

Buy For Baby - Get Your Kid's need

Ms. Danwada Sharanya, Mr. Enjamuri Sagar, Ms. Pasarakonda Nikhitha,
Mr. Srinadh Karne

Dept. of EEE
Vardhaman College of Engineering.
Hyderabad, India

Abstract- E-commerce, or business-to-consumer trade, is the most well-known commercial application of the World Wide Web.. The primary objective of an e-commerce site is to sell products and services online. The aim of this project is to develop an e-commerce website particularly for infants till teenage group that provides a seamless and user-friendly on-line shopping experience. The e-commerce website will serve as a one-stop destination for customers to explore and purchase a wide range of products conveniently from the comfort of their homes. The platform will prioritize user experience by implementing an intuitive and visually appealing interface that ensures easy navigation and efficient browsing. Key features we have included in this platform are: User registration and Authentication, Product Catalog and Search, Shopping cart and checkout, reviews and ratings and Responsive design which can be worked in smart phones, tablets etc. In order to develop an e-commerce website, a number of Technologies must be studied and understood. These include multi-tiered architecture, server and client-side scripting techniques, implementation technologies such as HTML, CSS and JS, relational databases (Sanity.io).

Keywords- React JS, E-Commerce, Online shopping, User experience, Product catalog, Search functionality, Shopping cart, Reviews and ratings, Responsive design

1. Introduction

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming common place. As we can see we have many online platforms to buy products online particularly for kids like firstcry.com, amazon.in, flipkart etc., but these applications are not giving the better User Experience (UX) which definitely a bad feedback to that organization. The objective of this project is to develop the best UX E-commerce application for the client as the experience can be bought from the comfort of home through the Internet. Over three quarters of the 10,000 respondents report having purchased items online. The most cited reason for using the web for personal shopping was convenience (65). Although the issue of security remains the primary reason why more people do not purchase items online. As more people gain confidence in current encryption technologies, more and more users can be expected to frequently purchase items online. A good e-commerce site should present the following factors to the customers for better usability:

- Easy scanning and selecting items in a list.
- Effective categorical organization of products.
- Simple navigation from home page to information and order links for specific products.
- Obvious shopping links or buttons.
- Minimal and effective security notifications or messages.

1.1 Consistent Layout Of Product Information

The main objective is to develop a Single Page Web application under the domain of E-Commerce specifically for the kids. So that the users can feel very easy to access the products and can buy the products within few seconds. As compared to the existing web applications this platform is very easy to use and easy to understand the details. After conducting a survey on the customers of the existing platforms, one thing we can hear is the bad user experience.

Building an eCommerce website typically requires several key components to ensure functionality, security, and a smooth user experience. Here are the essential components:

1. Domain Name: Choosing a unique domain name for the project is very much important, because domain of the project represents our application. It is the identity of the application.

Web Hosting: Finding a reliable web hosting provider that offers the required storage, bandwidth, and security features for your eCommerce website. In this case, Netlify is the best option for hosting the deployment and hosting. Because it is free of cost and provides free storage up to some limit.

2. Design and Themes: Customizing the appearance of BUYFORBABY using pre-designed themes or by creating a unique design that aligns with your brand. First of all, we need to create a theme for the website. Customization of the website is available in some online platforms like Figma.com etc. So that we get a clear idea of the website UI.

Product Catalog: A well-structured product catalog that includes product descriptions, images, prices, variations (if applicable), and any other relevant information.

Search Functionality: Incorporate a search feature on your website to help users find specific products quickly.

Responsive Design: Ensure your website is mobile-friendly and optimized for various devices and screen sizes. Responsive design is crucial to accommodate users accessing your website from smartphones and tablets.

Types of E-Commerce Business Models

Generally, we have mainly 4 types of traditional business models. Each has its benefits and challenges, and many companies operate in several of these categories simultaneously. Knowing what bucket your big idea fits in will help you think creatively about what your opportunities and threats might be.

1. B2C – Business to consumer

B2C businesses sell to their end-user. The B2C model is the most common business model, so there are many unique approaches under this umbrella. Anything you buy in an online store as a consumer — think wardrobe, household supplies, entertainment — is done as part of a B2C transaction. The decision-making process for a B2C purchase is much shorter than a business-to-business (B2B) purchase, especially for items that have a low value. Think about it: it's much easier for you to decide on a new pair of tennis shoes than for your company to vet and purchase a new email service provider or food caterer. Because of this shorter sales cycle, B2C businesses typically spend less marketing dollars to make a sale, but also have a lower average order value and less recurring orders than their B2B counterparts and B2C doesn't only include products, but services as well.

