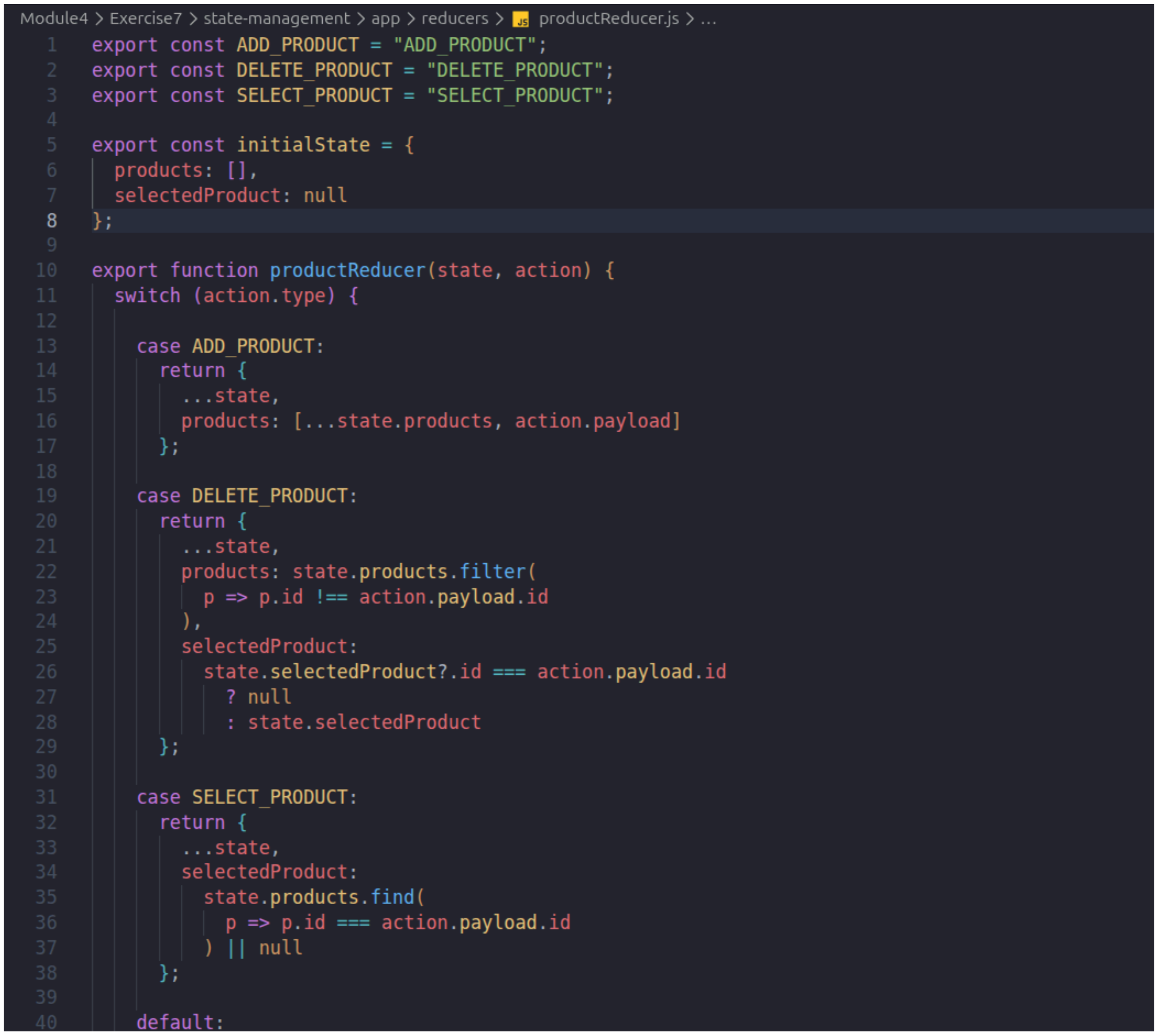
Module 4

Exercise 7 (NextJs State Management)

Q1.) Implement useReducer hook to GET, ADD, DELETE products.

