integrations

**Phase 4: Testing and Launch (8 weeks)**

1. Unit testing
2. Integration testing
3. Security auditing
4. Launch preparation

**Phase 5: Maintenance and Updates (Ongoing)**

1. Regular security updates
2. Feature enhancements
3. User support

**Team Structure:**

1. Project Manager
2. Frontend Developer
3. Backend Developer
4. DevOps Engineer
5. QA Engineer
6. Security Specialist

**Estimated Development Time:** 30 weeks (approximately 7.5 months)

**Estimated Development Cost:** $250,000 - $500,000 (depending on team size and location)

Please let me know if you'd like to proceed with the development of Maya Exchange or require further clarification on any aspect.

Would you like to:

1. Proceed with development
2. Request clarification on specific features
3. Modify the scope or timeline
4. Discuss budget and pricing

**Tech Architecture, Components & Features**

**Project Name :** Maya Exchange

**Tech Stack:**

1. **Frontend:** HTML, CSS, React.tsx, tsx, java, kotlin
2. **Backend & Security and encryption:** C#, Perl, Ruby, Python
3. **Encryption algorithms:** CRYSTALS-Kyber, FALCON, Bcrypt
4. **Hashing Algorithms:** SHA-256, Skein, Grøstl, Whirlpool, Streebog
5. **Database:** MongoDB, MySQL, PostgreSQL, Amazon RDS, Amazon DynamoDB, Amazon Redshift, Apache Cassandra
6. **Integration:** Amazon DMS, AWS RDBS, AWS API Gateway, AWS Lambda
7. **Cryptocurrency Integration:** APIs (cryptoapi.io, tradinview), Web3.js, own liquidity pool
8. **Banking Integration APIs & SDKs:** debit, credit, rupay, visa, master card, amex, UPI SDKs
9. **Security**: OpenSSL, SSL/TLS, 2FA, JWS, Oauth Token, Encryption, access controls
10. **AWS:** Amazon EC2, Amazon RDS, Amazon S3, Amazon CloudWatch
11. **Orchestration & Containerization & Deployment:** Podman& Kubernetes
12. **(CI/CD):** GitLab CI/CD & Jenkins

**Architecture:**

1. Frontend: User interface (Web, Mobile)
2. Backend: API, Services (Microservices architecture)
3. Database: Relational databases (e.g., MySQL) for storing user data, transactions, and order books
4. Cryptocurrency Integration: APIs for cryptocurrency exchanges, wallets, and payment processors
5. Banking Integration: APIs for banking services, UPI, and payment gateways
6. Security: Encryption, 2FA, access controls

**Components:**

1. **User Management :**
   1. KYC/AML checks
   2. User profiles
   3. Account management
2. **Crypto Management :**
   1. Wallet Integration
   2. Exchange Integration
   3. Trading engine
3. **Banking Management :**
   1. Upi & IMPS & Individual Bank interface integration
   2. Payment gateway integration
   3. Fiat currency management
4. **Transaction Management :**
   1. Order books
   2. Trade matching
   3. Settlement
5. **Cross-Border Transactions Management :**
   1. International payment processing
   2. Currency conversion
   3. Compliance with regulations

**Local Currency to Crypto Feature (Same Scenario With Local Currency to crypto) :**

**1. User Interface :**

* 1. Select recipient (User 2)
  2. Choose local currency (INR)
  3. Select desired cryptocurrency or local currency

**2. Backend Processing :**

* 1. Verify user balance and KYC/AML status
  2. Convert local currency to cryptocurrency (or other local currency)
  3. Execute trade on cryptocurrency exchange or OTC desk
  4. Update user balances and transaction history

**Cloud Services Integration :**

**AWS:**

* 1. Amazon EC2 (scalable infrastructure)
  2. Amazon RDS (database management)
  3. Amazons3 (storage)
  4. Amazon CloudWatch (monitoring)

**Advanced Features**

1. **Artificial Intelligence (AI)**: Implement AI-powered trading bots and market analysis tools
2. **Machine Learning (ML)**: Integrate ML algorithms for predictive modeling and risk management
3. **Blockchain Analytics**: Utilize blockchain analytics tools for transaction tracking and security
4. **Compliance**: Implement automated compliance checks for regulatory requirements

**Deployment Strategy**

1. **Containerization**: Use Podman for containerization
2. **Orchestration**: Utilize Kubernetees & Podman for container orchestration
3. **Continuous Integration/Continuous Deployment (CI/CD)**: Implement CI/CD pipelines using GitLab CI/CD