2. B2B – Business to Business

In a B2B business model, a business sells its product or service to another business. Sometimes the buyer is the end user, but often the buyer resells to the consumer. B2B transactions generally have a longer sales cycle, but higher

order value and more recurring purchases. Recent B2B innovators have made a place for themselves by replacing catalogs and order sheets with e-commerce storefronts and improved targeting in niche markets. In 2020, close to half of B2B buyers are millennials — nearly double the amount from 2012. As younger generations enter the age of making business transactions, B2B selling in the online space is becoming more important.

3.C2B – Consumer To Business

C2B businesses allow individuals to sell goods and services to companies. In this e-commerce model, a site might allow customers to post the work they want to be completed and have businesses bid for the opportunity. Affiliate marketing services would also be considered C2B. Elance (now Upwork) was an early innovator in this model by helping businesses hire freelancers. The C2B e-commerce model's competitive edge is in pricing for goods and services. This approach gives consumers the power to name their price or have businesses directly compete to meet their needs. Recent innovators have creatively used this model to connect companies to social media influencers to market their products.

4.C2C – Consumer to Consumer

C2C business - also called an online marketplace connects consumers to exchange goods and services and typically make their money by charging transaction or listing fees. Online businesses like Craigslist and eBay pioneered this model in the early days of the internet. C2C businesses benefit from self-propelled growth by motivated buyers and sellers, but face a key challenge in quality control and technology maintenance.

Another important factor in the design of an e-commerce site is feedback. The interactive cycle between a user and a web site is not complete until the web site responds to a command entered by the user. According to Norman, "feedback—sending back to the user information about what action has actually been done, what result has been accomplished—is a well-known concept in the science of control and information theory. Imagine trying to talk to someone when you cannot even hear your own voice, or trying to draw a picture with a pencil that leaves no mark: there would be no feedback".

Web site feedback often consists of a change in the visual or verbal information presented to the user. Simple examples include highlighting a selection made by the user or filling a field on a form based on a user's selection from a pull-down list. Another example is using the sound of a cash register to confirm that a product has been added to an electronic shopping cart.

Completed orders should be acknowledged quickly. This may be done with an acknowledgment or fulfillment page. The amount of time it takes to generate and download this page, however, is a source of irritation for many e-commerce users. Users are quick to attribute meaning to events. A blank page, or what a user perceives to be "a long time" to receive an acknowledgment, may be interpreted as "there must be something wrong with the order." If generating an acknowledgment may take longer than what may be reasonably expected by the user, then the design should include intermediate feedback to the user indicating the progress being made toward acknowledgment or fulfillment.

2. Page Layout

The developed application has different pages like home page, store page, login page, search for the products page, single product page, cart (products added for the check-out).

According to the requirements of the application, website constitutes of six different pages.

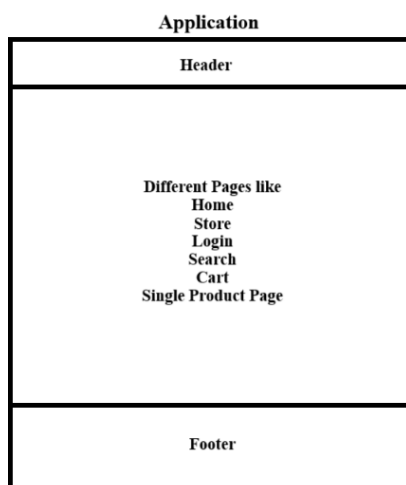
The first page is home page which is the main page of the website. When a user enters into the website, user sees the first page. It shows the top trending products, most bought products and overview of the all categories available. The first page consists of the header which is used to navigate through the different pages in the website and consists of the body of the page where the content of the application is inserted and finally footer which is the last part of the website describes about the website and also about the copyrights, social media platforms, contact details.

The second page which is the store page consists of all products in it. All the products are differentiated according to their category. Each time a developed React js component named Product Card is rendered in the store page, which has some properties to navigate to the separate single product page and add to the cart functionality. The third page is login page. User can create his/her own account in the website. We have provided an authentication functionality for the user. Google authentication is provided so that user can login/sign out from the account. Another page is search page, though we can include the search functionality in the store page itself. But to achieve dynamic search functionality a separate page has been designed, where an input field is provided to search with cool animations. A single product page is also provided in the application, because we cannot provide all the product information in the product card. So it is necessary to include a dynamic separate single page of the product. It includes the detailed information of the product and add to cart functionality. Cart page, though it is a component but considered as a page, the products which have been added in the bag can be seen here.

