

```
import Modal from '@shared/components/modal';

import { Button } from '@shared/components/ui/Button';

import Input from '@shared/components/ui/Input';

import {
  Select,
  SelectContent,
  SelectItem,
  SelectTrigger,
  SelectValue,
} from '@shared/components/ui/select';

import { languageOptions } from '@shared/constants/explorer';

import LoadingSpinner from '@shared/components/loading-spinner';

import useEditModal from '@shared/hooks/edit-modal';

import { Plus } from 'lucide-react';

import { useAuthContext } from '@shared/components/ui/auth.provider';
```

```
interface EditExplorerModalProps {
  isOpen: boolean;
  onClose: () => void;
  entityType: 'agent' | 'prompt' | 'tool';
  entityId: string;
  onEditSuccessfully: () => void;
}
```

```
function EditExplorerModal({
  isOpen,
```

```
onClose,  
  
entityType,  
  
entityId,  
  
onEditSuccessfully,  
  
}: EditExplorerModalProps) {  
  
  const { user } = useAuthContext();  
  
  const {  
  
    inputState,  
  
    setInputState,  
  
    submit,  
  
    validateMutation,  
  
    debouncedCheckUniqueField,  
  
    addUseCase,  
  
    isPending,  
  
    removeUseCase,  
  
    addRequirement,  
  
    removeRequirement,  
  
  } = useEditModal({ entityId, entityType, onClose, onEditSuccessfully });  
  
  if (!user) return null;  
  
  return (  
  
    <Modal  
  
      className="max-w-2xl"  
  
      isOpen={isOpen}  
  
      onClose={onClose}
```

```
title={` Update ${entityType.charAt(0).toUpperCase() + entityType.slice(1)}`}
```

```
>
```

```
<div className="flex flex-col gap-2 overflow-y-auto no-scrollbar h-[75vh] relative px-4">
```

```
<div className="flex flex-col gap-1">
```

```
<span>Name</span>
```

```
<div className="relative">
```

```
<Input
```

```
  value={inputState.name}
```

```
  onChange={(value) =>
```

```
    setInputState({ ...inputState, name: value })
```

```
  }
```

```
  placeholder="Enter name"
```

```
/>
```

```
</div>
```

```
</div>
```

```
<div className="flex flex-col gap-1">
```

```
<span>Description</span>
```

```
<textarea
```

```
  value={inputState.description}
```

```
  onChange={(e) =>
```

```
    setInputState({ ...inputState, description: e.target.value })
```

```
  }
```

```
  placeholder="Enter description"
```

```
  className="w-full h-20 p-2 border rounded-md bg-transparent outline-0 resize-none"
```

```
/>
```

```
</div>
```

```

<div className="flex flex-col gap-1">

  <span className="capitalize">{entityType}</span>

  <div className="relative">

    <textarea

      value={inputState.uniqueField}

      onChange={(e) => {

        setInputState({ ...inputState, uniqueField: e.target.value });

        debouncedCheckUniqueField(e.target.value);

      }}

      required

      placeholder={`Enter ${entityType} here...`}

      className="w-full h-20 p-2 border rounded-md bg-transparent outline-0 resize-none"

    />

    {validateMutation.isPending ? (

      <div className="absolute right-2 top-2">

        <LoadingSpinner />

      </div>

    ) : (

      <div className="absolute right-2.5 top-2.5">

        {inputState?.uniqueField?.length > 0 &&

          validateMutation.data && (

            <span

              className={

                validateMutation.data.valid

                  ? 'text-green-500'

                  : 'text-red-500'

            }

```

```

    }
  >
    {validateMutation.data.valid ? " : "}
  </span>
  )}
</div>
)}
</div>
{inputState?.uniqueField?.length > 0 &&
  !validateMutation.isPending &&
  validateMutation.data &&
  !validateMutation.data.valid && (
    <span className="text-red-500 text-sm">
      {validateMutation.data.error}
    </span>
  )}
</div>
<div className="flex flex-col gap-1">
  <span>Tags</span>
  <Input
    value={inputState.tags}
    onChange={(value) => setInputState({ ...inputState, tags: value })}
    placeholder="Tools, Search, etc."
  />
</div>
{ /* UseCases */}

