**Functional Requirement Document**

**Project Name: Custom RPC NFT Minting Bot**

**1. Introduction**

This document outlines the functional requirements for the **Custom RPC NFT Minting Bot**. The bot facilitates NFT minting on the Solana blockchain, allowing users to select RPC nodes, manage wallets, track fees, and execute transactions via Telegram.

**2. Functional Requirements**

**2.1 User Features**

1. **RPC Selection:** Users can choose a custom Solana RPC for transactions.
2. **NFT Minting:** Users specify the number of NFTs to mint.
3. **Stage Selection:** Users select the minting stage (presale, public sale, etc.).
4. **Wallet Management:** Users can add and select wallets for minting.
5. **Transaction Monitoring:** The bot provides real-time transaction fees and compute unit details.
6. **Telegram Integration:** Users interact with the bot via Telegram for executing minting operations.

**2.2 Admin Features**

1. **Manage RPC Nodes:** Admins can add, update, and remove RPC nodes.
2. **Wallet Management:** Admins can oversee and manage user wallets.
3. **Transaction Logging:** All transactions are recorded for auditing and debugging purposes.

**2.3 Automation Features**

1. **RPC Health Check:** The bot automatically tests RPC availability and responsiveness.
2. **Gas & Compute Unit Estimation:** Estimates transaction fees before minting to optimize costs.

**3. Non-Functional Requirements**

* **Security:** Encrypt sensitive data (private keys, API keys, user information).
* **Scalability:** Support multiple concurrent users minting NFTs.
* **Performance:** Optimize RPC selection for fast and efficient transactions.
* **Reliability:** Implement automatic fallback to a healthy RPC if one fails.
* **Logging & Monitoring:** Store logs for tracking transactions, debugging, and auditing.

**4. Technology Stack**

| **Component** | **Technology** |
| --- | --- |
| Backend | Node.js, Express.js |
| Database | MongoDB |
| Blockchain | Solana Web3.js |
| Messaging | Telegram Bot API |
| API Calls | Axios |
| Environment | dotenv |

**5. System Architecture**

1. **User sends a command to the Telegram bot.**
2. **Bot forwards API requests to the Express.js backend.**
3. **Backend validates input, retrieves RPC and wallet data from MongoDB.**
4. **Backend interacts with the Solana blockchain via RPC for minting.**
5. **Transaction results are logged and sent back to the user via Telegram.**

**6. API Endpoints**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| POST | /api/mint | Mint NFTs based on user input |
| GET | /api/check-rpc | Check RPC status and availability |
| POST | /api/add-wallet | Add a new wallet |
| GET | /api/wallets | Retrieve all user wallets |
| POST | /api/add-rpc | Add a new RPC endpoint |
| GET | /api/rpcs | Get a list of available RPCs |

**7. Database Schema**

**Wallet Model**

{

"address": "string",

"privateKey": "string"

}

**RPC Model**

{

"url": "string",

"status": "boolean"

}

**8. Telegram Bot Commands**

| **Command** | **Description** |
| --- | --- |
| /start | Start bot interaction |
| /mint | Initiate NFT minting process |
| /rpc | List available RPCs |
| /wallets | Display user's wallets |
| /fees | Show estimated gas fees and compute units |

**9. Future Enhancements**

* **Whitelist support for presale minting.**
* **Multi-wallet minting support.**
* **Web dashboard for transaction monitoring.**
* **Fee optimization based on network conditions.**
* **Failover RPC selection for uninterrupted transactions.**

**10. Deployment Plan**

1. **Setup MongoDB database (local or cloud).**
2. **Deploy the Express.js backend on a cloud platform (e.g., VPS, AWS, DigitalOcean).**
3. **Deploy the Telegram bot and integrate it with the backend.**
4. **Conduct testing on Solana testnet before mainnet launch.**

**11. Conclusion**

The **Custom RPC NFT Minting Bot** is designed to provide an automated and efficient NFT minting experience. By leveraging **Telegram for user interaction** and **Solana Web3.js for blockchain integration**, the bot ensures flexibility, reliability, and security for seamless NFT minting operations.