Main Application

1. Home page
 - Popular products
 - Details of the categories
 - Trusted partners section
2. Store page
 - Left section of the categories
 - Products of the selected category
3. Login page
 - Sign In with Google section
4. Search page
 - Products of the search term
5. Single product page
 - Detailed information of product
 - You may also like section
6. Cart page
 - Selected products to check out.

3. Page Style



While designing any application, it is very necessary to create a clean and organized layout that is easy to navigate. In any website, it is necessary to have some sections to have a good user experience. For any application it consists of header, body and a footer. Header and footer will always be static in the website but the content between them will change. In the header we provide all the links of the website to navigate to different pages across the website.

Order of the sections as follows:

1. Header (static)
2. Body (could be any page)
3. Footer (static)

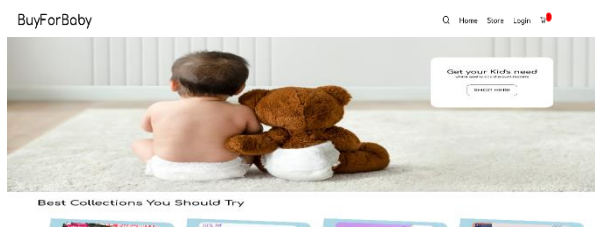


Fig. 1. Page style

In order to design a web site, the relational database must be designed first. Conceptual design can be divided into two parts: The data model and the process model. The data model focuses on what data should be stored in the database while the process model deals with how the data is processed. To put this in the context of the relational database, the data model is used to design the relational tables. The process model is used to design the queries that will access and perform operations on those tables.

3.1. Front-end Design

This is the main section of the development of the website. This includes some of the programming languages like HTML, CSS and JAVASCRIPT(React).

1. HTML

HTML is a markup language used to structure and present content on the World Wide Web. It defines the structure of web pages and determines how elements within a page are displayed in a web browser. HTML documents consist of a series of elements (tags) enclosed in angle brackets. Elements can be nested within each other, forming a hierarchical structure. The structure is defined using opening tags, content, and closing tags.

Tags: HTML tags are used to define different elements within a web page. Tags are written in angle brackets

("</> "). They can be self-closing tags or have opening and closing tags. For example, the <p> tag represents a paragraph, and the tag represents an image.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First Webpage</title>
  </head>
  <body>
    <h1>
      My First Webpage
    </h1>
    <p>This is a paragraph...</p>
  </body>
</html>
```

Fig. 2. HTML example

HTML is the foundation of web development and an essential skill for creating and structuring web pages. It provides the backbone for displaying content on the internet and plays a vital role in creating user-friendly and accessible websites.

2. CSS

CSS stands for Cascading Style Sheets. It is a style sheet language used for describing the presentation and formatting of a document written in HTML or XML. CSS allows web developers to control the visual appearance of web pages by defining how elements should be displayed, including layout, colors, fonts, and other stylistic aspects.

```
/* Selectors and Declarations */
h1 {
  color: blue;
  font-size: 24px;
}

p {
  color: green;
  font-size: 18px;
  margin-bottom: 20px;
}

img {
  width: 200px;
  border: 2px solid black;
}

/* Box Model */
body {
  padding: 20px;
}
```

Fig. 3. CSS example

Here are some key points about CSS:

Separation of Style and Content: CSS separates the presentation layer from the content layer in web development. HTML defines the structure and content of a web page, while CSS is responsible for specifying how that content should be presented.

Selectors and Declarations: CSS uses selectors to target HTML elements and apply styles to them. Selectors can be based on element names, class names, IDs, attributes, or relationships between elements. Declarations consist of a property and a value, defining the specific styles to be applied.

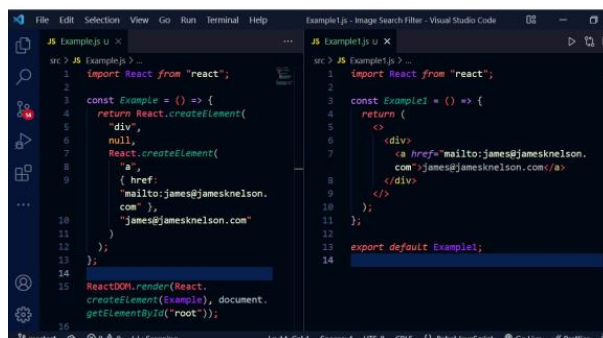
3. React JS

React.js, often referred to as React, is an open-source JavaScript library used for building user interfaces (UIs). It was developed by Facebook and released in 2013. Here's some important information about React.js:

3.2 Component-Based Architecture: React.js follows a component-based architecture, where the UI is broken down into reusable and independent components. Components encapsulate their own logic and state, making it easier to build complex UIs by composing smaller, reusable pieces.

1. Virtual DOM: React introduces a virtual representation of the DOM (Document Object Model) known as the Virtual DOM. Instead of directly manipulating the browser's DOM, React builds a virtual tree of elements and efficiently updates only the necessary parts when the underlying data changes. This approach improves performance and enhances the user experience.

2. JSX: React uses JSX (JavaScript XML), an extension to JavaScript that allows you to write HTML-like syntax within JavaScript code. JSX enables the seamless integration of HTML markup and JavaScript logic, making it easier to create React components.



