

- graph-nft-marketplace-fcc

The screenshot shows the VS Code editor with the Explorer sidebar on the left and the editor window on the right. The Explorer sidebar shows the project structure for 'graph-nft-marketplace-fcc', including files like 'package.json', 'README.md', 'schema.graphql', 'subgraph.yaml', 'tsconfig.json', 'yarn.lock', and 'src'. The editor window displays the 'TS mapping.ts' file, which contains the following code:

```

1 import { BigInt, Address } from "@graphprotocol/graph-ts"
2 import {
3   NftMarketplace,
4   ItemBought as ItemBoughtEvent,
5   ItemCanceled as ItemCanceledEvent,
6   ItemListed as ItemListedEvent,
7 } from "../generated/NftMarketplace/NftMarketplace"
8 import { ItemListed, ActiveItem, ItemBought, ItemCanceled } from "../generated/schema"
9
10 export function handleItemListed(event: ItemListedEvent): void {
11   let itemListed = ItemListed.load(
12     getIdFromEventParams(event.params.tokenId, event.params.nftAddress)
13   )
14   let activeItem = ActiveItem.load(
15     getIdFromEventParams(event.params.tokenId, event.params.nftAddress)
16   )
17   if (!itemListed) {
18     itemListed = new ItemListed(
19       getIdFromEventParams(event.params.tokenId, event.params.nftAddress)
20     )
21   }
22   if (!activeItem) {
23     activeItem = new ActiveItem(
24       getIdFromEventParams(event.params.tokenId, event.params.nftAddress)
25     )
26   }
27   itemListed.seller = event.params.seller
28   activeItem.seller = event.params.seller
29
30   itemListed.nftAddress = event.params.nftAddress
31   activeItem.nftAddress = event.params.nftAddress
32
33   itemListed.tokenId = event.params.tokenId

```

- hardhat-nft-marketplace-fcc

The screenshot shows the VS Code editor with the Explorer sidebar on the left and the editor window on the right. The Explorer sidebar shows the project structure for 'hardhat-nft-marketplace-fcc', including files like 'NftMarketplace.sol', 'contracts', 'sublsson', 'test', 'utils', 'env.example', '.gitignore', '.npmignore', '.prettierrc', '.solhint.json', 'hardhat.config.js', 'helper-hardhat-config.js', 'package.json', 'README.md', 'yarn.lock', and 'nextjs-nft-marketplace-moralis-fcc'. The editor window displays the 'NftMarketplace.sol' file, which contains the following code:

```

9 error PriceNotMet(address nftAddress, uint256 tokenId, uint256 price);
10 error ItemNotForSale(address nftAddress, uint256 tokenId);
11 error NotListed(address nftAddress, uint256 tokenId);
12 error AlreadyListed(address nftAddress, uint256 tokenId);
13 error NoProceeds();
14 error NotOwner();
15 error NotApprovedForMarketplace();
16 error PriceMustBeAboveZero();
17
18 contract NftMarketplace is ReentrancyGuard {
19   struct Listing {
20     uint256 price;
21     address seller;
22   }
23
24   event ItemListed(
25     address indexed seller,
26     address indexed nftAddress,
27     uint256 indexed tokenId,
28     uint256 price
29   );
30
31   event ItemCanceled(
32     address indexed seller,
33     address indexed nftAddress,
34     uint256 indexed tokenId
35   );
36
37   event ItemBought(
38     address indexed buyer,
39     address indexed nftAddress,
40     uint256 indexed tokenId

```

- nextjs-nft-marketplace-moralis-fcc

The screenshot shows the VS Code editor interface. On the left, the Explorer sidebar displays the project structure for 'nextjs-nft-marketplace-moralis-fcc'. The file 'index.js' is selected under the 'pages' directory. The main editor window shows the code for 'index.js', which is a Next.js page component. The code imports 'Image' from 'next/image', 'styles' from './styles/Home.module.css', and 'useMoralisQuery', 'useMoralis' from 'react-moralis'. It also imports 'NFTBox' from '../components/NFTBox'. The component defines a 'Home' function that uses 'useMoralis' to get 'isWeb3Enabled' and 'useMoralisQuery' to fetch 'listedNfts'. It then maps over 'listedNfts' to render 'NFTBox' components. The code is as follows:

```

1 import Image from "next/image"
2 import styles from "../styles/Home.module.css"
3 import { useMoralisQuery, useMoralis } from "react-moralis"
4 import NFTBox from "../components/NFTBox"
5
6 export default function Home() {
7   const { isWeb3Enabled } = useMoralis()
8   const { data: listedNfts, isFetching: fetchingListedNfts } = useMoralisQuery(
9     // TableName
10    // Function for the query
11    "ActiveItem",
12    (query) => query.limit(10).descending("tokenId")
13  )
14   console.log(listedNfts)
15
16   return (
17     <div className="container mx-auto">
18       <h1 className="py-4 px-4 font-bold text-2xl">Recently Listed</h1>
19       <div className="flex flex-wrap">
20         {isWeb3Enabled ? (
21           fetchingListedNfts ? (
22             <div>Loading...</div>
23           ) : (
24             listedNfts.map((nft) => {
25               console.log(nft.attributes)
26               const { price, nftAddress, tokenId, marketplaceAddress, seller } =
27                 nft.attributes
28               return (
29                 <div>
30                   <NFTBox
31                     price={price}
32                     nftAddress={nftAddress}
33                     tokenId={tokenId}

```

- nextjs-nft-marketplach-telegraph-fcc

The screenshot shows the VS Code editor interface. On the left, the Explorer sidebar displays the project structure for 'nextjs-nft-marketplach-telegraph-fcc'. The file 'index.js' is selected under the 'pages' directory. The main editor window shows the code for 'index.js', which is a Next.js page component. The code imports 'Image' from 'next/image', 'styles' from './styles/Home.module.css', and 'useMoralisQuery', 'useMoralis' from 'react-moralis'. It also imports 'NFTBox' from '../components/NFTBox'. The component defines a 'Home' function that uses 'useMoralis' to get 'isWeb3Enabled' and 'useMoralisQuery' to fetch 'listedNfts'. It then maps over 'listedNfts' to render 'NFTBox' components. The code is as follows:

```

1 import Image from "next/image"
2 import styles from "../styles/Home.module.css"
3 import { useMoralisQuery, useMoralis } from "react-moralis"
4 import NFTBox from "../components/NFTBox"
5
6 export default function Home() {
7   const { isWeb3Enabled } = useMoralis()
8   const { data: listedNfts, isFetching: fetchingListedNfts } = useMoralisQuery(
9     // TableName
10    // Function for the query
11    "ActiveItem",
12    (query) => query.limit(10).descending("tokenId")
13  )
14   console.log(listedNfts)
15
16   return (
17     <div className="container mx-auto">
18       <h1 className="py-4 px-4 font-bold text-2xl">Recently Listed</h1>
19       <div className="flex flex-wrap">
20         {isWeb3Enabled ? (
21           fetchingListedNfts ? (
22             <div>Loading...</div>
23           ) : (
24             listedNfts.map((nft) => {
25               console.log(nft.attributes)
26               const { price, nftAddress, tokenId, marketplaceAddress, seller } =

```