A1.) NextJS

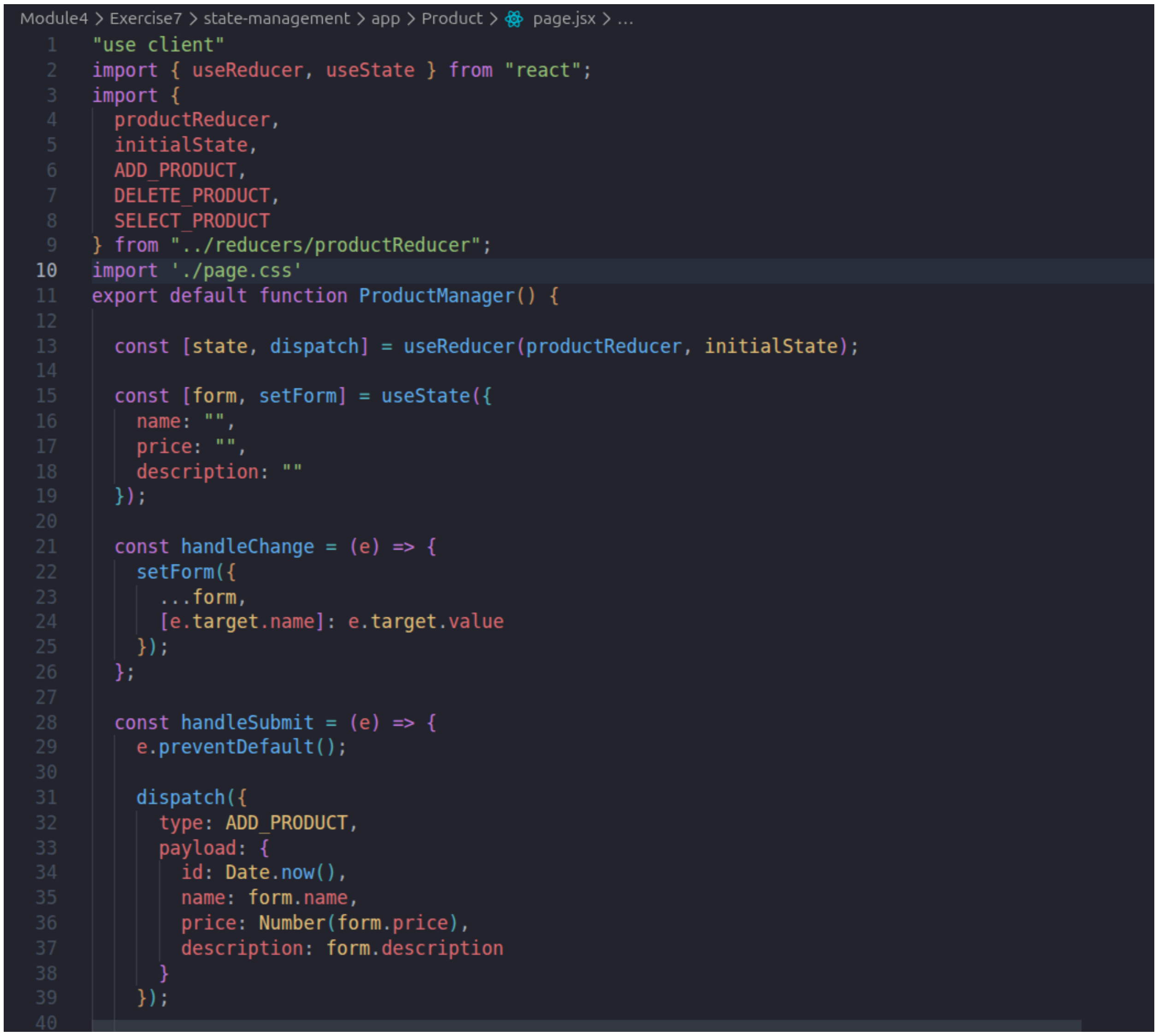
Reducer



Module4 > Exercise7 > state-management > app > reducers > productReducer.js > ...

```javascript
export const ADD_PRODUCT = "ADD_PRODUCT";
export const DELETE_PRODUCT = "DELETE_PRODUCT";
export const SELECT_PRODUCT = "SELECT_PRODUCT";

export const initialState = {
  products: [],
  selectedProduct: null
};

export function productReducer(state, action) {
  switch (action.type) {

    case ADD_PRODUCT:
      return {
        ...state,
        products: [...state.products, action.payload]
      };

    case DELETE_PRODUCT:
      return {
        ...state,
        products: state.products.filter(
          p => p.id !== action.payload.id
        ),
        selectedProduct:
          state.selectedProduct?.id === action.payload.id
            ? null
            : state.selectedProduct
      };

    case SELECT_PRODUCT:
      return {
        ...state,
        selectedProduct:
          state.products.find(
            p => p.id === action.payload.id
          ) || null
      };

    default:
```

