## Backend Setup Documentation (Detailed)

### 1. Setting Up the Backend

#### Step 1: Clone the Repository

Clone the project from your version control system GitHub(Also Attached zip):

Navigate into the project directory:

#### Step 2: Install Dependencies

Ensure you have Node.js installed. Install all the required dependencies with the following command:

npm install

This will install all the dependencies listed in the package.json file, including Express and any other required modules.

#### Step 3: Setup Environment Variables

Create a .env file in the root of the project if not already there:

#### Step 4: Start the Application

To run the application locally:

npm run dev

## 

## API Details (With Detailed Explanation)

#### **1. Create KYC Entry**

POST /create

* Function:
  + Accepts KYC details, including image, wallet address, and custom metadata.
  + Validates input, stores image and metadata, and calls mintNFT to create an NFT.
  + On successful NFT creation, updates the KYC entry with the NFT address.
* Request Example:

{

"walletAddress": "0x123...",

"name": "User NFT",

"symbol": "USER",

"description": "User's first NFT",

"customMetadata": "{\"level\":\"basic\"}"

}

#### 2. **Delete KYC Entry**

DELETE /deleteKYC/:walletAddress

* Function:
  + Deletes a KYC entry based on the provided walletAddress.

#### 3. **Display KYC Entry**

GET /display

* Function:
  + Retrieves KYC data based on the walletAddress.
* Response Example:

{

"walletAddress": "0x123...",

"referenceId": "NFT\_ADDRESS",

"metadata": {

"name": "User NFT",

"kycStatus": "null"

}

}

#### 4. **Update KYC Entry**

PUT /update

* Function:
  + Updates metadata or kycStatus for an existing KYC entry.

#### 5. **Update KYC Status**

POST /updateKYC

* Function:
  + Specifically handles updates to kycStatus, referenceId, and explorerLink.
  + Ensures consistency by updating both main fields and metadata.

#### 6. **Retrieve Image for KYC Entry**

GET /image/:id

* Function:
  + Fetches and returns the image binary for a specific KYC entry based on its ID.

#### 7. **Retrieve Metadata for KYC Entry**

GET /metadata/:id

* Function:
  + Serves the metadata of a KYC entry in JSON format.