Amazon E-Com Website - Full Code Report

Website Links - https://www.amazon.com/

This document contains the complete source code for the amazon-e-com-website project, organized by file location.

Author: ADITYA KAILAS AHIRRAO

Project File Structure

```
amazon-e-com-website/
├── .gitignore
├── package.json
├── README.md
├── public/
│   └── index.html
└── src/
    ├── App.css
    ├── App.js
    ├── index.js
    ├── components/
    │   ├── Header.js
    │   ├── ProductCard.js
    │   └── CartItem.js
    ├── context/
    │   └── CartContext.js
    ├── data/
    │   └── products.js
    └── pages/
        ├── HomePage.js
        ├── ProductListPage.js
        ├── ProductDetailPage.js
        └── CartPage.js
```

1. Root Files

These files are located in the main amazon-e-com-website/ directory.

package.json

This file defines the project, its dependencies, and its scripts.

```
{
  "name": "amazon-e-com-website",
  "version": "1.0.0",
  "description": "A dynamic Amazon-themed e-commerce site built with React by [Your Name].",
  "author": "[Your Name]",
  "private": true,
  "dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-router-dom": "^6.15.0",
    "react-scripts": "5.0.1"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

README.md

This is the main documentation file for your GitHub repository.

```
# Amazon E-Com Website

This is a dynamic e-commerce website clone, styled to look like Amazon, built by **[Your Name]**.

This project was built with React.js and demonstrates a complete, multi-page application.

## Tech Stack

* **Frontend:** React.js
* **Routing:** React Router v6
* **State Management:** React Context API (for Shopping Basket)

## Features

* Amazon-themed styling and layout.
* Browse a grid of products.
* View individual product details.
* A dynamic, working "Shopping Basket".
* Add and remove items from the basket.

## How to Run This Project Locally

1.  Clone the repository:
    `git clone https://github.com/[Your-Username]/amazon-e-com-website.git`
2.  Navigate into the project directory:
    `cd amazon-e-com-website`
3.  Install all required dependencies:
    `npm install`
4.  Start the development server:
    `npm start`

The application will be running at `http://localhost:3000`.
```

.gitignore

This file tells Git which files (like node_modules) to ignore.

```
# See [https://help.github.com/articles/ignoring-files/](https://help.github.com/articles/ignoring-files/) for more about ignoring files.

# dependencies
/node_modules
/.pnp
.pnp.js

# testing
/coverage

# production
/build

# misc
.DS_Store
.env.local
.env.development.local
.env.test.local
.env.production.local

npm-debug.log*
yarn-debug.log*
yarn-error.log*
```

2. Public Folder

This folder contains the main HTML file that your React app is injected into.

public/index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Amazon E-Com Website created by [Your Name]"
    />
    <title>Amazon E-Com Website</title>
  </head>
  <body>
    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>
```

3. Source Files (src/)

This directory contains all of your React components, pages, styles, and logic.

src/index.js

This is the main entry point for the React application.

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import { BrowserRouter } from 'react-router-dom';
import App from './App';
import { CartProvider } from './context/CartContext';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    {/* BrowserRouter handles the app's routing */}
    <BrowserRouter>
      {/* CartProvider makes the cart state available to all components */}
      <CartProvider>
        <App />
      </CartProvider>
    </BrowserRouter>
  </React.StrictMode>
);
```

src/App.js

This is the top-level component that defines the page layout and routes.

```
import React from 'react';
import { Routes, Route } from 'react-router-dom';
import Header from './components/Header';
import HomePage from './pages/HomePage';
import ProductListPage from './pages/ProductListPage';
import ProductDetailPage from './pages/ProductDetailPage';
import CartPage from './pages/CartPage';
import './App.css'; // Import global styles

function App() {
  return (
    <div className="App">
      {/* Header is on every page */}
      <Header />

      {/* Main content area */}
      <main className="amz-main-container">
        {/* Routes define which page component to show based on the URL */}
        <Routes>
          <Route path="/" element={<HomePage />} />
          <Route path="/products" element={<ProductListPage />} />
          <Route path="/products/:id" element={<ProductDetailPage />} />
          <Route path="/cart" element={<CartPage />} />
        </Routes>
      </main>

      {/* A simple footer for every page */}
      <footer className="amz-footer">
        Back to top
      </footer>
    </div>
  );
}

export default App;
```

src/App.css

This file contains all the global CSS styles for the Amazon theme.