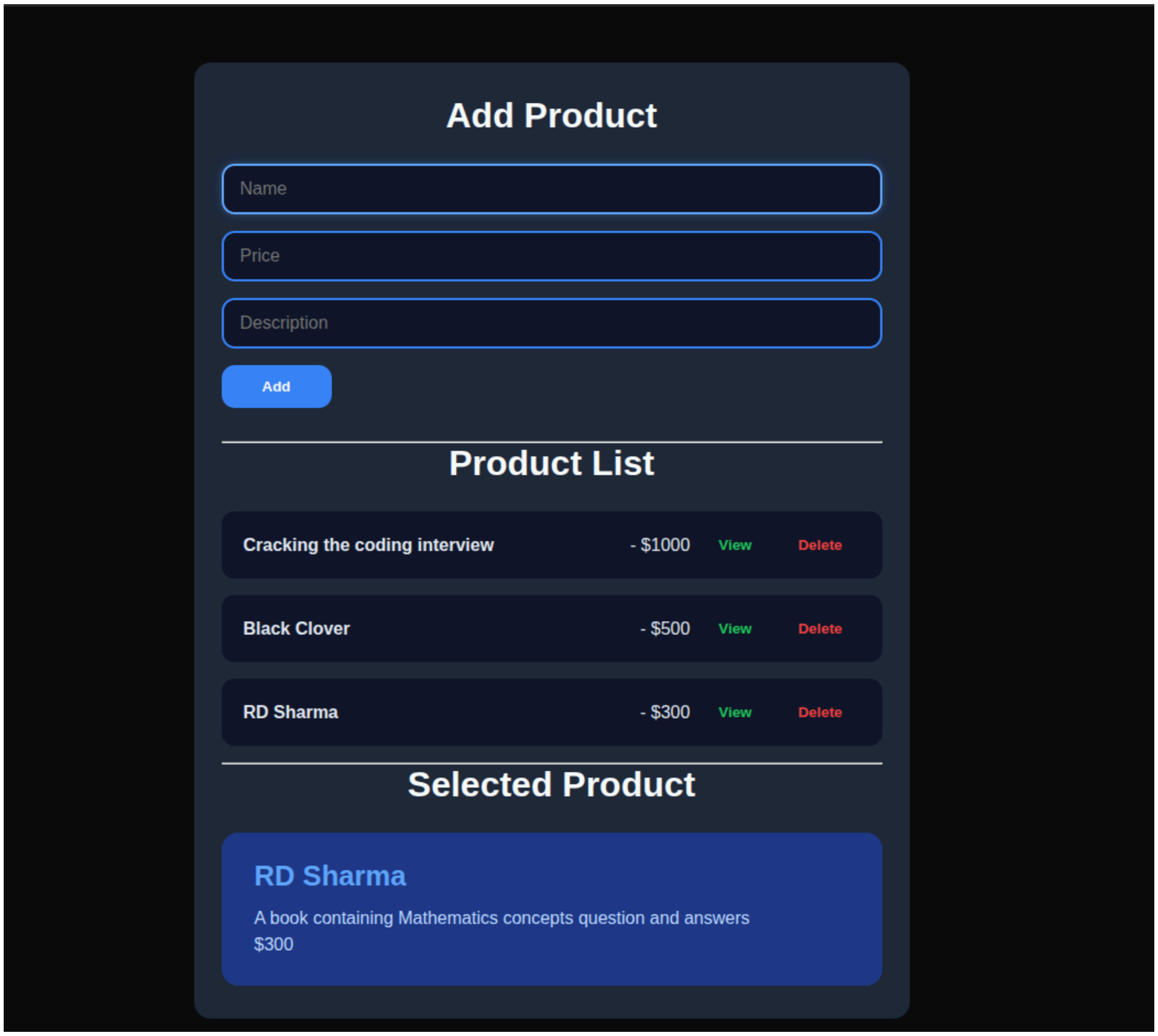
Implementation



Module4 > Exercise7 > state-management > app > Product > page.jsx > ...

```jsx
"use client"
import { useReducer, useState } from "react";
import {
  productReducer,
  initialState,
  ADD_PRODUCT,
  DELETE_PRODUCT,
  SELECT_PRODUCT
} from "../reducers/productReducer";
import './page.css'
export default function ProductManager() {

  const [state, dispatch] = useReducer(productReducer, initialState);

  const [form, setForm] = useState({
    name: "",
    price: "",
    description: ""
  });

  const handleChange = (e) => {
    setForm({
      ...form,
      [e.target.name]: e.target.value
    });
  };

  const handleSubmit = (e) => {
    e.preventDefault();

    dispatch({
      type: ADD_PRODUCT,
      payload: {
        id: Date.now(),
        name: form.name,
        price: Number(form.price),
        description: form.description
      }
    });

```

Output



Add Product

Name

Price

Description

Add

Product List

Cracking the coding interview - $1000 View Delete

Black Clover - $500 View Delete

RD Sharma - $300 View Delete

Selected Product

RD Sharma

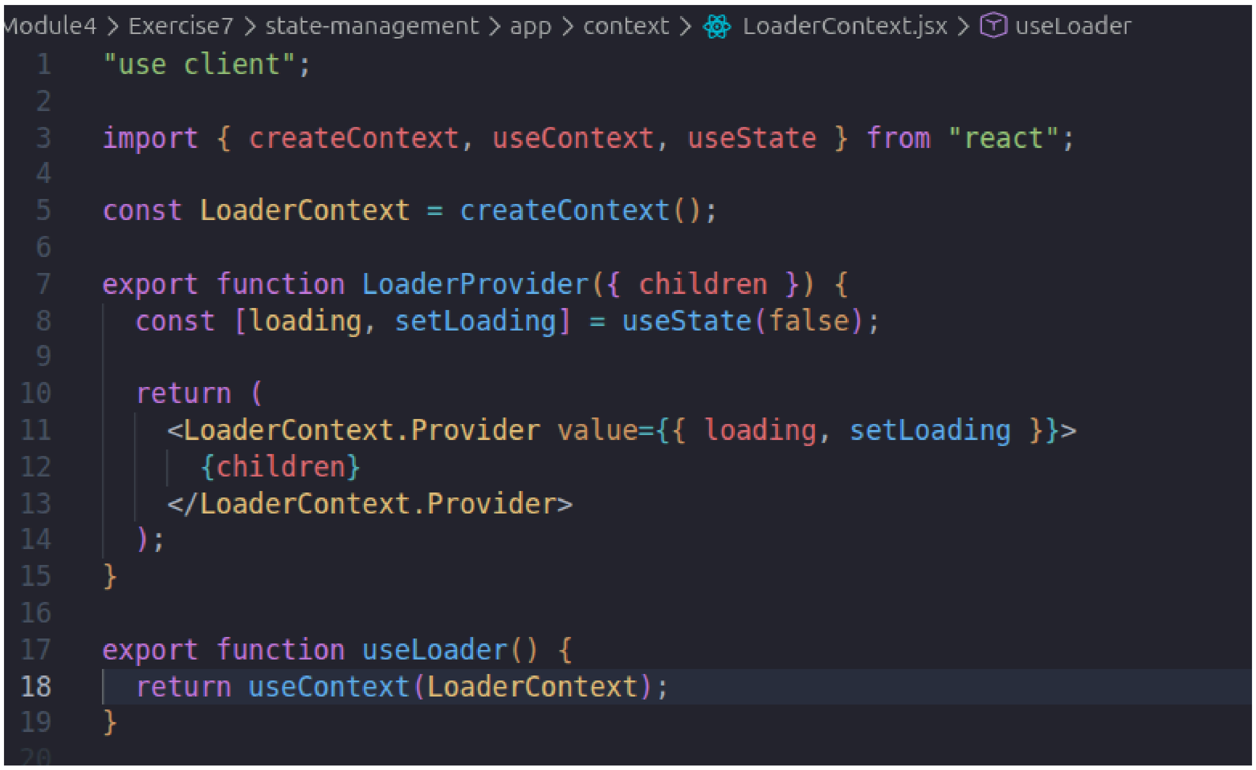
A book containing Mathematics concepts question and answers

$300

Q2.) Implement useContext hook to implement a loader in a page while api is in loading state.

