1. Карточка Pending NFT
2. Страница с Pending к покупке NFT
3. Переиспользуемый компонент пагинации
4. Карточка NFT к покупке

1.

```
"use client";

import { FC } from "react";
import Image from "next/image";
import BuyNFTwithAllowance_Button from "./BuyNFTwithAllowance_Button";
import { INFT_Pending_Card } from "~~/types/frontendTypes";

const PendingNFT_Card: FC<INFT_Pending_Card> = ({
  allowanceId,
  cid,
  hotelName,
  apartmentName,
  nftPrice,
  incomePerNft,
  nftQuantity,
  price,
  cb,
}) => {
  return (
    <div className="card p-4 desktop:p-6 w-cardMobile
desktop:w-cardDesktop items-center bg-bgCardDark shadow-xl">
      <div className="relative w-[296px] h-[176px] desktop:w-[272px]
desktop:h-[176px]">
        <Image
          priority
          alt="NFT image"

src={`${process.env.NEXT_PUBLIC_GATEWAY_URL}/ipfs/${cid}?pinataGate
wayToken=${process.env.NEXT_PUBLIC_PINATA_GATEWAY_KEY}`}
          fill
          className="object-cover"
          sizes="(min-width: 1200px) 640px, 220px"
        />
      </div>
```

```
      <div className="card-body mb-6 p-0 gap-0">
        <div className="mobile:max-desktop:mb-6 grow">
          <p className="mb-4 mt-4 text-1xl text-textMain font-medium
desktop:mt-6">
            {hotelName} {apartmentName}
          </p>
          <ul className="flex flex-wrap gap-y-4 [&>*]:basis-1/2
first:[&_span]:mb-2 desdtop:gap-36 text-sm">
            <li>
              <span className="block text-shadow ">NFT price:</span>
              <span className="font-medium text-textMain">
                {(BigInt(nftPrice) / BigInt(process.env.NEXT_PUBLIC_DECIMALS ||
1)).toString()}
              </span>
            </li>
            <li>
              <span className="block text-shadow ">Income per NFT:</span>
              <span className="font-medium text-textMain">
                {(BigInt(incomePerNft) /
BigInt(process.env.NEXT_PUBLIC_DECIMALS || 1)).toString()}
              </span>
            </li>
            <li>
              <span className="block text-shadow ">Quantity:</span>
              <span className="font-medium
text-textMain">{nftQuantity.toString()}</span>
            </li>
            <li>
              <span className="block text-shadow ">Total price:</span>
              <span className="font-medium text-white">
                <b>{(BigInt(price) / BigInt(process.env.NEXT_PUBLIC_DECIMALS
|| 1)).toString()}</b>
              </span>
            </li>
          </ul>
        </div>
      </div>
      <BuyNFTwithAllowance_Button allowanceid={BigInt(allowanceId)}
price={BigInt(price)} cb={cb} className="w-full">
        Complete transaction
```

```
      </BuyNFTwithAllowance_Button>
    </div>
  );
};

export default PendingNFT_Card;
```