```
src > JS Example.js u x
1 import React from "react";
2
3 const Example = () => {
4   return React.createElement(
5     "div",
6     null,
7     React.createElement(
8       "a",
9       { href: "mailto:james@jamesknelson.com" },
10      "james@jamesknelson.com"
11     )
12   );
13 };
14
15 ReactDOM.render(React.createElement(Example), document.getElementById("root"));
16
```

```
src > JS Example1.js u x
1 import React from "react";
2
3 const Example1 = () => {
4   return (
5     <div>
6       <a href="mailto:james@jamesknelson.com" >james@jamesknelson.com</a>
7     </div>
8   );
9 };
10
11 export default Example1;
12
13
14
```

Fig. 4. React js example

3.3 Data Base Design

Sanity is a content platform that provides a headless content management system (CMS) for building websites, applications, and digital experiences. It offers a structured content management approach, allowing developers and content creators to work collaboratively on creating and delivering content.

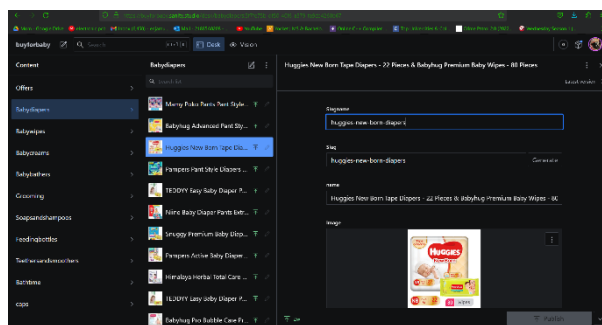
1. Real-time Collaboration: Sanity offers real-time collaboration features, enabling multiple content contributors to work simultaneously on the same content. This capability is beneficial for teams working on content creation, editing, and reviewing processes.

2. Structured Content: Sanity allows developers to define content schemas using its flexible and customizable schema definition language, called Sanity Schema. Content is structured based on defined schemas, ensuring consistency and organization in managing different content types.

3. Rich Text Editing: Sanity provides a powerful and extensible rich text editing interface called Portable Text. It allows content creators to create and format text content using various formatting options, including headings, lists, links, images, and more.

4. Customizable Content APIs: Sanity provides a Graph QL-based query language, known as GROQ (Graph-Relational Object Queries), to query and retrieve content from the CMS. It also offers RESTful APIs and client libraries in multiple programming languages, allowing developers to access and consume content from their applications easily.

5. Deployment and Scalability: Sanity offers scalable hosting options, including a serverless infrastructure that automatically scales based on traffic and demand. It provides deployment options to suit different needs, including self-hosting, deploying to cloud platforms like AWS and Google Cloud, or using Sanity's managed hosting service.

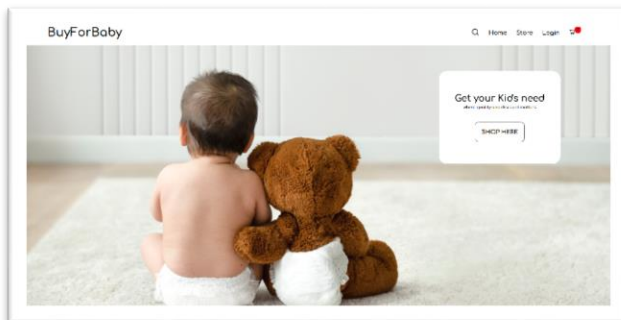


Sanity is known for its developer-friendly approach, flexibility, and collaborative features. It empowers developers and content teams to build content-driven applications and experiences while providing powerful tools for content management and delivery.

4. Figures and Tables

Here are the outputs of the web application designed according to the requirements. All the images are divided by the number of pages included in the website.

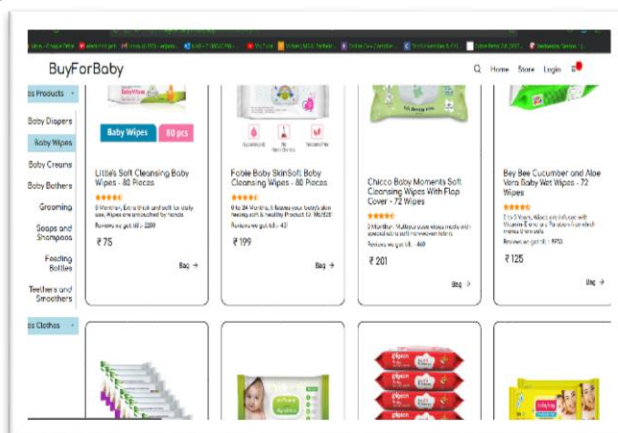
1.Home page