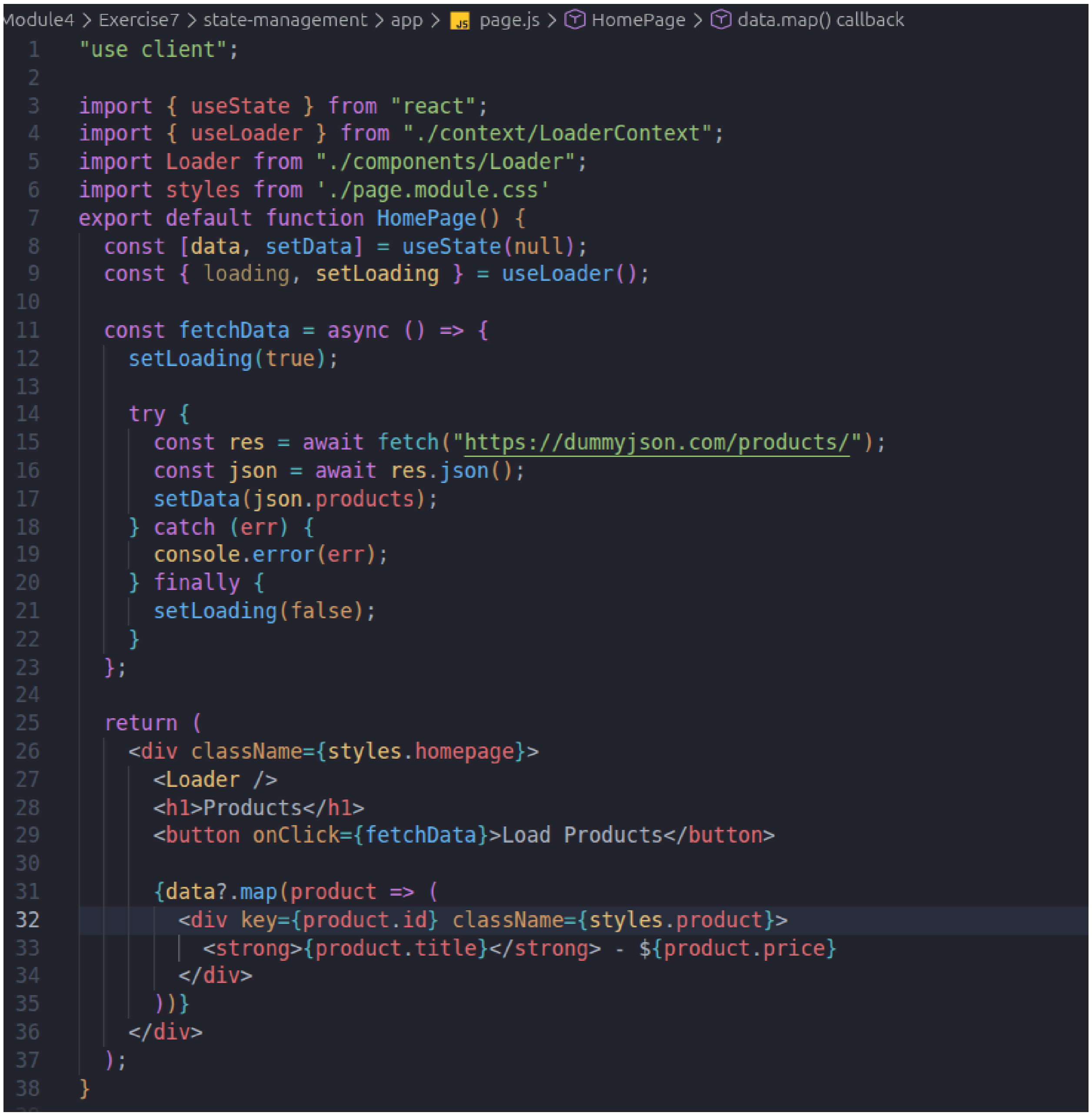
A2.) NextJs

Context


Module4 > Exercise7 > state-management > app > context > LoaderContext.jsx > useLoader

```jsx
"use client";

import { createContext, useContext, useState } from "react";

const LoaderContext = createContext();

export function LoaderProvider({ children }) {
  const [loading, setLoading] = useState(false);

  return (
    <LoaderContext.Provider value={{ loading, setLoading }}>
      {children}
    </LoaderContext.Provider>
  );
}

export function useLoader() {
  return useContext(LoaderContext);
}
```

