1. **Before delete update: “Are you sure you..”**

import React, { useState } from "react";

import { Alert, AlertIcon, Flex, Icon, Image, Skeleton, Spinner, Stack, Text,} from "@chakra-ui/react";

import { useRouter } from "next/router";

import { AiOutlineDelete } from "react-icons/ai";

import { BsChat, BsDot } from "react-icons/bs";

import { FaReddit } from "react-icons/fa";

import { IoArrowDownCircleOutline, IoArrowDownCircleSharp, IoArrowRedoOutline,

  IoArrowUpCircleOutline, IoArrowUpCircleSharp, IoBookmarkOutline,} from "react-icons/io5";

import { Post } from "../../../atoms/postsAtom";

import Link from "next/link";

import moment from "moment";

export type PostItemProps = {

  post: Post;

  userIsCreator: boolean;

  userVoteValue?: number;

  onVote: (event: React.MouseEvent<SVGElement, MouseEvent>, post: Post, vote:number, communityId:string) => void;

  onDeletePost: (post: Post) => Promise<boolean>;

  onSelectPost?: (post: Post) => void;

  homePage?: boolean;

};

const PostItem: React.FC<PostItemProps> = ({

  post,

  userIsCreator,

  userVoteValue,

  onVote,

  onSelectPost,

  onDeletePost,

  homePage

}) => {

  const [loadingImage, setLoadingImage] = useState(true);

  const [loadingDelete, setLoadingDelete] = useState(false);

  const singlePostView = !onSelectPost; // function not passed to [pid]

  const [error, setError]=useState(false);

  const router =useRouter();

  const [showMessage, setShowMessage] = useState(false);

  //

  const handleDelete = async (event: React.MouseEvent<HTMLDivElement, MouseEvent>) => {

    event.stopPropagation();

    if (!userIsCreator) {

      setShowMessage(true);

      setTimeout(() => {

        setShowMessage(false);

      }, 5000);

      return;

    }

    setLoadingDelete(true);

    try {

      const success = await onDeletePost(post);

      if (!success) throw new Error("Failed to delete post");

      if(singlePostView){

        router.push(`/r/${post.communityId}`)

      }

    } catch (error: any) {

      console.log("Error deleting post", error.message);

      setError(error.message);

    }

    setLoadingDelete(false);

  };

  return (

    <Flex

      border="1px solid"

      bg="white"

      borderColor= {singlePostView ? "white" : "gray.300"}

      borderRadius="8px 8px 8px 8px"

      cursor={singlePostView ? "unset" : "pointer"}

      \_hover={{ borderColor: singlePostView ? 'none' : "gray.500" }}

    >

      <Flex

        direction="column"

        align="center"

        bg={singlePostView ? "none" : "gray.100"}

        p={2}

        width="40px"

        borderRadius="8px 8px 8px 8px"

      >

        <Icon

          as={

            userVoteValue === 1 ? IoArrowUpCircleSharp : IoArrowUpCircleOutline

          }

          color={userVoteValue === 1 ? "brand.100" : "gray.400"}

          fontSize={22}

          cursor="pointer"

          onClick={(event)=>onVote(event, post, 1, post.communityId)}

        />

        <Text fontSize="9pt" fontWeight={600}>

          {post.voteStatus}

        </Text>

        <Icon

          as={

            userVoteValue === -1

              ? IoArrowDownCircleSharp

              : IoArrowDownCircleOutline

          }

          color={userVoteValue === -1 ? "#4379FF" : "gray.400"}

          fontSize={22}

          cursor="pointer"

          onClick={(event)=>onVote(event, post, -1, post.communityId)}

        />

    </Flex>

        <Flex direction="column" width="100%">

          {

            error && (

              <Alert status="error">

                <AlertIcon/>

                <Text mr={2}>{error}</Text>

              </Alert>

            )

          }

          <Stack spacing={1} p="10px 10px">

            <Stack direction="row" spacing={0.6} align="center" fontSize="9pt">

              {homePage && (

                  <>

                    {post.communityImageURL ? (

                      <Image

                        borderRadius="full"

                        boxSize="18px"

                        src={post.communityImageURL.toString()}  //.to string chatgpt solution

                        mr={2}

                        alt="communityLogo"

                      />

                    ) : (

                      <Icon as={FaReddit} fontSize={18} mr={1} color="blue.500" />

                    )}

                    <Link href={`r/${post.communityId}`}>

                      <Text

                        fontWeight={700}

                        \_hover={{ textDecoration: "underline" }}

                        onClick={(event) => event.stopPropagation()}

                      >{`r/${post.communityId}`}</Text>

                    </Link>

                    <Icon as={BsDot} color="gray.500" fontSize={8} />

                  </>

                )}

                <Text>Posted by u/{post.creatorDisplayName}{" "}

                  {moment(new Date(post.createdAt.seconds \* 1000)).fromNow()}

                </Text>

            </Stack>

            <Text fontSize="12pt" fontWeight={600} >{post.title}</Text>

            <Text fontSize="10pt">{post.body}</Text>

            {post.imageURL && (

              <Flex justify="center" align="center" p={2}>

                {loadingImage && (

                  <Skeleton height="200px" width="100%" borderRadius={4} />

                )}

                <Image

                  // width="80%"

                  // maxWidth="500px"

                  maxHeight="460px"

                  src={post.imageURL}

                  alt="Post Image"

                  display={loadingImage ? "none" : "unset"}

                  onLoad={() => setLoadingImage(false)}

                />

              </Flex>

            )}

          </Stack>

          <Flex ml={1} mb={0.5} color="gray.500" fontWeight={600}

           onClick={() => onSelectPost && post && onSelectPost(post)}>

            <Flex

              align="center"

              p="8px 10px"

              borderRadius={4}

              \_hover={{ bg: "gray.200" }}

              cursor="pointer"

            >

              <Icon as={BsChat} mr={2} />

              <Text fontSize="9pt">{post.numberOfComments}</Text>

            </Flex>

            {/\* <Flex

              align="center"

              p="8px 10px"

              borderRadius={4}

              \_hover={{ bg: "gray.200" }}

              cursor="pointer"

            >

              <Icon as={IoArrowRedoOutline} mr={2} />

              <Text fontSize="9pt">Share</Text>

            </Flex>

            <Flex

              align="center"

              p="8px 10px"

              borderRadius={4}

              \_hover={{ bg: "gray.200" }}

              cursor="pointer"

            >

              <Icon as={IoBookmarkOutline} mr={2} />

              <Text fontSize="9pt">Save</Text>

            </Flex>\*/}

           {true && (

            <Flex

              align="center"

              p="8px 10px"

              borderRadius={4}

              \_hover={{ bg: "gray.200" }}

              cursor="pointer"

              onClick={handleDelete}

            >

              {loadingDelete ? (

                <Spinner size="sm" />

              ) : (

                <>

                  <Icon as={AiOutlineDelete} mr={2} />

                  <Text fontSize="9pt">Delete</Text>

                </>

              )}

            </Flex>

             )}

            {showMessage &&

              <Flex  align="center"

                color="red.400"

                fontSize="9pt"

                borderRadius={4}>Only Post-Creator can delete the post

              </Flex>

            }

          </Flex>

      </Flex>

    </Flex>

  );

};

export default PostItem;