```
/* Amazon Color Palette */
:root {
  --amz-dark-blue: #131921;
```

```css
  --amz-light-blue: #232f3e;
  --amz-orange: #febd69;
  --amz-yellow: #f0c14b;
  --amz-white: #ffffff;
  --amz-grey: #f0f2f2;
  --amz-link-blue: #007185;
  --amz-text-dark: #0F1111;
  --amz-text-light: #555;
  --amz-prime-blue: #00a8e1;
  --amz-price-red: #B12704;
}

/* Global Styles */
body {
  font-family: 'Amazon Ember', Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: var(--amz-grey);
  color: var(--amz-text-dark);
}

/* Main container for page content */
.amz-main-container {
  max-width: 1500px;
  margin: 20px auto;
  padding: 0 20px;
}

a {
  text-decoration: none;
  color: var(--amz-link-blue);
}
a:hover {
  text-decoration: underline;
  color: var(--amz-orange);
}

button {
  font-family: inherit;
  cursor: pointer;
  border-radius: 8px;
  border: 1px solid;
}
```

```
/* Common Amazon Button Style */
.amz-button {
  background: linear-gradient(to bottom, #f7dfa5, #f0c14b);
  border-color: #a88734 #9c7e31 #846a29;
  color: var(--amz-text-dark);
  padding: 8px 16px;
  font-size: 14px;
}
.amz-button:hover {
  background: linear-gradient(to bottom, #f5d78e, #eeb933);
}

/* --- Header / Navbar --- */
.amz-navbar {
  background-color: var(--amz-dark-blue);
  color: var(--amz-white);
  padding: 10px 15px;
  display: flex;
  align-items: center;
  gap: 15px;
  /* Make it stick to the top */
  position: sticky;
  top: 0;
  z-index: 100;
}

.amz-navbar .amz-logo {
  font-size: 1.8rem;
  font-weight: bold;
  color: var(--amz-white);
  padding-right: 10px;
}
.amz-navbar .amz-logo:hover {
  text-decoration: none;
  outline: 1px solid var(--amz-white);
  border-radius: 3px;
}

.amz-navbar .nav-link {
  color: var(--amz-white);
  display: flex;
  flex-direction: column;
```

```css
  padding: 5px;
  border: 1px solid transparent;
}
.amz-navbar .nav-link:hover {
  border-color: var(--amz-white);
  border-radius: 3px;
  text-decoration: none;
}

.nav-link .nav-line-one {
  font-size: 12px;
  color: #ccc;
}
.nav-link .nav-line-two {
  font-size: 14px;
  font-weight: 700;
}

.amz-searchbar {
  flex-grow: 1;
  display: flex;
}
.amz-searchbar input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: none;
  border-radius: 4px 0 0 4px;
}
.amz-searchbar button {
  background-color: var(--amz-orange);
  border: none;
  padding: 0 15px;
  border-radius: 0 4px 4px 0;
}
.amz-searchbar button:hover {
  background-color: #f3a847;
}

.amz-basket-link {
  display: flex;
  align-items: flex-end;
  gap: 5px;
```

```
}
.amz-basket-link .basket-count {
  font-size: 18px;
  font-weight: 700;
  color: var(--amz-orange);
}

/* --- Home Page --- */
.homepage-hero {
  background: linear-gradient(to bottom, #232f3e, #131921);
  color: var(--amz-white);
  padding: 40px;
  text-align: center;
  margin-bottom: 20px;
  border-radius: 8px;
}
.homepage-hero h1 {
  margin: 0;
}

.category-grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
  gap: 20px;
}
.category-card {
  background-color: var(--amz-white);
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 1px 3px rgba(0,0,0,0.1);
}
.category-card h2 {
  margin-top: 0;
}
.category-card img {
  width: 100%;
  height: 250px;
  object-fit: cover;
  margin-bottom: 10px;
  border-radius: 4px;
}

/* --- Product List Page --- */
```

```
.product-list-container {
  background-color: var(--amz-white);
  padding: 20px;
  border-radius: 8px;
}

.product-list-grid {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));
  gap: 20px;
}