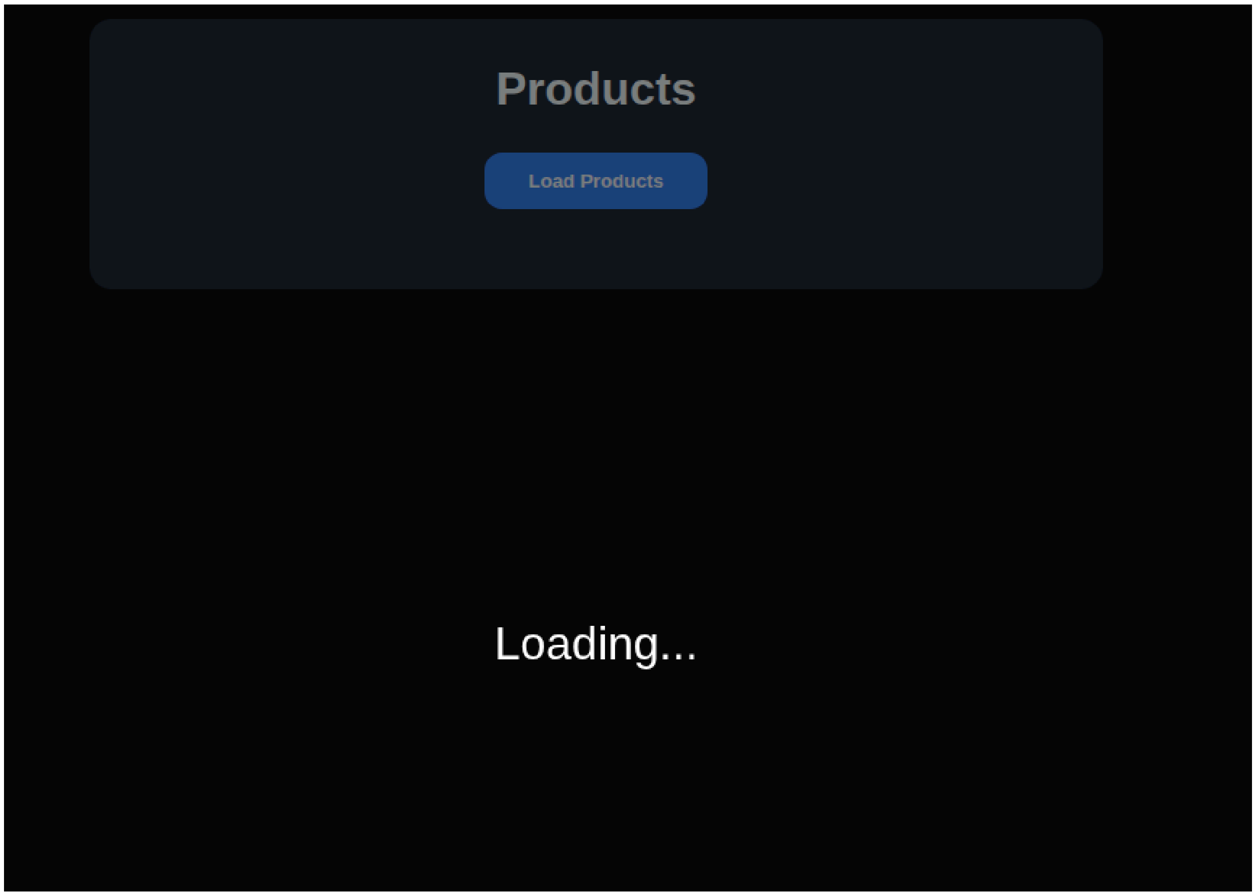
Page



odule4 > Exercise7 > state-management > app > page.js > HomePage > data.map() callback

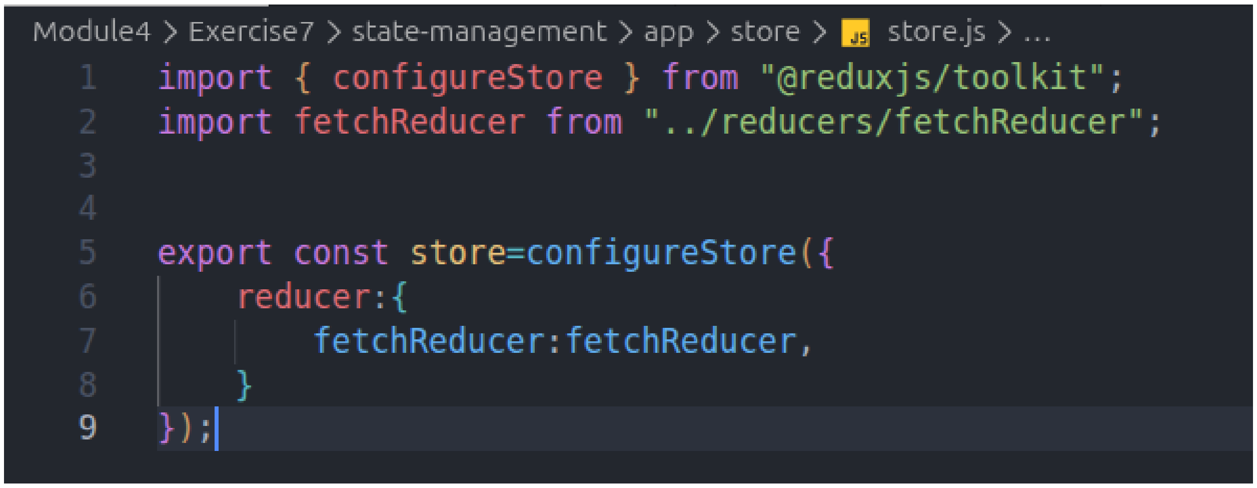
```
"use client";

import { useState } from "react";
import { useLoader } from "./context/LoaderContext";
import Loader from "./components/Loader";
import styles from './page.module.css'
export default function HomePage() {
  const [data, setData] = useState(null);
  const { loading, setLoading } = useLoader();

  const fetchData = async () => {
    setLoading(true);

    try {
      const res = await fetch("https://dummyjson.com/products/");
      const json = await res.json();
      setData(json.products);
    } catch (err) {
      console.error(err);
    } finally {
      setLoading(false);
    }
  };

  return (
    <div className={styles.homepage}>
      <Loader />
      <h1>Products</h1>
      <button onClick={fetchData}>Load Products</button>

      {data?.map(product => (
        <div key={product.id} className={styles.product}>
          <strong>{product.title}</strong> - ${product.price}
        </div>
      ))}
    </div>
  );
}
```

