Concepts in Focus

1. Context

Context can also be used:

- When data has to be available between different **Routes**.
- When data has to be available for more number of **Components**

2. Using Context in methods

- We can add **consumer** or **provider** in the class method when it returns **JSX**.
- In the below code snippet, we have used **Consumer** Component in `renderProductDetailsView` because this method is returning **JSX**.

Example:

File: src/components/ProductItemDetails

JSX

```
import { Component } from "react";
import Cookies from "js-cookie";
import Loader from "react-loader-spinner";
import { BsPlusSquare, BsDashSquare } from '

import CartContext from "../../context/Cart(

import Header from "../Header";
import SimilarProductItem from "../SimilarPr

import "./index.css";

const apiStatusConstants = {
  initial: "INITIAL",
  success: "SUCCESS",
  failure: "FAILURE",
  inProgress: "IN_PROGRESS",
};

class ProductItemDetails extends Component
  state = {
    productData: {},
    similarProductsData: [],
    apiStatus: apiStatusConstants.initial,
    quantity: 1,
  };

  componentDidMount() {
    this.getProductData();
  }

  getFormattedData = (data) => ({
    availability: data.availability,
    brand: data.brand,
    description: data.description,
    id: data.id,
    imageUrl: data.image_url,
    price: data.price,
    rating: data.rating,
    title: data.title,
    totalReviews: data.total_reviews,
  });

  getProductData = async () => {
    const { match } = this.props;
    const { params } = match;
    const { id } = params;

    this.setState({
      apiStatus: apiStatusConstants.inProgre
    });
    const jwtToken = Cookies.get("jwt_token'
    const apiUrl = `https://apis.ccbp.in/pr
    const options = {
      headers: {
        Authorization: `Bearer ${jwtToken}`,
      },
      method: "GET",
    };
    const response = await fetch(apiUrl, op
    if (response.ok) {
      const fetchedData = await response.js
      const updatedData = this.getFormattedI
      const updatedSimilarProductsData = fe
        (eachSimilarProduct) => this.getForr
      );
      this.setState({
        productData: updatedData,
        similarProductsData: updatedSimilarF
        apiStatus: apiStatusConstants.succes
      });
    }
    if (response.status === 404) {
      this.setState({
        apiStatus: apiStatusConstants.failur
      });
    }
  };

  renderLoadingView = () => (
    <div className="products-details-loader-
      <Loader type="ThreeDots" color="#0b69
    </div>
  );

  renderFailureView = () => (
    <div className="product-details-error-vi
      <img
        alt="error view"
        src="https://assets.ccbp.in/frontenc
        className="error-view-image"
      />
      <h1 className="product-not-found-headi
      <button type="button" className="butto
        Continue Shopping
      </button>
    </div>
  );

  onDecrementQuantity = () => {
    const { quantity } = this.state;
    if (quantity > 1) {
      this.setState((prevState) => ({ quant
    }
  };

  onIncrementQuantity = () => {
    this.setState((prevState) => ({ quantit
  };

  renderProductDetailsView = () => (
    <CartContext.Consumer>
      {(value) => {
        const { productData, quantity, simi
        const {
          availability,
          brand,
          description,
          imageUrl,
          price,
          rating,
          title,
          totalReviews,
        } = productData;
        const { addCartItem } = value;
        const onClickAddToCart = () => {
          addCartItem({ ...productData, qua
        };

        return (
          <div className="product-details-s
            <div className="product-details-
              <img src={imageUrl} alt="prod
              <div className="product">
                <h1 className="product-name'
                <p className="price-details'
                <div className="rating-and-
                  <div className="rating-co
                    <p className="rating">{
                    <img
                      src="https://assets.c
                      alt="star"
                      className="star"
                    />
                  </div>
                  <p className="reviews-cou
                </div>
                <p className="product-descr
                <div className="label-value-
                  <p className="label">Avai
                  <p className="value">{ava
                </div>
                <div className="label-value-
                  <p className="label">Bran
                  <p className="value">{bra
                </div>
                <hr className="horizontal-li
                <div className="quantity-con
                  <button
                    type="button"
                    className="quantity-cont
                    onClick={this.onDecremen
                  >
                    <BsDashSquare className=
                  </button>
                  <p className="quantity">{
                  <button
                    type="button"
                    className="quantity-cont
                    onClick={this.onIncremen
                  >
                    <BsPlusSquare className=
                  </button>
                </div>
                <button
                  type="button"
                  className="button add-to-c
                  onClick={onClickAddToCart}
                >
                  ADD TO CART
                </button>
              </div>
            </div>
            <h1 className="similar-products-
            <ul className="similar-products-
              {similarProductsData.map((each
                <SimilarProductItem
                  productDetails={eachSimila
                  key={eachSimilarProduct.id
                />
              ))}
            </ul>
          </div>
        );
      }}
    </CartContext.Consumer>
  );

  renderProductDetails = () => {
    const { apiStatus } = this.state;

    switch (apiStatus) {
      case apiStatusConstants.success:
        return this.renderProductDetailsView
      case apiStatusConstants.failure:
        return this.renderFailureView();
      case apiStatusConstants.inProgress:
        return this.renderLoadingView();
      default:
        return null;
    }
  };

  render() {
    return (
      <>
        <Header />
        <div className="product-item-detail
          {this.renderProductDetails()}
        </div>
      </>
    );
  }
}

export default ProductItemDetails;
```

Collapse

3. Final code

Run the below command in your IDE to download the final code.

```
ccbp start RJSIVIJ1YR
```