/* --- Product Card Component --- */
.amz-product-card {
  background-color: var(--amz-white);
  border: 1px solid #ddd;
  border-radius: 8px;
  padding: 15px;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  transition: box-shadow 0.2s;
}
.amz-product-card:hover {
  box-shadow: 0 4px 12px rgba(0,0,0,0.1);
}
.amz-product-card img {
  max-width: 100%;
  height: 220px;
  object-fit: contain;
  margin-bottom: 15px;
}
.amz-product-card .product-title {
  font-size: 1rem;
  color: var(--amz-link-blue);
  display: -webkit-box;
  -webkit-line-clamp: 2;
  -webkit-box-orient: vertical;
  overflow: hidden;
  text-overflow: ellipsis;
  margin-bottom: 5px;
  height: 2.4em; /* 2 lines */
}
```

```
.amz-product-card .product-title:hover {
  color: var(--amz-orange);
}
.amz-product-card .product-rating {
  font-size: 14px;
  color: #f0c14b;
  margin-bottom: 5px;
}
.amz-product-card .product-price {
  font-size: 1.3rem;
  font-weight: 500;
  color: var(--amz-text-dark);
  margin-bottom: 5px;
}
.amz-product-card .product-price sup {
  font-size: 0.7rem;
  top: -0.5em;
}
.amz-product-card .product-prime-logo {
  font-size: 14px;
  font-weight: bold;
  color: var(--amz-prime-blue);
  margin-bottom: 10px;
}
.amz-product-card .product-prime-logo::before {
  content: '✓';
  color: var(--amz-orange);
}

/* --- Product Detail Page --- */
.amz-product-detail {
  display: grid;
  grid-template-columns: 1.5fr 1fr;
  gap: 30px;
  background-color: var(--amz-white);
  padding: 30px;
  border-radius: 8px;
}
.amz-product-detail .detail-image {
  text-align: center;
}
.amz-product-detail .detail-image img {
  max-width: 100%;
  max-height: 450px;
  object-fit: contain;
}
.amz-product-detail .detail-info {
  border-left: 1px solid #eee;
  padding-left: 30px;
}
.amz-product-detail .detail-info .detail-price {
  font-size: 1.7rem;
  color: var(--amz-price-red);
  margin: 10px 0;
}
.amz-product-detail .detail-info .detail-prime {
  font-size: 16px;
  font-weight: bold;
  color: var(--amz-prime-blue);
}

/* --- Cart (Basket) Page --- */
.amz-basket-page {
  display: grid;
  grid-template-columns: 3fr 1fr;
  gap: 20px;
  align-items: start;
}
.amz-basket-items {
  background-color: var(--amz-white);
  padding: 20px;
  border-radius: 8px;
}
.amz-basket-items h1 {
  margin-top: 0;
  padding-bottom: 10px;
  border-bottom: 1px solid #ddd;
}
.amz-basket-empty {
  font-size: 1.2rem;
  padding: 20px 0;
}
.amz-basket-empty a {
  font-weight: 700;
}
```

```
.amz-basket-subtotal {
  background-color: var(--amz-white);
  padding: 20px;
  border-radius: 8px;
  position: sticky;
  top: 80px; /* Stick below the navbar */
}
.amz-basket-subtotal h3 {
  margin-top: 0;
  font-weight: 400;
  font-size: 1.2rem;
}
.amz-basket-subtotal h3 strong {
  font-weight: 700;
  color: var(--amz-price-red);
}

/* --- Cart Item Component --- */
.amz-cart-item {
  display: grid;
  grid-template-columns: 150px 1fr;
  gap: 15px;
  padding: 15px 0;
  border-bottom: 1px solid #eee;
}
.amz-cart-item img {
  width: 150px;
  height: 150px;
  object-fit: contain;
}
.amz-cart-item .item-info h4 {
  margin: 0 0 5px 0;
  font-size: 1.1rem;
  color: var(--amz-link-blue);
  font-weight: 700;
}
.amz-cart-item .item-info .item-price {
  font-size: 1.1rem;
  font-weight: 700;
  margin-bottom: 10px;
}
.amz-cart-item .item-info .delete-button {
  background: none;
  border: none;
  color: var(--amz-link-blue);
  padding: 0;
  font-size: 13px;
  border-radius: 0;
}
.amz-cart-item .item-info .delete-button:hover {
  text-decoration: underline;
  color: var(--amz-text-light);
}

/* --- Footer --- */
.amz-footer {
  background-color: var(--amz-light-blue);
  color: var(--amz-white);
  text-align: center;
  padding: 20px;
  margin-top: 30px;
  font-size: 14px;
}
.amz-footer:hover {
  background-color: #37475A;
  cursor: pointer;
}
```

4. Data (src/data/)

src/data/products.js

This file acts as a simple, fake database for your website.