Output



Q3.) Implement Redux to fetch api 'https://dummyjson.com/quotes'.
A3.) NextJs

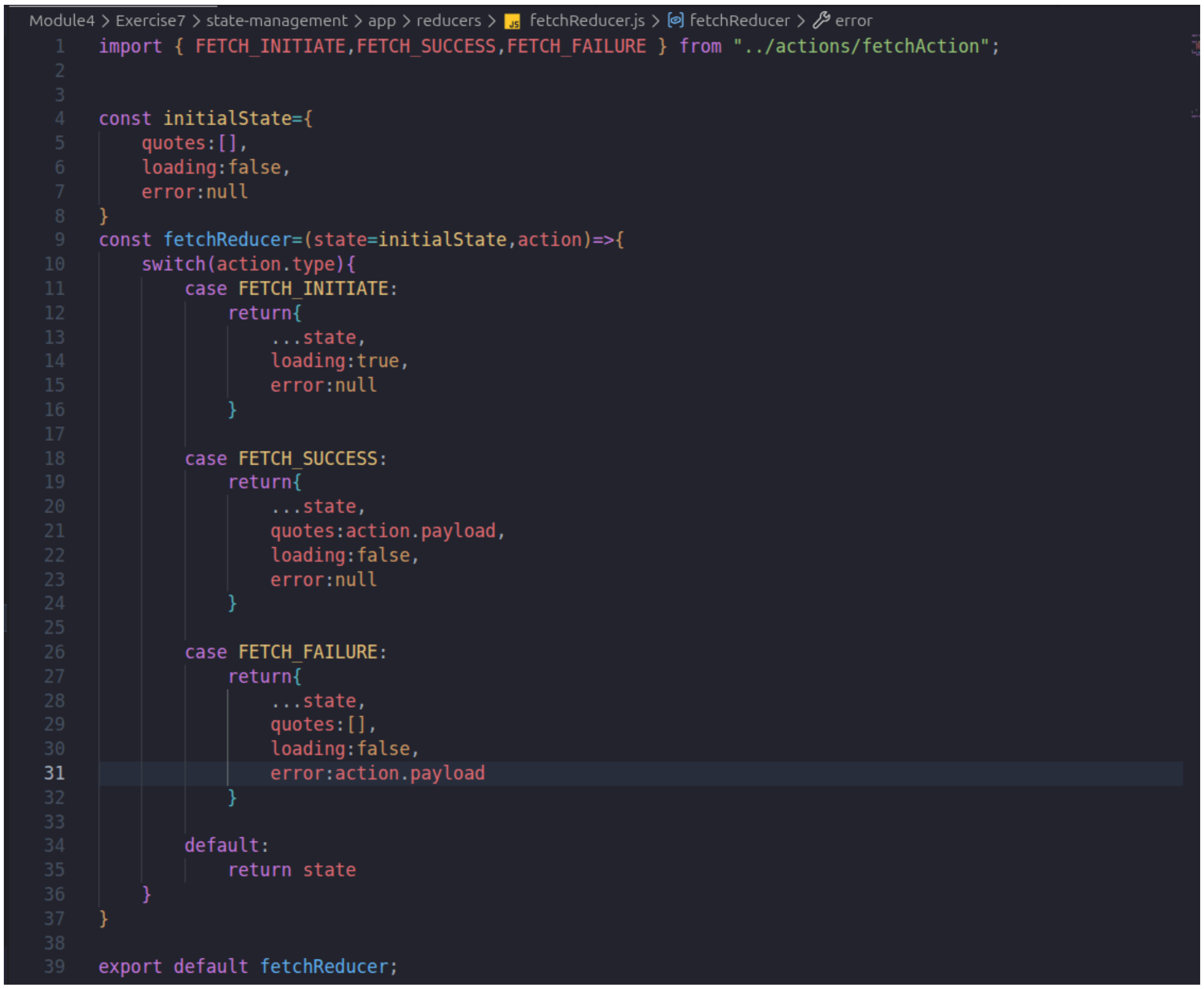
store.js



Module4 > Exercise7 > state-management > app > store > store.js > ...

```js
import { configureStore } from "@reduxjs/toolkit";
import fetchReducer from "../reducers/fetchReducer";


export const store=configureStore({
    reducer:{
        fetchReducer:fetchReducer,
    }
});
```

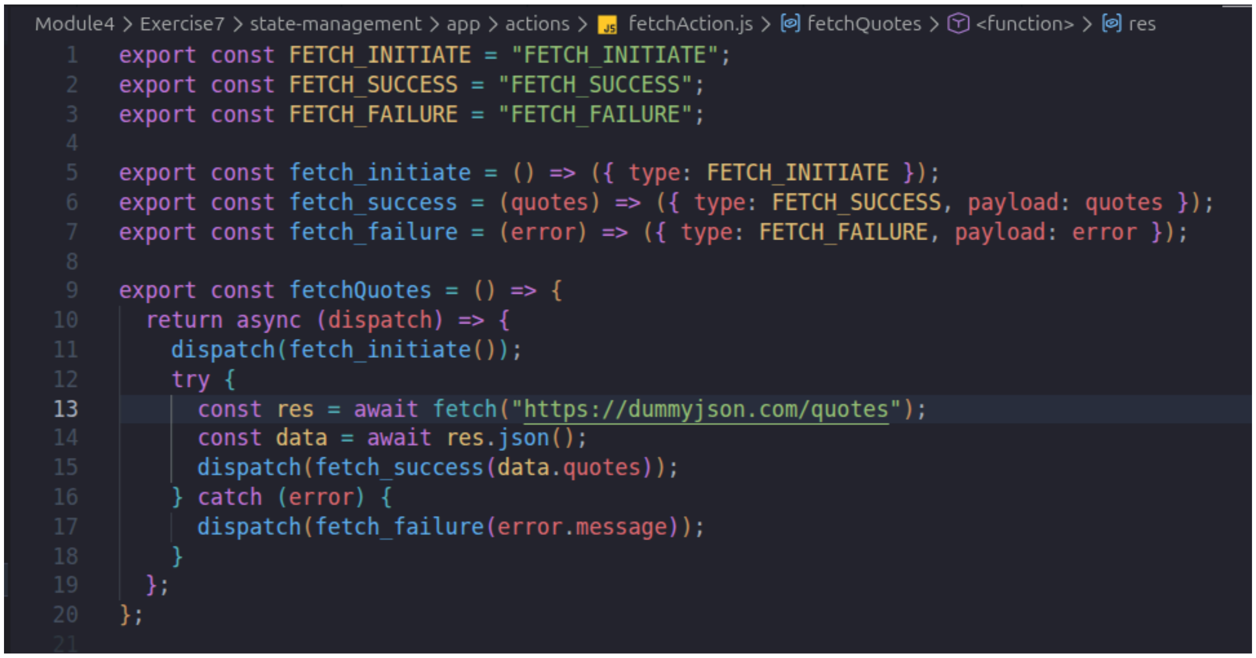
fetchreducer.js



Module4 > Exercise7 > state-management > app > reducers > fetchReducer.js > fetchReducer > error

```js
import { FETCH_INITIATE,FETCH_SUCCESS,FETCH_FAILURE } from "../actions/fetchAction";


const initialState={
    quotes:[],
    loading:false,
    error:null
}
const fetchReducer=(state=initialState,action)=>{
    switch(action.type){
        case FETCH_INITIATE:
            return{
                ...state,
                loading:true,
                error:null
            }

        case FETCH_SUCCESS:
            return{
                ...state,
                quotes:action.payload,
                loading:false,
                error:null
            }

        case FETCH_FAILURE:
            return{
                ...state,
                quotes:[],
                loading:false,
                error:action.payload
            }

        default:
            return state
    }
}

export default fetchReducer;
```