**Security Measures**

1. **Encryption**: Implement end-to-end encryption for data protection
2. **Access Controls**: Enforce strict access controls and role-based permissions
3. **Two-Factor Authentication**: Require 2FA for all user accounts
4. **Regular Security Audits**: Conduct regular security audits and penetration testing

**Scalability**

1. **Horizontal Scaling**: Scale infrastructure horizontally using cloud services
2. **Load Balancing**: Implement load balancing for efficient traffic distribution
3. **Caching**: Utilize caching mechanisms for improved performance

**Database Combination**

**Traditional Databases**

1. **MongoDB**: Store customer profiles, order history, and market data.
2. **MySQL**: Manage transactional workloads, user accounts, and banking information.
3. **PostgreSQL**: Store complex relational data, trade history, and order books.

**AWS Databases**

1. **Amazon RDS (MySQL/PostgreSQL)**: Relational database for ACID-compliant transactions.
2. **Amazon DynamoDB**: High-performance key-value store for real-time market data.
3. **Amazon Redshift**: Data warehousing and analytics for sales data and customer behavior.

**Transaction Database**

1. **Apache Cassandra**: Distributed, scalable database for transaction history and trade matching engine.

**Integration Strategies**

1. **Data Replication**: Amazon Data Migration Service (DMS) for replicating data between traditional and AWS databases.
2. **API Gateway**: AWS API Gateway for creating a unified API layer accessing data from multiple databases.
3. **Serverless Functions**: AWS Lambda for interacting with databases and performing tasks like data processing and notifications.

**Data Flow**

1. User data → MySQL
2. Customer profiles → MongoDB
3. Market data → MongoDB and Amazon DynamoDB
4. Transaction data → Apache Cassandra
5. Trade history → PostgreSQL
6. Analytics data → Amazon Redshift

**Key Considerations**

1. **Data Consistency**: Ensure data consistency across databases using replication and validation mechanisms.
2. **Performance**: Optimize performance through indexing, query optimization, and caching.
3. **Cost**: Monitor costs associated with each database and AWS service.
4. **Management**: Establish clear management procedures for backups, security, scaling, and updates.

**Benefits**

1. **Flexibility**: Leverage strengths of each database for specific use cases.
2. **Scalability**: Handle high traffic and large data volumes.
3. **Performance**: Optimize query performance and data processing.
4. **Cost-Effective**: Reduce costs by choosing the right database for each use case.

**Components**

1. User Management
2. Crypto Management
3. Banking Management
4. Transaction Management
5. Cross-Border Transactions Management
6. Analytics and Reporting

**Features**

1. Local Currency to Crypto
2. P2P (Crypto Currency)
3. Crypto to Local Currency
4. P2P ( Local Currency)
5. P2P (Local Currency To Other International Country Currency)
6. P2P (Other International Country Currency To Local Currency)
7. Real-time Market Data
8. Advanced Analytics and Reporting
9. Secure Authentication and Authorization

**Transaction Database (Apache Cassandra) Schema**

1. transactions table: transaction\_id, user\_id, currency, amount, timestamp
2. order\_books table: order\_id, user\_id, currency, amount, price
3. trade\_history table: trade\_id, user\_id, currency, amount, price, timestamp

**Tech Stack:**

1. **Frontend:** HTML, CSS, React.tsx, tsx, java, kotlin
2. **Backend & Security and encryption:** C#, Perl, Ruby, Python
3. **Encryption algorithms:** CRYSTALS-Kyber, FALCON, Bcrypt
4. **Hashing Algorithms:** SHA-256, Skein, Grøstl, Whirlpool, Streebog
5. **Database:** MongoDB, MySQL, PostgreSQL, Amazon RDS, Amazon DynamoDB, Amazon Redshift, Apache Cassandra
6. **Integration:** Amazon DMS, AWS RDBS, AWS API Gateway, AWS Lambda
7. **Cryptocurrency Integration:** APIs (cryptoapi.io, tradinview), Web3.js, own liquidity pool
8. **Banking Integration APIs & SDKs:** debit, credit, rupay, visa, master card, amex, UPI SDKs
9. **Security**: OpenSSL, SSL/TLS, 2FA, JWS, Oauth Token, Encryption, access controls
10. **AWS:** Amazon EC2, Amazon RDS, Amazon S3, Amazon CloudWatch
11. **Orchestration & Containerization & Deployment:** Podman& Kubernetes
12. **(CI/CD):** GitLab CI/CD & Jenkins