```
// This file acts as a simple, fake database for your website.

const products = [
  {
    id: 1,
    name: 'Echo Dot (5th Gen) | Smart speaker with clock and Alexa | Cloud Blue',
    price: 59.99,
    rating: 4.5,
    ratingsCount: 14550,
    isPrime: true,
    description: 'Our most popular smart speaker with Alexa, the Echo Dot features a sleek
design and improved audio for vibrant sound anywhere in your home. With the new clock
```

```
display, you can see the time, alarms, and timers at a glance.',
    imageUrl:
'[https://placehold.co/400x400/99e6ff/000000?text=Echo+Dot](https://placehold.co/400x400/99e6ff/000000?text=Echo+Dot)',
  },
  {
    id: 2,
    name: 'Kindle Paperwhite (16 GB) – Now with a 6.8" display and adjustable warm light',
    price: 139.99,
    rating: 4.8,
    ratingsCount: 32010,
    isPrime: true,
    description: 'Read comfortably with a glare-free, 6.8" Paperwhite display, adjustable warm light, and up to 10 weeks of battery life. Now with 16GB of storage.',
    imageUrl:
'[https://placehold.co/400x400/e6e6e6/000000?text=Kindle](https://placehold.co/400x400/e6e6e6/000000?text=Kindle)',
  },
  {
    id: 3,
    name: 'Amazon Basics 48-Pack AA High-Performance Alkaline Batteries, 10-Year Shelf Life',
    price: 15.99,
    rating: 4.7,
    ratingsCount: 750000,
    isPrime: false,
    description: '48-pack of 1.5-volt AA alkaline batteries for reliable performance across a wide range of devices. 10-year leak-free shelf life.',
    imageUrl:
'[https://placehold.co/400x400/f0c14b/000000?text=AA+Batteries](https://placehold.co/400x400/f0c14b/000000?text=AA+Batteries)',
  },
  {
    id: 4,
    name: 'Instant Pot Duo 7-in-1 Electric Pressure Cooker, 6 Quart, Stainless Steel',
    price: 89.00,
    rating: 4.7,
    ratingsCount: 150000,
    isPrime: true,
    description: 'The #1 best-selling pressure cooker. 7-in-1 appliance: pressure cooker, slow cooker, rice cooker, steamer, sauté pan, yogurt maker, and warmer. 6-quart capacity.',
    imageUrl:
'[https://placehold.co/400x400/cccccc/000000?text=Instant+Pot](https://placehold.co/400x400/cccccc/000000?text=Instant+Pot)',
  },
  {
    id: 5,
    name: 'Logitech MX Master 3S - Wireless Performance Mouse, Ergonomic, 8K DPI, Quiet Clicks',
    price: 99.99,
    rating: 4.6,
    ratingsCount: 12800,
    isPrime: true,
    description: 'An icon remastered. Experience ultimate precision and performance with MagSpeed electromagnetic scrolling, 8K DPI tracking, and quiet clicks.',
    imageUrl: '[https://placehold.co/400x400/333333/ffffff?text=MX+Master+3S](https://placehold.co/400x400/333333/ffffff?text=MX+Master+3S)',
  },
  {
    id: 6,
    name: 'Simple Modern Insulated Tumbler with Lid and Straw | 24oz',
    price: 23.99,
    rating: 4.9,
    ratingsCount: 89000,
    isPrime: true,
    description: 'This double-walled, vacuum-insulated tumbler keeps your drinks cold for hours. Made from 18/8 stainless steel and fits in most cupholders.',
    imageUrl: '[https://placehold.co/400x400/66cccc/000000?text=Tumbler](https://placehold.co/400x400/66cccc/000000?text=Tumbler)',
  },
];

// Function to simulate fetching all products from an API
export const getProducts = () => {
  return new Promise((resolve) => {
    // Simulate a network delay
    setTimeout(() => {
      resolve(products);
    }, 300);
  });
};

// Function to simulate fetching a single product by its ID
export const getProductById = (id) => {
  return new Promise((resolve) => {
    // Find the product in our fake database
    const product = products.find((p) => p.id === parseInt(id));
    setTimeout(() => {
      resolve(product);
    }, 300);
  });
};
```

5. Context (src/context/)

src/context/CartContext.js

This file handles the global state for the shopping basket using React's Context API.