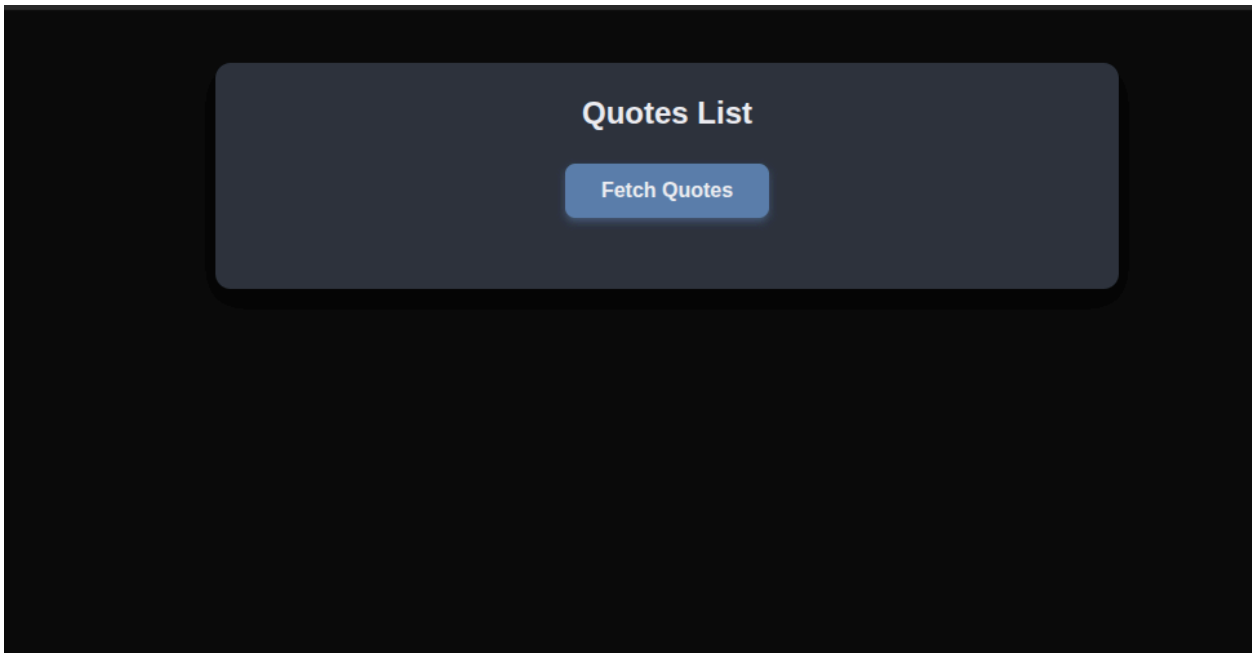
[fetchaction.js]



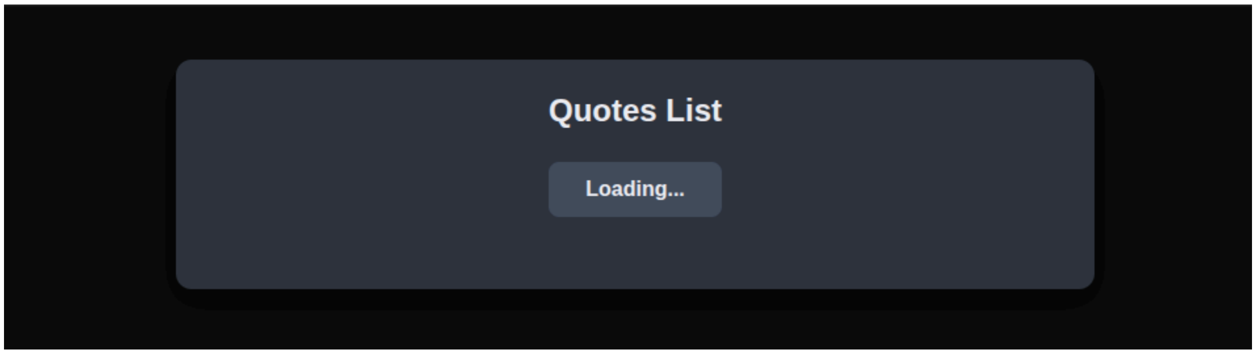
Module4 > Exercise7 > state-management > app > actions > fetchAction.js > fetchQuotes > <function> > res

```javascript
export const FETCH_INITIATE = "FETCH_INITIATE";
export const FETCH_SUCCESS = "FETCH_SUCCESS";
export const FETCH_FAILURE = "FETCH_FAILURE";

export const fetch_initiate = () => ({ type: FETCH_INITIATE });
export const fetch_success = (quotes) => ({ type: FETCH_SUCCESS, payload: quotes });
export const fetch_failure = (error) => ({ type: FETCH_FAILURE, payload: error });

export const fetchQuotes = () => {
  return async (dispatch) => {
    dispatch(fetch_initiate());
    try {
      const res = await fetch("https://dummyjson.com/quotes");
      const data = await res.json();
      dispatch(fetch_success(data.quotes));
    } catch (error) {
      dispatch(fetch_failure(error.message));
    }
  };
};
```

