```
import React, { FormEvent, SyntheticEvent, useState } from 'react';
import { Plus, PlusCircle, X } from 'lucide-react';
import {
  Select,
  SelectContent,
  SelectItem,
  SelectTrigger,
  SelectValue,
} from '@/shared/components/ui/select';
import {
  Dialog,
  DialogContent,
  DialogFooter,
  DialogHeader,
  DialogTitle,
  DialogTrigger,
} from '@/shared/components/ui/dialog';
import { Button } from '@/shared/components/ui/Button';
import Input from '@/shared/components/ui/Input';
import { cn } from '@/shared/utils/cn';
import { useOrganizationStore } from '@/shared/stores/organization';
import LoadingSpinner from '@/shared/components/loading-spinner';
import { ROLES } from '@/shared/constants/organization';
import { useInviteModal } from '../../../hooks/invite';
import { UserOrganizationsProps } from '../../../types';
```

```
export default function InviteModal({
  currentOrganization,
}: {
  currentOrganization: UserOrganizationsProps;
}) {
  const {
    email,
    isValidEmail,
    inviteRole,
    openDialog,
    isDisabledInvite,
    setOpenDialog,
    setInviteRole,
    inviteUser,
    handleEmailChange,
    handleOpenModal,
  } = useInviteModal({ currentOrganization });

  const isLoading = useOrganizationStore((state) => state.isLoading);
  const inviteRoles = ROLES.slice(2);

  return (
    <Dialog open={openDialog} onOpenChange={setOpenDialog}>
      <DialogTrigger asChild>
        <Button
          className="gap-0.5"
          variant="secondary"
          onClick={handleOpenModal}
          disabled={isDisabledInvite}
        >
          <Plus size={20} /> Invite
        </Button>
      </DialogTrigger>
      <DialogContent className="sm:max-w-[520px]">
        <DialogHeader>
          <DialogTitle>Invite members by email address</DialogTitle>
        </DialogHeader>
        <form onSubmit={inviteUser}>
          <div className="flex gap-2 items-center mt-3 w-full">
            <p className="w-full">Email</p>
            <p className="w-2/4">Role</p>
          </div>
          <div className="flex items-center gap-2 w-full">
            <div className="w-full">
              <label htmlFor="email" className="text-right hidden">
                Email
              </label>
              <Input
                id="email"
                type="email"
                value={email}
                className={cn(
            'my-2 w-full',
            !isValidEmail && ' border-red-600',
          )}
          placeholder="swarms@example.com"
          onChange={handleEmailChange}
        />
      </div>
      <div className="w-2/4">
        <Select
          onValueChange={(value) => {
            setInviteRole(value);
          }}
          value={inviteRole}
        >
          <SelectTrigger className="w-full">
            <SelectValue placeholder="Reader" />
          </SelectTrigger>
          <SelectContent>
            {inviteRoles?.map((role) => (
              <SelectItem key={role.label} value={role.value}>
                {role.label}
              </SelectItem>
            ))}
          </SelectContent>
        </Select>
      </div>
```

```
          </div>
          <small
            className={cn(
              'text-red-500 text-sm invisible',
              !isValidEmail && 'visible',
            )}
          >
            Please enter a valid email address
          </small>
          <DialogFooter className="sm:justify-between mt-3">
            <Button type="submit" className="w-1/3" disabled={!isValidEmail}>
              {isLoading ? <LoadingSpinner /> : 'Invite'}
            </Button>
          </DialogFooter>
        </form>
      </DialogContent>
    </Dialog>
  );
}
```