```
import React, { createContext, useState, useContext } from 'react';

// 1. Create the Context
// This is what components will use to get the cart state
const CartContext = createContext();

// 2. Create a custom hook for easy access
// Instead of importing useContext and CartContext in every file,
// we just import useCart()
export const useCart = () => {
  return useContext(CartContext);
};

// 3. Create the Provider component
// This component will wrap our entire app and "provide" the cart state
export const CartProvider = ({ children }) => {
  const [cartItems, setCartItems] = useState([]);

  // Function to add a product to the cart
  const addToCart = (product) => {
    setCartItems((prevItems) => {
      // Check if the item is already in the cart
      const itemExists = prevItems.find((item) => item.id === product.id);

      if (itemExists) {
        // If it exists, just alert the user (or you could increase quantity)
        alert(`${product.name} is already in your basket!`);
        return prevItems;
      } else {
        // If it doesn't exist, add it to the cart
        alert(`Added ${product.name} to basket!`);
        // We add quantity: 1 for future use
        return [...prevItems, { ...product, quantity: 1 }];
      }
    });
  };

  // Function to remove an item from the cart
  const removeFromCart = (productId) => {
    setCartItems((prevItems) => {
      // Return a new array that filters out the item with the matching id
      return prevItems.filter((item) => item.id !== productId);
    });
  };

  // 4. Define the value to be passed to consumers
  // This object contains the state and the functions to update it
  const value = {
    cartItems,
    addToCart,
    removeFromCart,
  };

  // 5. Return the provider with the value
  return <CartContext.Provider value={value}>{children}</CartContext.Provider>;
};
```

6. Components (src/components/)

These are the reusable UI building blocks for the application.

src/components/Header.js

This component is the main navigation bar at the top of the page.

```
import React from 'react';
import { Link } from 'react-router-dom';
import { useCart } from '../context/CartContext';
// In a real app, you'd use an icon library like react-icons
// import { FiSearch, FiShoppingCart } from 'react-icons/fi';

const Header = () => {
```

```
// Get the cartItems from our custom hook
const { cartItems } = useCart();
const basketItemCount = cartItems.length; // Count items

return (
  <header className="amz-navbar">
    {/* Logo */}
    <Link to="/" className="amz-logo">
      amazon
    </Link>

    {/* Location (Hardcoded) */}
    <Link to="#" className="nav-link">
      <span className="nav-line-one">Deliver to</span>
      <span className="nav-line-two">Your Name</span>
    </Link>

    {/* Search Bar */}
    <div className="amz-searchbar">
      <input type="text" placeholder="Search Amazon-E-Com-Website..." />
      <button>{/* <FiSearch /> */ '🔍'}</button>
    </div>

    {/* Links */}
    <Link to="#" className="nav-link">
      <span className="nav-line-one">Hello, [Your Name]</span>
      <span className="nav-line-two">Account & Lists</span>
    </Link>

    <Link to="/products" className="nav-link">
      <span className="nav-line-one">Browse</span>
      <span className="nav-line-two">All Products</span>
    </Link>

    {/* Basket */}
    <Link to="/cart" className="nav-link amz-basket-link">
      {/* <FiShoppingCart size="30px" /> */}
      <span className="basket-count">{basketItemCount}</span>
      <span className="nav-line-two" style={{alignSelf: 'flex-end'}}>
        Basket
      </span>
    </Link>
  </header>
  );
};

export default Header;
```

src/components/ProductCard.js

This component displays a single product preview in the product list.

