

```

import { ForwardedRef, RefObject, forwardRef } from 'react';

import { ChevronDown } from 'lucide-react';

import { DropdownProps, EditorType, LanguageType } from '../type';

import useToggle from '@shared/hooks/toggle';

import { useOnClickOutside } from '@shared/hooks/onclick-outside';

import { useEditorContext } from '../../ui/editor.provider';


const Dropdown = forwardRef(function Dropdown(
  { label, options, type }: DropdownProps,
  ref: ForwardedRef<HTMLDivElement> | RefObject<HTMLDivElement>,
) {
  const { isOn, toggle, setOff } = useToggle();

  const { handleChange } = useEditorContext();

  useOnClickOutside(ref as RefObject<HTMLDivElement>, setOff);

  return (
    <div ref={ref as ForwardedRef<HTMLDivElement>} onClick={toggle}>
      <div className="dropdown-title capitalize w-[120px] text-secondary dark:text-foreground border
border-slate-600 dark:border-[#f9f9f914] hover:text-slate-50 transition-all duration-300 ease-in-out">
        {label} <ChevronDown className="text-primary" />
      </div>
      {isOn && (
        <div className="dropdown-menu top-[94px] w-[120px] flex flex-col">
          {options.map((option, i) => {
            const name =
              type === EditorType.language

```

? (option as LanguageType).name

: (option as string);

return (

<div

key={i}

className="hover:bg-destructive py-2 px-2 cursor-pointer"

onClick={() => handleChange?.(type, option)}

>

<button className="capitalize text-left text-secondary dark:text-foreground

hover:text-slate-50 text-sm transition-all duration-300 ease-in-out">

{name}

</button>

</div>

);

}}}

</div>

}}

</div>

);

});

export default Dropdown;