2.

```
"use client";

import { useCallback, useEffect, useState } from "react";
import { useSearchParams } from "next/navigation";
import { GetAllowancesByAddressDocument, GetAllowancesByAddressQuery,
execute } from "~~/.graphclient";
import Dropdown from "~~/components/Dropdown";
import Gallery from "~~/components/Gallery";
import PendingNFT_Card from "~~/components/PendingNFT_Card";
import { dashboardNFTFilter } from "~~/dataTemp";
// import { useScaffoldReadContract } from "~~/hooks/scaffold-eth";
import { IOption } from "~~/types/frontendTypes";

// import { getAllowancesQueryByAddress } from "~~/utils/helpers";

// import { getAllowancesQueryByAddress } from "~~/utils/helpers";

// type IAllowances = {
//   nftalloweds: NFTallowed[];
//   propertyAddeds: PropertyAdded[];
//   apartmentAddeds: ApartmentAdded[];
//   apartmentTypeAddeds: ApartmentTypeAdded[];
// };

// export const createPendingNftGallery = ({ nftalloweds, propertyAddeds,
apartmentAddeds }: IAllowances) => {
//   const allowances = nftalloweds.map(({ ammountShares, apartmentId,
hotelId, cid, nftPrice, id }) => {
//     const hotel = propertyAddeds.find(hotel => hotel.hotelId === hotelId);
//     const apartment = apartmentAddeds.find(
```

```
//       apartment => apartment.hotelId === hotelId && apartment.apartmentId
=== apartmentId,
//     );
//     // const apartmentType = apartmentTypeAddeds.find(type => type.typeId
=== apartment?.typeId);
//     const title = hotel?.name + " " + apartment?.name;
//     return {
//       id,
//       title,
//       nftDetails: {
//         nftPrice: BigInt(nftPrice) /
BigInt(process.env.NEXT_PUBLIC_DECIMALS || 1) + " USDT",
//         incomePerNft: "120 USDT / year",
//         nftQuantity: ammountShares,
//         price: (BigInt(nftPrice) * BigInt(ammountShares)) /
BigInt(process.env.NEXT_PUBLIC_DECIMALS || 1) + " USDT",
//       },
//       imgUrl:
`${process.env.NEXT_PUBLIC_GATEWAY_URL}/ipfs/${cid}?pinataGatewayT
oken=${process.env.NEXT_PUBLIC_PINATA_GATEWAY_KEY}`,
//     };
//   });
//   return allowances;
// };

const PendingNFTs_Page = () => {
  const searchParams = useSearchParams();
  const to = (searchParams?.get("to") as `0x${string}`) || `0x0`;

  // const { data: allowances } = useScaffoldReadContract({
  //   contractName: "RealEstateERC1155",
  //   functionName: "getAllowances",
  //   args: [to],
  // });

  // const pendingNfts = allowances?.filter(({ isSold }) => isSold === false);

  const [allowances, setAllowances] =
useState<GetAllowancesByAddressQuery["nftalloweds"]>([]);
```

```
  const getAllowancesForAddress = useCallback(async () => {
    const allowancesForAddress = await
execute(GetAllowancesByAddressDocument, { to, isSold: false });
    setAllowances(allowancesForAddress.data.nftalloweds);
  }, [to]);

  useEffect(() => {
    getAllowancesForAddress();
  }, [getAllowancesForAddress]);

  const onNFTClick = (option: IOption) => console.log(option.value);
  return (
    <>
      <div className="flex flex-col desktop:flex-row desktop:justify-between">
        <h3 className="text-3xl text-textMain mb-6 text-center desktop:mb-12
desktop:text-left">Pending NFTs</h3>
        <div className="desktop:[&>*]:w-[200px]">
          <Dropdown options={dashboardNFTFilter} onChange={onNFTClick}
label="Sort by" />
        </div>
      </div>
      {allowances && allowances.length > 0 && (
        <Gallery
          collections={allowances}
          card={PendingNFT_Card}
          cb={getAllowancesForAddress}
          className="flex flex-wrap gap-8 items-center justify-center
desktop:justify-center"
        />
      )}
      {allowances && allowances.length === 0 && <p className="text-center
text-textMain text-2xl">There is empty</p>}
    </>
  );
};

export default PendingNFTs_Page;
```

3. import { Dispatch, FC, SetStateAction } from "react";

```
type IPagination = {
  pendingAllowancesAmount: number;
  soldAllowancesAmount: number;
  counter: number;
  setCounter: Dispatch<SetStateAction<number>>;
};

const Pagination: FC<IPagination> = ({ counter, setCounter,
pendingAllowancesAmount, soldAllowancesAmount }) => {
  const onNextClick = () => setCounter(prev => ++prev);
  const onpPrevClick = () => setCounter(prev => --prev);
  const isNextDisabled =
    pendingAllowancesAmount >= soldAllowancesAmount
      ? pendingAllowancesAmount / 10 < 10
      : soldAllowancesAmount / 10 < 10;

  return (
    <div className="join justify-center">
      <button onClick={onpPrevClick} className="join-item btn btn-accent
w-[56px]">
        «
      </button>
      <div className="join-item flex justify-center items-center p-3
bg-bgFooterDark text-textMain">
        Page {counter + 1}
      </div>
      <button disabled={isNextDisabled} onClick={onNextClick}
className="join-item btn btn-accent w-[56px]">
        »
      </button>
    </div>
  );
};
export default Pagination;
```

4.

```
import { FC } from "react";
import Image from "next/image";
```

```
import Action_Button from "./Action_Button";
import Bascket_Icon from "./icons/Bascket_Icon";

type INFT_Card_ForBuy = { id: string; title: string; price: string; imgUrl: string };

const NFT_Card_ForBuy: FC<INFT_Card_ForBuy> = ({ title, price, imgUrl })
=> {
  return (
    <div className="bg-bgCardDark p-6">
      <Image alt="NFT iamage" src={imgUrl} width={296} height={182}
sizes="100vw" className="mb-6" />
      <div className="flex justify-between">
        <div className="flex flex-col justify-between text-textMain text-1xl">
          <p>{title}</p>
          <p>{price}</p>
        </div>
        <Action_Button className="w-[58px] h-[58px]">
          <Bascket_Icon />
        </Action_Button>
      </div>
    </div>
  );
};

export default NFT_Card_ForBuy;
```