```tsx
import { trpc } from '@/shared/utils/trpc/trpc';
import { Heart } from 'lucide-react';
import React, { useEffect, useState } from 'react';
import { useAuthContext } from '@/shared/components/ui/auth.provider';
import { useToast } from '@/shared/components/ui/Toasts/use-toast';
import { cn } from '@/shared/utils/cn';

interface LikeButtonProps {
  itemId: string;
  type: 'comment' | 'reply';
  isLiked: boolean;
  likesCount: number;
  refetchLikes?: () => void;
}

export default function LikeButton({
  itemId,
  type,
  isLiked,
  likesCount,
  refetchLikes,
}: LikeButtonProps) {
  const { user } = useAuthContext();
  const toast = useToast();

  const [isLoading, setIsLoading] = useState(false);
```

```
const likesMutation = trpc.explorerOptions.likeItem.useMutation();
const unlikesMutation = trpc.explorerOptions.unlikeItem.useMutation();

const handleLike = async () => {
  if (!user) {
    toast.toast({
      description: 'Log in to perform this action',
      style: { color: 'red' },
    });
    return;
  }

  if (!itemId || !type || isLoading) return;

  setIsLoading(true);

  if (isLiked) {
    unlikesMutation
      .mutateAsync({ itemId, itemType: type })
      .then(() => {
        refetchLikes?.();
      })
      .catch((err) => {
        console.error(err);
      })
```

```
        .finally(() => setIsLoading(false));
    } else {
      likesMutation
        .mutateAsync({ itemId, itemType: type })
        .then(() => {
          refetchLikes?.();
        })
        .catch((err) => {
          console.error(err);
        })
        .finally(() => setIsLoading(false));
    }
  };

  return (
    <button
      onClick={handleLike}
      className={cn(
                'outline-none border-none shadow-none flex group items-center gap-1 md:gap-2.5
cursor-pointer rounded-sm px-2 hover:bg-red-300',
        isLoading ? 'opacity-50 pointer-events-none bg-red-300' : '',
      )}
    >
      <Heart
        fill={isLiked ? '#ab2f33' : 'none'}
        className={isLiked ? 'stroke-[#ab2f33]' : 'group-hover:stroke-black'}
```

```
        size={18}
      />{' '}
      <span className="group-hover:text-black">
        {likesCount} like{likesCount !== 1 && 's'}
      </span>
    </button>
  );
}
```