```

```
<div className="flex flex-col gap-1">

  <span>Use Cases</span>

  <div className="flex flex-col gap-2">

    {inputState.useCases.map((useCase, index) => (

      <div key={index} className="flex gap-4 items-center">

        <span className="w-8">

          <span># {index + 1}</span>

        </span>

        <div className="w-full flex flex-col gap-1 py-2">

          <Input

            value={useCase.title}

            onChange={(value) =>

              setInputState((prev) => {

                const newUseCases = [...prev.useCases];

                newUseCases[index].title = value;

                return { ...prev, useCases: newUseCases };

              })

            }

            placeholder="Use case title"

          />

          <textarea

            value={useCase.description}

            onChange={(e) =>

              setInputState((prev) => {

                const newUseCases = [...prev.useCases];

                newUseCases[index].description = e.target.value;
```

```

        return { ...prev, useCases: newUseCases };
    })
}

placeholder="Description"

className="w-full h-20 p-2 border rounded-md bg-transparent outline-0 resize-none"

/>

</div>

<div className="w-4">

    {index > 0 && (

        <button

            onClick={() => removeUseCase(index)}

            className="text-red-500"

            >

                </button>

        )}

    </div>

</div>

)}}

<div className="flex justify-center">

    <button onClick={addUseCase} className="text-blue-500">

        <Plus />

    </button>

</div>

</div>

</div>

```

```
{entityType === 'agent' ||
```

```
(entityType === 'tool' && (
```

```
<>
```

```
<div className="flex flex-col gap-1">
```

```
<span>Language</span>
```

```
<Select
```

```
value={inputState.language}
```

```
onValueChange={(value) =>
```

```
  setInputState({ ...inputState, language: value })
```

```
}
```

```
>
```

```
<SelectTrigger className="w-1/2 cursor-pointer capitalize">
```

```
<SelectValue placeholder={inputState.language} />
```

```
</SelectTrigger>
```

```
<SelectContent className="capitalize">
```

```
{languageOptions.map((option) => (
```

```
<SelectItem key={option} value={option}>
```

```
{option}
```

```
</SelectItem>
```

```
)))
```

```
</SelectContent>
```

```
</Select>
```

```
</div>
```

```
<div className="flex flex-col gap-1 mt-2">
```

```
<span>Requirements</span>
```

```
<div className="flex flex-col gap-2">
```



```

{inputState.requirements?.map((requirement, index) => (
  <div key={index} className="flex gap-4 items-center">
    <span className="w-10"> {index + 1}</span>
    <div className="w-full flex flex-col md:flex-row gap-1 py-2">
      <Input
        value={requirement.package}
        onChange={(value) =>
          setInputState((prev) => {
            const newRequirements = [
              ...(prev.requirements ?? []),
            ];
            newRequirements[index].package = value;
            return { ...prev, requirements: newRequirements };
          })
        }
        placeholder="Enter package name"
      />
      <Input
        value={requirement.installation}
        onChange={(value) =>
          setInputState((prev) => {
            const newRequirements = [
              ...(prev.requirements ?? []),
            ];
            newRequirements[index].installation = value;
            return { ...prev, requirements: newRequirements };
          })
        }
      />
    </div>
  </div>
)}

```

```

    })

  }

  placeholder="pip install package"

/>

</div>

<div className="w-4">

  {index > 0 && (

    <button

      onClick={() => removeRequirement(index)}

      className="text-red-500 text-sm"

    >

      </button>

    )}

  </div>

</div>

)}}

<div className="flex justify-center">

  <button

    onClick={addRequirement}

    className="text-blue-500 text-sm"

  >

    <Plus />

  </button>

</div>

</div>

```

```
</div>
```

```
</>
```

```
)))
```

```
</div>
```

```
<div className="flex justify-end gap-2">
```

```
<Button disabled={isPending} onClick={submit}>
```

```
Update
```

```
</Button>
```

```
</div>
```

```
</Modal>
```

```
);
```

```
}
```

```
export default EditExplorerModal;
```