Centerlized NFT Marketplace (APIs)

1- Signup

Description:

1- Convert the password into a hash before storing it in the database.
2- For email verification, generate a verification link and send it to the provided user email address. Use sendgrid to send email.

Request:

```
{
  "name": "John Doe",
  "email": "johndoe@example.com",
  "password": "password123"
}
```

Response:

```
{
  "status": "success",
  "message": "Please check your email to verify your account."
}
```

2- Verify-email

Description:

Validate the token received by the user via email.

Request:

```
{
"token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
}
```

Response:

```
{
  "status": "success",
  "message": "Email verified successfully. Your account is now active."
}
```

3- Login

Description:

1- Authenticate user by validating their credentials and generate access token for subsequent authenticated requests.
2- Token should be expired after 10 minutes.

Request:

```
{
  "email": "johndoe@example.com",
  "password": "password123"
}
```

Response:

```
{
  "status": "success",
  "message": "Login successful",
  "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJKb2huIERvZSIsImlhdCI6MTYxNjI
zOTAyMn0.3DcHf_hDb_xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "user": {
    "id": "123456789",
    "name": "John Doe",
    "email": "johndoe@example.com"
  }
}
```

4- Mint-nft:

Description:

1- Upload metadata to IPFS (Pinata).
2- Sign and send transaction for minting using a private key. Private key should be in .env file.

Request:

```
{
  "name": "My Awesome NFT",
  "description": "This is an amazing piece of artwork.",
  "image": "file",
  "wallet": "0x021abcd…",
  "attributes": {
    "artist": "John Doe",
    "year": 2023,
    "edition": "1 of 10"
  }
}
```

Response:

```
{
  "status": "success",
"message": "Transaction sent successfully",
  "hash": "0xcv2ka232….",
  }
```

3- List-nft

Description:

1- Check NFT received in escrow wallet
2- Verify seller is owner.

Request:

```
{
  "tokenId": "2",
  "price": 1.5,
  "transfer_hash": "0x021abcd…",
```

```
  "wallet": "0x021abcd…",
}
```

Response:

```
{
  "status": "success",
  "message": "NFT listed successfully",
}
```

4- Listed-nfts

Description:

API to get all listed NFTs on marketplace.

Response:

```
{
  "status": "success",
  "message": "Listed NFTs retrieved successfully",
  "nfts": [
    {

      "tokenId": "2",
      "price": 1.5,
      "seller": "0x1234567890abcdef",
      "metadata": {
        "name": "My Awesome NFT",
        "description": "This is an amazing piece of artwork.",
        "image": "https://example.com/image.jpg",
        "attributes": {
          "artist": "John Doe",
          "year": 2023,
          "edition": "1 of 10"
        }
      }
    },
    {
```

```
      "tokenId": "3",
      "price": 2.0,
      "seller": "0xabcdef123456789",
      "metadata": {
        "name": "Another NFT",
        "description": "A different piece of artwork.",
        "image": "https://example.com/another-image.jpg",
        "attributes": {
          "artist": "Jane Smith",
          "year": 2022,
          "edition": "1 of 5"
        }
      }
    }
  ]
}
```

5- Buy-nft:

Description:

1- Check eth received in escrow wallet.
2- Transfer NFT to buyer.
3- Transfer 2% ETH to escrow wallet
4- Transfer remaining to seller

Request:


```
{
  "nftId": "123456789",
  "buyer": "0x0987654321…..",
  "paymentHash": "0xabc345wx…..",
}
```


Response:


```
{
  "status": "success",
  "message": "Transaction sent successfully ",
  "transactionId": "0xabc123def456",
```

```
}
```

6- Nft-detail

Request:

NFT id in params

Response:

```
{
  "status": "success",
  "message": "NFT details retrieved successfully",
  "nft": {
    "tokenId": "2",
    "name": "My Awesome NFT",
    "description": "This is an amazing piece of artwork.",
    "image": "https://example.com/image.jpg",
    "attributes": {
      "artist": "John Doe",
      "year": 2023,
      "edition": "1 of 10"
    },
    "owner": "0x1234567890abcdef"
  }
}
```

Note:

1- Use event listener of etherJS library for transaction confirmation.
2- Use proper folder structure and break your code into small, reusable modules
3- Implement error handling mechanisms, including proper error object creation, try-catch blocks.
4- Implement appropriate validations on request body fields.
5- Use .env file for configuration variables like database url, escrow wallet public and private key, sendgrid api key etc.