```javascript
const { ethers } = require("ethers");

const abi = [
  {
    "inputs": [],
    "stateMutability": "nonpayable",
    "type": "constructor"
  },
  {
    "anonymous": false,
    "inputs": [
      {
        "indexed": true,
        "internalType": "uint256",
        "name": "drugId",
        "type": "uint256"
      },
      {
        "indexed": false,
        "internalType": "string",
        "name": "drugName",
        "type": "string"
      },
      {
        "indexed": false,
        "internalType": "string",
        "name": "manufacturer",
        "type": "string"
      },
      {
        "indexed": false,
```

```json
      "internalType": "uint256",
      "name": "manufacturingDate",
      "type": "uint256"
    }
  ],
  "name": "DrugManufactured",
  "type": "event"
},
{
  "anonymous": false,
  "inputs": [
    {
      "indexed": true,
      "internalType": "uint256",
      "name": "drugId",
      "type": "uint256"
    },
    {
      "indexed": true,
      "internalType": "address",
      "name": "from",
      "type": "address"
    },
    {
      "indexed": true,
      "internalType": "address",
      "name": "to",
      "type": "address"
    },
    {
      "indexed": false,
      "internalType": "uint256",
```

```json
      "name": "transferDate",
      "type": "uint256"
     }
    ],
    "name": "DrugTransferred",
    "type": "event"
   },
   {
    "inputs": [],
    "name": "drugCount",
    "outputs": [
     {
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
     }
    ],
    "stateMutability": "view",
    "type": "function"
   },
   {
    "inputs": [
     {
      "internalType": "uint256",
      "name": "",
      "type": "uint256"
     }
    ],
    "name": "drugs",
    "outputs": [
     {
      "internalType": "string",
```

```json
      "name": "drugName",
      "type": "string"
     },
     {
      "internalType": "string",
      "name": "manufacturer",
      "type": "string"
     },
     {
      "internalType": "uint256",
      "name": "manufacturingDate",
      "type": "uint256"
     },
     {
      "internalType": "address",
      "name": "trackingHistory",
      "type": "address"
     }
    ],
    "stateMutability": "view",
    "type": "function"
   },
   {
    "inputs": [
     {
      "internalType": "uint256",
      "name": "_drugId",
      "type": "uint256"
     }
    ],
    "name": "getDrugDetails",
    "outputs": [
```

```json
      {
       "internalType": "string",
       "name": "",
       "type": "string"
      },
      {
       "internalType": "string",
       "name": "",
       "type": "string"
      },
      {
       "internalType": "uint256",
       "name": "",
       "type": "uint256"
      },
      {
       "internalType": "address",
       "name": "",
       "type": "address"
      }
     ],
     "stateMutability": "view",
     "type": "function"
    },
    {
     "inputs": [
      {
       "internalType": "uint256",
       "name": "drugId",
       "type": "uint256"
      },
      {
```

```json
    "internalType": "string",
    "name": "_drugName",
    "type": "string"
   },
   {
    "internalType": "string",
    "name": "_manufacturer",
    "type": "string"
   },
   {
    "internalType": "uint256",
    "name": "_manufacturingDate",
    "type": "uint256"
   }
  ],
  "name": "manufactureDrug",
  "outputs": [],
  "stateMutability": "nonpayable",
  "type": "function"
 },
 {
  "inputs": [],
  "name": "owner",
  "outputs": [
   {
    "internalType": "address",
    "name": "",
    "type": "address"
   }
  ],
  "stateMutability": "view",
  "type": "function"
```

```
    },
    {
      "inputs": [
        {
          "internalType": "uint256",
          "name": "_drugId",
          "type": "uint256"
        },
        {
          "internalType": "address",
          "name": "_to",
          "type": "address"
        }
      ],
      "name": "transferDrugOwnership",
      "outputs": [],
      "stateMutability": "nonpayable",
      "type": "function"
    }
]


if (!window.ethereum) {
  alert('Meta Mask Not Found')
  window.open("https://metamask.io/download/")
}


export const provider = new
ethers.providers.Web3Provider(window.ethereum);

export const signer = provider.getSigner();

export const address =
"0x00D83E73d2D5b6e658bc881B27b00EA005Ea49ec"
```

```
export const contract = new ethers.Contract(address, abi,
signer)
```