```tsx
'use client';

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

import { updatePassword } from '@/shared/utils/auth-helpers/server';

import { handleRequest } from '@/shared/utils/auth-helpers/client';

import { useRouter } from 'next/navigation';

import React, { useState } from 'react';


interface UpdatePasswordProps {

  redirectMethod: string;

}


export default function UpdatePassword({

  redirectMethod,

}: UpdatePasswordProps) {

  const router = redirectMethod === 'client' ? useRouter() : null;

  const [isSubmitting, setIsSubmitting] = useState(false);


  const handleSubmit = async (e: React.FormEvent<HTMLFormElement>) => {

    setIsSubmitting(true); // Disable the button while the request is being handled

    await handleRequest(e, updatePassword, router);

    setIsSubmitting(false);

  };


  return (

    <div className="my-8">
```

```jsx
<form
  noValidate={true}
  className="mb-4"
  onSubmit={(e) => handleSubmit(e)}
>
  <div className="grid gap-2">
    <div className="grid gap-1">
      <label htmlFor="password">New Password</label>
      <input
        id="password"
        placeholder="Password"
        type="password"
        name="password"
        autoComplete="current-password"
        className="w-full p-3 rounded-md bg-zinc-800 text-white"
      />
      <label htmlFor="passwordConfirm">Confirm New Password</label>
      <input
        id="passwordConfirm"
        placeholder="Password"
        type="password"
        name="passwordConfirm"
        autoComplete="current-password"
        className="w-full p-3 rounded-md bg-zinc-800 text-white"
      />
    </div>
```

```jsx
          <Button
            variant="outline"
            type="submit"
            className="mt-1"
            loading={isSubmitting}
          >
            Update Password
          </Button>
        </div>
      </form>
    </div>
  );
}
```