Amazona Ecomerce app -🡪Details

1. Instalation of the react app .
2. Home Screen

* List products 🡪 inside frontend folder 🡪in src🡪 create data.js and inside that create a object name data which contains details of the products.

Then in the App.js file in the main section we are importing the data.js file and using map function traverse through the object and display …

Apply css to that products name ..and used <a> tag in images and names.

* Add Routing 🡪

First need to install the react router dom (npm i react-router-dom)

Then import BrowseRouter in the App.js page..

Then create a Screens folder 🡪 then HomeScreen.js🡪

Inside that move all the content of the Products that was previously present in app.js then using route we can get the deltails of the HomeScreen.js file.

Then we added ProductScreen in App.js and inside ProductScreen we use UseParams HOOK to get the ProductSlug value using params hook in react router-dom.

To resolve the page refress issue when moving from one page to another we have to use the Link tag of react instead of using the <a> tag.

Change the <a> tag in present in the HomeScreen to <Link> tag of react and href 🡪to

* Create NodeJS Server 🡪

Now we are creating a new folder Backend to perform the backend operations

Install the NodeJS in it (npm init) and then update the package.json file {type : “module”}

Now, we have to create server.js in backend folder..Then install express dependencies..

Then we have to copy the data.js file from the frontend folder to the backend folder

And install nodemon to auto compile and load the file.

* Fetch Products From Backend 🡪

1. Set proxy in package.json – in the frontend folder..

This proxy is done to get access of the backend in the frontend

1. npm install axios : To get data from backend we ar using axios libraby… to know more about this axios visit([axios-http.com](https://axios-http.com/docs/intro)).

Install the axios in the frontend folder

1. use state hook
2. use effect hook
3. use reducer hook

* **Manage State By reducer HOOK**

Here we are replacing useReducer in state of useState bcz when we want to send an ajex request to backend state is complex and nextState depends on the previous one

Link for useState ,useReducers -🡪 <https://reactjs.org/docs/hooks-reference.html>

🡪action.payload contains all produts from backend

🡪To define useReducer we need to define an array that contains two values the first one is object and the second parameters are dispatch

* In package.json we import (logger) use-reducer-logger
* In HomeScreen replace useState with useReducers

1. React Bootstrap

* **ADD BOOTSTRAP**

Install react-bootstrap bootstrap

In app.js Converting Simple header main and footer to reactBootstrap

Navbar Container, LinkContainer,

* In package.json we install bootstrap react-bootstrap ans react-router-bootstrap
* In app.js we change the simple header into react-bootstrap header main and footer
* In index.css we remove the header design and give the class name of div used in app.js site-container
* In index.js we import bootstrap css from bootstrap components
* Create Product Component

Using react BootStarap desing the products in <Row> and <Col> tags …

1. We are creating a components folder and inside that we are creating product.js file where we are using the code of the products details…

This is Done usings props which helps in reusability of code

In the product.js file further we desing the code using react -Bootstarap using different tags like 🡪 Card, card.Body,Card.Title,Card.Text,Button

1. Then in the components folder we created rating.js file and done the rating and numreviews operatons in that . we import the fontowsum font in the index.html page inside the public folder ans done Some changes in the index.css folder .

Create Product and Rating Components..

* At first go to the HomeScreen and to change the product list from simple div to Bootstrap styles 🡪changes ar using tags(Row , Col,)
* Use of props ---- to use product items in multiple places we create a product component …Which is placed inside components folder and in the Prodcuct.js file . then import the details in the HomeScreen
* Now in the Product.js file we are changing the div into the Card tags card.Body , Card.Text, Card.Title all are from react-Bootstrap

Changes Done --Simple div to Card components of react-bootstrap which is also responsive

🡪Resolve the error of 🡪 each child in a list should get a unique ‘key’

Error resolve is done 🡪 go to the HomeScreen and after the map function inside the Col add the key Attribute { key ={product.slug}}

* In the HomeScreen.js HE

🡪Rating Components -----

In Product.js file add Rating components .

Inside components folder add The Rating.js file inside which keep the details of the rating and numReviews..

Then go to index.html in the public folder to import fontawesome

1. Product Screen

* Add Product Screen---

1 . Create Prodcut Details Screen – Open the ProductScreen.js inside the screens folder. Here we have to fetch the product from backend

2. Open server.js and then change app.get('/api/products/slug/:slug', (req, res) => {

  const product = data.products.find((x) => x.slug === req.params.slug);

  if (product) {

    res.send(product);

  } else {

    res.status(404).send({ message: 'Product Not Found' });

  }

});

In the ProductScreen we have use many react-Bootstrap tags like Row,Col,ListGroup,ListGroup.Item

Checking countInStock value we are also checking the avaivility of the product .

* To show the product name in the page title We ar using a package name

React Helmet async.. Inside the frontend folder install (npm I react-helmet-async) and then go to index.js and then wrap the whole App inside HelmetProvider . and then go to productScreen and before heading

 <Helmet>

                <title>{product.name}</title>

              </Helmet>

Also do the change in the HomeScreen.js

* **Create-Loading-and-Message-Component**
* In the HomeScreen.js replace the loading div to a component name LoadingBox

Inside the component folder create a LoadingBox.js.

By this we are creating a universal loading and use it in every part of the project

In the LoadingBox page we used a Spinner icon from react-bootstrap

Same create a MessageBox component in the component folder

And inside messageBox use Alert from react-Bootstrap

* Create utils.js fle to resolve the error message of product not found

And then go to the productScreen and go to the try catch and as a payload return getError(err).

1. Cart Screen
2. Create React Context.

1. Complete Add To cart Functionality.
2. Create Cart Screen
3. Complete Cart Screen.