**Know Your Customer(KYC)**

**KYC (Know Your Customer) Verification Process**

**There are 3 Different stages**:

**Stage 1: Mobile Number Verification**

**Stage 2: Document Verification**

**Stage 3: Facial Verification**

**Stage 4: Bank Account Verification**

**Stage 1: Mobile Number Verification**

1. **Collect User Information:** Gather the user's name and mobile number during registration.
2. **Send OTP:** Generate a unique one-time password (OTP) and send it to the provided mobile number via SMS or a similar channel.
3. **Verify OTP:** Prompt the user to enter the received OTP to confirm ownership of the mobile number.
4. **Telecom Database Check:** If available and feasible, cross-verify the provided name and mobile number with the telecom provider's database to enhance security.

**Stage 2: Document Verification:**

1. **Document Submission:** Request the user to upload a clear copy of a valid government-issued ID (e.g., passport, driver's license, national ID card).
2. **Data Extraction:** Use OCR or other image processing techniques to extract relevant information from the uploaded document, such as name, date of birth, and document number.
3. **Document Authentication:** Verify the authenticity of the document using a reputable third-party identity verification service.
4. **Data Matching:** Compare the extracted information with the data provided during registration and the information verified in Stage 1 (mobile number verification).

**Stage 3: Facial Recognition:**

1. **Liveness Check:** Conduct a liveness check to ensure the user is a real person and not a spoofing attempt. This can involve asking the user to perform specific actions (e.g., blink, smile, head movement) while capturing a video or image.
2. **Facial Feature Extraction:** Extract facial features from the captured image or video.
3. **Biometric Comparison:** Compare the extracted facial features with the photo on the uploaded ID and any previously stored biometric data (if applicable).

**Stage 4: Bank Account Verification:**

1. **Bank Selection:** Allow the user to select their bank from a list of supported banks.
2. **Account Details:** Request the user to enter their bank account number and other relevant details.
3. **Bank Verification:** Verify the provided bank account details using the bank's API or a third-party verification service.
4. **Name Matching:** Ensure that the name associated with the bank account matches the previously verified information (from mobile number, document, and facial recognition).
5. **Additional Checks (Optional):** Consider implementing additional checks like address verification or credit history checks based on risk assessment.

**Key Considerations:**

1. **Data Privacy:** Handle all collected user data securely and in compliance with applicable data protection regulations (e.g., GDPR, CCPA).
2. **User Experience:** Design the verification process to be user-friendly and minimize friction for the user.
3. **Error Handling:** Implement robust error handling mechanisms to provide clear and informative feedback to the user in case of failures during any stage of the verification process.
4. **Compliance:** Adhere to all relevant KYC/AML regulations and industry standards.
5. **Continuous Improvement:** Regularly review and update the verification process to address emerging threats and improve accuracy.

**Maya-Exchange-C2LC&LC2C&C2C&LC2LC**

To implement a P2P (PeertoPeer) exchange system that supports the described features and user flow scenarios, we need to design a robust architecture that integrates blockchain technology, UPI infrastructure, and exchange logic. Below is a detailed blueprint for implementing each feature and scenario.

1. **General Architecture**

The system will support both direct P2P payments (like PhonePe/GPay) and a P2P ad trade center for cryptocurrency and local currency exchanges. The architecture will include:

1. **Frontend**
   * + A user friendly interface (web or mobile app) where users can:
     + Log in and manage their accounts.
     + Initiate direct P2P payments or browse the P2P ad trade center.
     + View transaction history and balances.
2. **Backend**
   * + Handles business logic, including:
     + User authentication and authorization.
     + Exchange rate calculations.
     + Transaction validation and execution.
     + Integration with blockchain networks and UPI infrastructure.
3. **Blockchain Network**
   * + Used for cryptocurrency transactions.
     + Smart contracts may be deployed to automate certain processes (e.g., Maya Pay).
4. **UPI Infrastructure:**
   * + Direct integration with UPI networks for local currency transactions.
     + Handle settlement, reconciliation, and fraud detection.
5. **Database:**
   * + Stores user data, transaction history, and wallet balances.
     + Includes tables for: Users: `id`, `name`, `email`, `crypto\_wallet\_address`, `upi\_id`.
     + Transactions: `id`, `user\_id`, `type` (LC2LC, C2C, etc.), `amount`, `status`, `timestamp`.
6. **P2P Ad Trade Center:**
   * + A marketplace where users can post ads for buying/selling cryptocurrency or local currency.
     + Includes features like order matching, escrow services, and dispute resolution.
7. **Workflows**
   1. **Direct P2P Payments (Maya Pay)**