```
import React from 'react';
import { Link } from 'react-router-dom';
import { useCart } from '../context/CartContext';

// This component displays a single product in a grid
const ProductCard = ({ product }) => {
  // Get the addToCart function from our context
  const { addToCart } = useCart();

  const handleAddToBasket = () => {
    addToCart(product);
  };

  // Format the price like Amazon
  const priceParts = product.price.toFixed(2).split('.');

  return (
    <div className="amz-product-card">
      <div>
        <Link to={`/products/${product.id}`}>
          <img src={product.imageUrl} alt={product.name} />
        </Link>

        <Link to={`/products/${product.id}`}>
          <h3 className="product-title">{product.name}</h3>
        </Link>

        <div className="product-rating">
          {'⭐'.repeat(Math.floor(product.rating))} {product.rating}
          <span style={{color: '#555'}}> ({product.ratingsCount})</span>
        </div>

        <div className="product-price">
```

```
          <sup>$</sup>
          <strong>{priceParts[0]}</strong>
          <sup>{priceParts[1]}</sup>
        </div>

        {product.isPrime && (
          <div className="product-prime-logo">
            Prime
          </div>
        )}
      </div>

      <button className="amz-button" onClick={handleAddToBasket}>
        Add to Basket
      </button>
    </div>
  );
};

export default ProductCard;
```

src/components/CartItem.js

This component displays a single item in the shopping basket page.

```
import React from 'react';
import { useCart } from '../context/CartContext';

// This component displays a single item in the shopping basket
const CartItem = ({ item }) => {
  // Get the function to remove items
  const { removeFromCart } = useCart();

  return (
    <div className="amz-cart-item">
      <img src={item.imageUrl} alt={item.name} />
      <div className="item-info">
        <h4>{item.name}</h4>
        <p className="item-price">
          <strong>${item.price.toFixed(2)}</strong>
        </p>
        {item.isPrime && (
          <div className="product-prime-logo" style={{marginBottom: '10px'}}>
            Prime
          </div>
        )}

        {/* Button to remove the item from the cart */}
        <button
          className="delete-button"
          onClick={() => removeFromCart(item.id)}
        >
          Delete
        </button>
      </div>
    </div>
  );
};

export default CartItem;
```

7. Pages (src/pages/)

These components represent the main pages of the application.

src/pages/HomePage.js

This is the main landing page (homepage) of the site.

```
import React from 'react';
import { Link } from 'react-router-dom';

// This is the landing page
const HomePage = () => {
  return (
    <div className="homepage-container">
      {/* Hero Banner */}
      <div className="homepage-hero">
        <h1>Welcome to Amazon E-Com Website</h1>
        <p>
          This is a React-based clone built by [Your Name]. Browse products,
          add to cart, and enjoy the dynamic experience.
        </p>
        <Link to="/products">
          <button className="amz-button" style={{fontSize: '1.1rem', padding: '10px 20px'}}>
      Shop All Products
     </button>
    </Link>
   </div>

   {/* Category Grid */}
   <h2>Shop by Category</h2>
   <div className="category-grid">
    <div className="category-card">
     <h3>Shop Electronics</h3>
     <img
src="[https://placehold.co/300x250/99e6ff/000000?text=Electronics](https://placehold.co/300x250/99e6ff/000000?text=Electronics)" alt="Electronics" />
     <Link to="/products">See more</Link>
    </div>

    <div className="category-card">
     <h3>Shop Home & Kitchen</h3>
     <img
src="[https://placehold.co/300x250/cccccc/000000?text=Home+&+Kitchen](https://placehold.co/300x250/cccccc/000000?text=Home+&+Kitchen)" alt="Home" />
     <Link to="/products">See more</Link>
    </div>

    <div className="category-card">
     <h3>Reading Corner</h3>
     <img
src="[https://placehold.co/300x250/e6e6e6/000000?text=Books](https://placehold.co/300x250/e6e6e6/000000?text=Books)" alt="Books" />
     <Link to="/products">See more</Link>
    </div>

    <div className="category-card">
     <h3>Everyday Essentials</h3>
     <img
src="[https://placehold.co/300x250/f0c14b/000000?text=Essentials](https://placehold.co/300x250/f0c14b/000000?text=Essentials)" alt="Deals" />
     <Link to="/products">Shop now</Link>
    </div>
   </div>
  </div>
 );
};
```

```
export default HomePage;
```

src/pages/ProductListPage.js

This page fetches and displays all available products.