Output



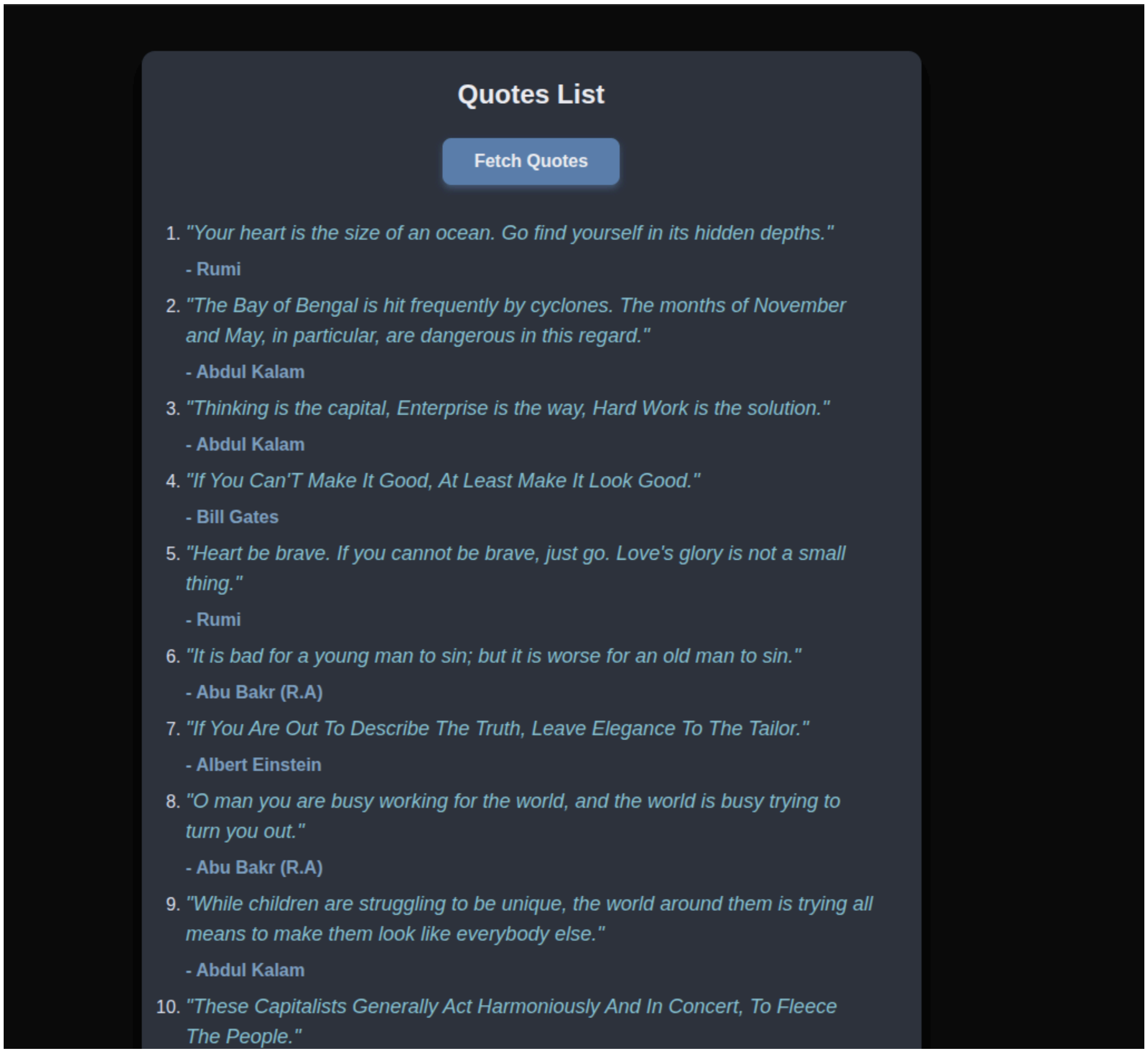
Loading



Quotes List

Loading...

Fetch

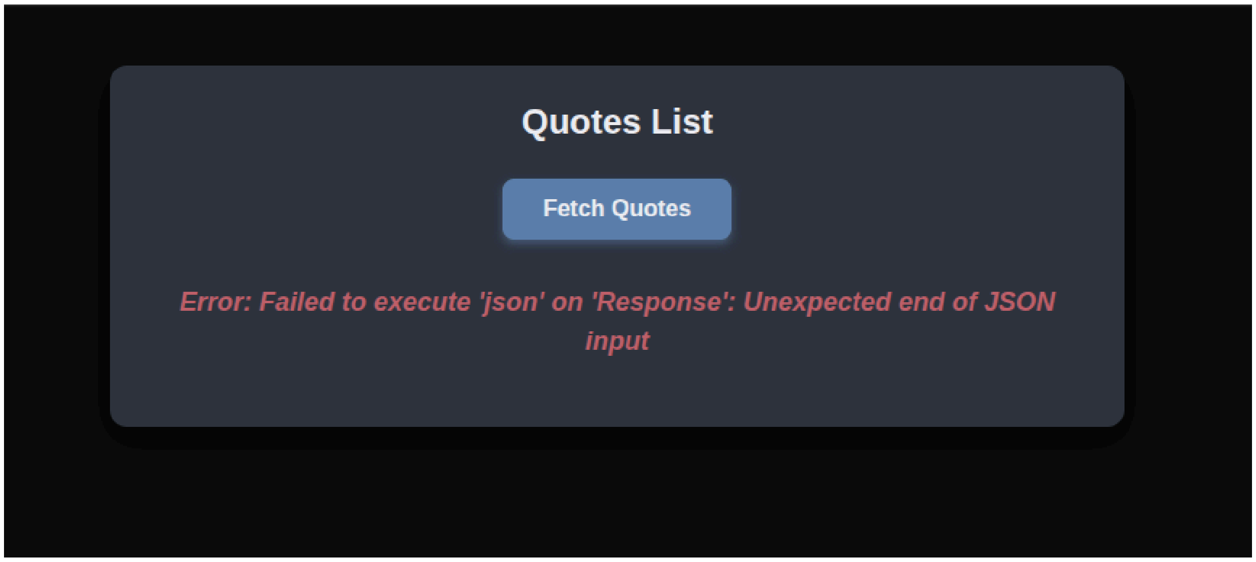


Quotes List

Fetch Quotes

1. *"Your heart is the size of an ocean. Go find yourself in its hidden depths."*
   - Rumi
2. *"The Bay of Bengal is hit frequently by cyclones. The months of November and May, in particular, are dangerous in this regard."*
   - Abdul Kalam
3. *"Thinking is the capital, Enterprise is the way, Hard Work is the solution."*
   - Abdul Kalam
4. *"If You Can'T Make It Good, At Least Make It Look Good."*
   - Bill Gates
5. *"Heart be brave. If you cannot be brave, just go. Love's glory is not a small thing."*
   - Rumi
6. *"It is bad for a young man to sin; but it is worse for an old man to sin."*
   - Abu Bakr (R.A)
7. *"If You Are Out To Describe The Truth, Leave Elegance To The Tailor."*
   - Albert Einstein
8. *"O man you are busy working for the world, and the world is busy trying to turn you out."*
   - Abu Bakr (R.A)
9. *"While children are struggling to be unique, the world around them is trying all means to make them look like everybody else."*
   - Abdul Kalam
10. *"These Capitalists Generally Act Harmoniously And In Concert, To Fleece The People."*

Error



Error: Failed to execute 'json' on 'Response': Unexpected end of JSON input