**Scenario 1: Local Currency to Local Currency (LC2LC)**

* + - 1. User Logs In and Selects "Send":
         * The user logs into the platform and navigates to the "Send" section.
      2. Chooses Recipient and Enters Amount:
         * The user selects the recipient (via phone number, email, or UPI ID) and enters the amount to send.
      3. Maya Exchange Transfers Funds via UPI Infrastructure:
         * The backend verifies the sender's UPI balance.
         * Initiates a direct UPI transfer from the sender's account to the recipient's account using your PSP infrastructure.
         * Updates the transaction status in the database.
      4. Notifications:
         * Both sender and recipient receive notifications about the successful transfer.

**Scenario 2: Crypto to Crypto (C2C)**

* + - 1. User Selects "Send" and Chooses Recipient:
         * The user selects the recipient's cryptocurrency wallet address.
      2. Selects Source and Target Cryptocurrencies:
         * The user chooses the source cryptocurrency (e.g., Bitcoin) and target cryptocurrency (e.g., Ethereum).
      3. Maya Exchange Executes the Trade:
         * The backend calculates the exchange rate using an external API.
         * Deducts the source cryptocurrency from the sender's wallet.
         * Credits the equivalent target cryptocurrency to the recipient's wallet.
         * Records the transaction in the database.
      4. Updates Balances :
         * Both sender and recipient see updated wallet balances in the app.
  1. **P2P Ad Trade Center:**

**Scenario 3: Posting an Ad:**

* + - 1. User Creates an Ad:
         * The user logs in and navigates to the "Post Ad" section.
         * Specifies whether they want to buy or sell (e.g., "Sell Bitcoin for INR").
         * Sets the price, minimum/maximum transaction limits, and payment methods (e.g., UPI, bank transfer).
      2. Ad is Published:
         * The ad is listed in the P2P ad trade center for other users to view.

**Scenario 4: Responding to an Ad:**

* + - 1. User Browses Ads:
         * The user browses available ads in the P2P ad trade center.
         * Filters by cryptocurrency, local currency, or payment method.
      2. User Initiates a Trade:
         * The user selects an ad and enters the desired transaction amount.
         * Confirms the trade, locking in the price.
      3. Escrow Service:
         * The system holds the cryptocurrency in escrow until the payment is confirmed.
      4. Payment is Made:
         * The buyer makes the payment using the specified method (e.g., UPI, bank transfer).
         * The seller confirms receipt of payment.
      5. Cryptocurrency is Released:
         * Once payment is confirmed, the system releases the cryptocurrency from escrow to the buyer's wallet.
      6. Dispute Resolution:
         * If there’s a dispute, the system allows both parties to raise a complaint, which is resolved by moderators.

1. **Key Features**
2. **Direct P2P Payments**
   * **UPI Integration:** Direct integration with UPI networks for seamless local currency transfers.
   * **Instant Notifications:** Realtime updates for successful and failed transactions.
   * **Transaction History:** Users can view all past transactions in a clean, organized interface.
3. **P2P Ad Trade Center**
   * **Order Matching:** Automatically match buyers and sellers based on preferences (e.g., price, payment method).
   * **Escrow System:** Securely hold funds during trades to prevent fraud.
   * **Dispute Resolution:** Moderators handle disputes between buyers and sellers.
   * **Reputation System:** Users earn ratings based on successful trades, improving trust in the marketplace.
4. **MultiCurrency Support**
   * Support for multiple cryptocurrencies (e.g., Bitcoin, Ethereum) and local currencies (e.g., INR, USD).
   * Realtime exchange rates for accurate conversions.
5. **Security and Compliance**
   * **KYC/AML Verification:** Verify user identities to comply with regulations.
   * **Fraud Detection:** Use machine learning to detect suspicious activity.
   * **Encryption:** Protect sensitive data (e.g., private keys, UPI IDs).
6. **Challenges and Considerations**
   1. **Regulatory Compliance**
      * Ensure compliance with KYC/AML laws and local financial regulations.
      * Obtain necessary licenses to operate as a PSP.
   2. **Security**
      * Protect sensitive data using encryption and multifactor authentication (MFA).
      * Regularly audit your systems for vulnerabilities.
   3. **Scalability**
      * Build a scalable infrastructure to handle high transaction volumes.
      * Use containerization (e.g., Podman, Kubernetes) and cloud services (e.g., AWS, GCP) deployment.
   4. **Trust and Reputation**
      * Build trust in the P2P ad trade center by implementing a robust reputation system and dispute resolution process.