```
import React, { useState, useEffect } from 'react';
import ProductCard from '../components/ProductCard';
import { getProducts } from '../data/products'; // Import our fake data function

// This page shows all products
const ProductListPage = () => {
  const [products, setProducts] = useState([]);
  const [loading, setLoading] = useState(true);

  // useEffect runs once when the component loads
  useEffect(() => {
    // Fetch products from our "database"
    getProducts().then((data) => {
      setProducts(data);
      setLoading(false); // We have the data, stop loading
    });
  }, []); // The empty array [] means this runs only once

  if (loading) {
    return <h2>Loading products...</h2>;
  }

  return (
    <div className="product-list-container">
      <h2>Results</h2>
      <p>{products.length} Products Found</p>
      <div className="product-list-grid">
        {/* Loop over the products and create a card for each one */}
        {products.map((product) => (
          <ProductCard key={product.id} product={product} />
        ))}
      </div>
    </div>
  );
};
```

```
export default ProductListPage;
```

src/pages/ProductDetailPage.js

This page fetches and displays the details for a single product.

```
import React, { useState, useEffect } from 'react';
import { useParams } from 'react-router-dom';
import { getProductById } from '../data/products';
import { useCart } from '../context/CartContext';

// This page shows details for one specific product
const ProductDetailPage = () => {
  const [product, setProduct] = useState(null);
  const [loading, setLoading] = useState(true);

  // useParams gets the ":id" part from the URL
  const { id } = useParams();

  // Get the addToCart function
  const { addToCart } = useCart();

  useEffect(() => {
    // Fetch the specific product by its ID
    getProductById(id).then((data) => {
      setProduct(data);
      setLoading(false);
    });
  }, [id]); // Re-run this effect if the ID in the URL changes

  if (loading) {
    return <h2>Loading...</h2>;
  }

  if (!product) {
    return <h2>Product not found.</h2>;
  }

  const priceParts = product.price.toFixed(2).split('.');

  return (
    <div className="amz-product-detail">
      {/* Column 1: Image */}
      <div className="detail-image">
        <img src={product.imageUrl} alt={product.name} />
      </div>

      {/* Column 2: Info */}
      <div className="detail-info">
        <h2>{product.name}</h2>
        <div className="product-rating">
          {'⭐'.repeat(Math.floor(product.rating))} {product.rating} stars | {product.ratingsCount} ratings
        </div>
        <hr />
        <div className="detail-price">
          <span style={{fontSize: '14px', color: 'var(--amz-text-light)'}}>Price: </span>
          <span style={{color: 'var(--amz-price-red)'}}>${priceParts[0]}.{priceParts[1]}</span>
        </div>
        {product.isPrime && (
          <div className="detail-prime" style={{fontSize: '18px'}}>
            ✓Prime FREE Delivery
          </div>
        )}
        <hr />
        <h3>About this item</h3>
        <p>{product.description}</p>
        <button
          className="amz-button"
          style={{width: '100%', padding: '12px'}}
          onClick={() => addToCart(product)}
        >
          Add to Basket
        </button>
      </div>
    </div>
  );
};

export default ProductDetailPage;
```

src/pages/CartPage.js

This page displays all items currently in the shopping basket and the total price.

```
import React from 'react';
import { useCart } from '../context/CartContext';
import CartItem from '../components/CartItem';
import { Link } from 'react-router-dom';

// This page shows the shopping basket
const CartPage = () => {
  // Get the cartItems state
  const { cartItems } = useCart();

  // Calculate the total price
  const total = cartItems.reduce((sum, item) => sum + item.price, 0);

  return (
    <div className="amz-basket-page">
      {/* Column 1: Items */}
      <div className="amz-basket-items">
        <h1>Shopping Basket</h1>
        <hr />
        {cartItems.length === 0 ? (
          <div className="amz-basket-empty">
            <p>Your Amazon Basket is empty.</p>
            <Link to="/products">Continue Shopping</Link>
          </div>
        ) : (
          <div>
            {/* Loop over items and render a CartItem component for each */}
            {cartItems.map((item) => (
              <CartItem key={item.id} item={item} />
            ))}
          </div>
        )}
      </div>

      {/* Column 2: Subtotal */}
      {cartItems.length > 0 && (
        <div className="amz-basket-subtotal">
          <h3>
            Subtotal ({cartItems.length} {cartItems.length === 1 ? 'item' : 'items'}):
            <strong> ${total.toFixed(2)}</strong>
          </h3>
          <button className="amz-button" style={{width: '100%'}}>
            Proceed to checkout
          </button>
        </div>
      )}
    </div>
  );
};

export default CartPage;
```