```
import { useState } from 'react'
import { Button } from "/components/ui/button"
import { Input } from "/components/ui/input"
import { Label } from "/components/ui/label"
import { Card, CardHeader, CardTitle, CardContent, CardFooter } from "/components/ui/card"
import { ArrowRight } from "lucide-react"

export default function StockPredictor() {
  const [symbol, setSymbol] = useState('')
  const [prediction, setPrediction] = useState<number | null>(null)
  const [isLoading, setIsLoading] = useState(false)
  const [error, setError] = useState('')

  const handlePredict = () => {
    if (!symbol.trim()) {
      setError('Please enter a stock symbol')
      return
    }

    setIsLoading(true)
    setError('')
    setPrediction(null)

    // Simulate API call with timeout
    setTimeout(() => {
      try {
        // Mock prediction logic - in a real app this would call an API
        const mockPrediction = Math.random() * 1000 + 50
        setPrediction(parseFloat(mockPrediction.toFixed(2)))
      } catch (err) {
        setError('Failed to get prediction. Please try again.')
      } finally {
        setIsLoading(false)
      }
    }, 1500)
  }

  const handleKeyDown = (e: React.KeyboardEvent) => {
    if (e.key === 'Enter') {
      handlePredict()
    }
  }

  return (
    <div className="min-h-screen flex items-center justify-center p-4">
      <Card className="w-full max-w-md">
        <CardHeader>
          <CardTitle className="text-2xl font-bold">Stock Price Predictor</CardTitle
>
        </CardHeader>
        <CardContent className="space-y-4">
          <div className="space-y-2">
            <Label htmlFor="symbol">Stock Symbol</Label>
            <Input
              id="symbol"
              placeholder="e.g. AAPL, MSFT"
              value={symbol}
              onChange={(e) => setSymbol(e.target.value.toUpperCase())}
              onKeyDown={handleKeyDown}
              disabled={isLoading}
            />
            {error && <p className="text-sm text-red-500">{error}</p>}
          </div>

          {prediction !== null && (
            <div className="p-4 bg-blue-50 rounded-lg">
              <p className="text-sm text-blue-800">Predicted price for {symbol}:</p>
              <p className="text-3xl font-bold text-blue-900">${prediction}</p>
              <p className="text-xs text-blue-600 mt-1">
                Prediction as of {new Date().toLocaleDateString()}
              </p>
            </div>
          )}
        </CardContent>
        <CardFooter>
          <Button
            onClick={handlePredict}
            disabled={isLoading}
            className="w-full"
          >
            {isLoading ? (
              'Predicting...'
            ) : (
              <>
                Predict Price <ArrowRight className="ml-2 h-4 w-4" />
              </>
            )}
          </Button>
        </CardFooter>
      </Card>
    </div>
  )
}
```