**User Accessible Features on Maya Exchange**

1. **Dashboard:**
   * Overview of account balance
   * Recent transactions
   * Quick links to common actions (send, receive, convert)
2. **User Profile:**
   * Edit personal information
   * Update contact details
   * Manage security settings (password, 2FA)
3. **KYC/AML Verification:**
   * Complete mobile number verification
   * Upload and verify documents (ID, passport, driver's license)
   * Perform facial recognition
   * Verify bank account details
4. **Send and Receive:**
   * Local Currency to Local Currency (LC2LC):
   * Send local currency to other users via UPI
   * Crypto to Crypto (C2C):
   * Send cryptocurrencies to other users
   * Crypto to Local Currency (CLC):
   * Convert and send cryptocurrencies to local currency
   * Local Currency to Crypto (LC2C):
   * Convert and send local currency to cryptocurrencies
5. **Transaction History:**
   * View past transactions
   * Filter by type (LC2LC, C2C, CLC, LC2C)
   * Download transaction history
6. **Convert Currency:**
   * Convert between different cryptocurrencies
   * Convert between local currencies and cryptocurrencies
   * View realtime exchange rates
7. **Staking and Earning:**
   * Stake cryptocurrencies to earn interest
   * Participate in liquidity pools
   * Earn rewards from staking and lending
8. **Trading**:
   * Spot Trading:

Trade cryptocurrencies against each other (e.g., BTC/USDT, ETH/BTC).

* + Margin Trading:

Trade with borrowed funds to increase potential gains (and risks).

* + Futures Trading:

Trade derivative contracts (long or short positions).

* + Options Trading:

Trade options contracts for hedging or speculative purposes.

1. **Deposit and Withdraw:**
   * Fiat Deposits:

Deposit fiat currency via bank transfer, credit/debit card, UPI.

* + Crypto Deposits:

Deposit cryptocurrencies to your Maya Exchange wallet.

* + Fiat Withdrawals:

Withdraw fiat currency to your bank account.

* + Crypto Withdrawals:

Withdraw cryptocurrencies to external wallets.

1. **CrossBorder Transactions:**
   * Send and receive funds internationally
   * Convert currencies for crossborder transactions
2. **P2P Trading:**
   * Trade cryptocurrencies with other users
   * Local currency to crypto P2P transactions
   * International currency to local currency P2P transactions
3. **RealTime Market Data:**
   * View live market prices
   * Track market trends and news
4. **Analytics and Reporting:**
   * Generate custom reports
   * Analyze trading performance
   * Track portfolio growth
5. **Security:**
   * Enable twofactor authentication (2FA)
   * Manage API keys
   * View and manage session activity
6. **Support:**
   * Access help center and FAQs
   * Contact customer support
   * Submit tickets for assistance
7. **Settings:**
   * Language preferences
   * Notification settings
   * Privacy settings
8. **Referral Program:**
   * Invite friends and earn rewards

Track referral status

* + Loyalty Programs:

Earn points for transactions

Redeem points for discounts or rewards

1. **NFT Marketplace (Future Feature):**
   * Buy and sell NFTs
   * Mint and list NFTs
2. **Advanced Features:**
   * Grid Trading:

Automate buying low and selling high

* + Copy Trading:

Copy trades of successful traders

* + Algorithmic Trading:

Use prebuilt or custom trading bots

1. **DeFi Integration:**
   * Access decentralized finance (DeFi) protocols
   * Participate in yield farming, liquidity mining, and other DeFi activities
2. **Maya Open Chain Integration:**
   * Interact with Maya Open Chainbased dApps
   * Swap tokens on Maya Open Chain
3. **Maya Visa Card:**
   * Spend cryptocurrencies at merchants
   * Earn rewards and cashback
4. **Maya Savings:**
   * Flexible savings: earn interest on idle funds
   * Fixed savings: lock funds for higher returns
5. **Maya Launchpad/Launchpool:**
   * Participate in token sales and initial coin offerings (ICOs)
   * Earn new tokens through staking
6. **Maya DEX:**
   * Trade on Maya's decentralized exchange
   * Noncustodial trading
7. **Maya Charity Foundation:**
   * Donate to charitable causes
   * Participate in social impact projects
8. **Maya Academy:**
   * Access educational resources
   * Learn about cryptocurrencies and blockchain technology