# Project Specification Document

## Project Name: Custom RPC NFT Minting Bot

## Overview

This project aims to develop an **automated NFT minting bot** that allows users to:

* Select a **custom RPC** for minting.
* Choose the **number of NFTs** to mint.
* Select the **minting stage** (e.g., presale, public sale).
* Choose a **wallet** for minting.
* Monitor **transaction fees & compute units**.
* Interact with the bot via **Telegram** for easy execution.

The backend will be built using **Node.js & Express**, with **MongoDB** for data storage, **Solana Web3 SDK** for blockchain interactions, and **Telegram Bot API** for user interaction.

## 1. Functional Requirements

### 1.1 User Features

1. **RPC Selection:** Users can choose a custom Solana RPC for transactions.
2. **NFT Minting:** Users can specify the number of NFTs to mint.
3. **Stage Selection:** Users can select between presale and public sale stages.
4. **Wallet Management:** Users can add and select wallets for minting.
5. **Transaction Monitoring:** The bot displays transaction fees and compute units.
6. **Telegram Integration:** Users interact via a Telegram bot for command execution.

### 1.2 Admin Features

1. **Manage RPC Nodes:** Admins can add, update, or remove RPC nodes.
2. **Wallet Management:** Admins can add or remove user wallets.
3. **Transaction Logging:** Logs are stored for auditing and debugging.

### 1.3 Automation Features

1. **RPC Health Check:** Automatically tests RPCs for availability.
2. **Gas & Compute Unit Estimation:** Estimates the fees before executing a mint transaction.

## 2. Non-Functional Requirements

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

## 3. Tech Stack

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

## 4. System Architecture

1. **User interacts with the Telegram bot.**
2. **Bot sends API requests to the Express.js server.**
3. **Server validates user input and fetches RPC & wallet data from MongoDB.**
4. **Server interacts with Solana blockchain via RPC for minting.**
5. **Transaction details are logged and displayed back to the user.**

## 5. API Endpoints

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

## 6. Database Schema

### Wallet Model

{

"address": "string",

"privateKey": "string"

}

### RPC Model

{

"url": "string",

"status": "boolean"

}

## 7. Telegram Bot Commands

| **Command** | **Description** |
| --- | --- |
| /start | Welcome message |
| /mint | Start NFT minting process |
| /rpc | List available RPCs |
| /wallets | List user wallets |
| /fees | Estimate gas fees and compute units |

## 8. Future Enhancements

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

## 9. Deployment Plan

1. **Setup MongoDB database (local or cloud).**
2. **Deploy the Express.js server on a cloud platform (VPS, AWS, etc.).**
3. **Deploy the Telegram bot and link it to the backend.**
4. **Test with testnet before launching on mainnet.**

## 10. Conclusion

This project provides a **powerful & automated** way to mint NFTs using **custom RPCs**, ensuring **efficiency & flexibility**. It leverages **Telegram** for user interaction and **Solana Web3.js** for blockchain operations, making **NFT minting seamless & accessible**.