```
front-end / src / components / home / JS / Home.jsx (8) Home
  1 import React from "react";
  2 import './Home.css';
  3 import { Link } from "react-router-dom";
  4
  5 //components
  6 import HomeChild from './HomeChild';
  7 import Offers from './Offers';
  8 import HomeSticky from './HomeSticky';
  9 import HomeCategory from './HomeCategory';
  10 import HomeSticky1 from './HomeSticky1';
  11 import HomePartners from './HomePartners';
  12
  13 const Home = () => {
  14   window.addEventListener("load", () => {
  15     window.scrollTo(0, 0);
  16   });
  17
  18   return (
  19     <div className="home-main">
  20       <HomeChild /> <!--child background image -->
  21       <div className="home-container">
  22         <div className="left">
  23           <h3>Get your Kids need</h3>
  24           <p>where quality and discount matters</p>
  25         </div>
  26         <div className="right">
  27           <Link className="home-btn" to="/store">
  28             Shop here
  29           </Link>
  30         </div>
  31       </div>
  32       <Offers />
  33       <HomeSticky />
  34       <HomeCategory />
  35       <HomeSticky1 />
  36       <HomePartners />
  37     </div>
  38   );
  39 };
  40
  41 export default Home;
```

Fig. 7 & 8. Home page of the application

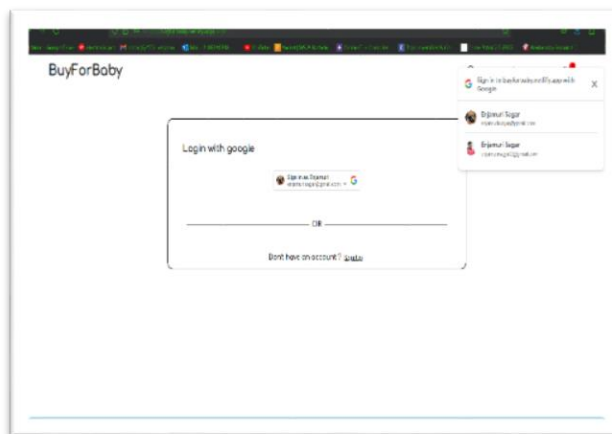
2. Store page



```
1 import React, { useState } from "react";
2 import { StoreCategory } from "./StoreCategory";
3 //UseParams
4 import { useParams, Link } from "react-router-dom";
5 //clients
6 import { client, urlFor } from "../lib/client";
7 //Spinner
8 import Spinner from "../Spinner";
9 //Icons
10 import { BsCurrencyRupee } from "react-icons/bs";
11 import { BsOutlineArrowRight } from "react-icons/ai";
12 import { BsStarHalf } from "react-icons/fa";
13 import { FaStar } from "react-icons/fa";
14 import StoreNav from "../StoreNav";
15 //Global
16 import { useStateContext } from "../context/StateContext";
17 const StoreCategory = () => {
18   const { slug } = useParams();
19   console.log(slug);
20   const [data, setData] = useState({});
21   const [loader, setLoader] = useState(false);
22   useEffect(() => {
23     const query = `?l_type = ${slug}`;
24     setLoader(true);
25     client
26       .fetch(query)
27       .then((res) => {
28         setLoader(false);
29         setData(res);
30       })
31       .catch((err) => console.log(err));
32   }, [slug]);
33   console.log(data);
34 }
35 export default StoreCategory;
```

Fig. 9&10. Store page of the application

3. Login page



```
1 import React from "react";
2 import { useNavigate } from "react-router-dom";
3 import { GoogleLogin } from "react-google-login";
4 //decode jwt token
5 import jwt_decode from "jwt-decode";
6 //Icons
7 import { FcGoogle } from "react-icons/fc";
8 //Global
9 import { useStateContext } from "../context/StateContext";
10 const Login = () => {
11   const { setUser } = useStateContext();
12   const navigate = useNavigate();
13   const googleResponse = (response) => {
14     var decoded = jwt_decode(response.credential);
15     console.log(decoded);
16     localStorage.setItem("user", JSON.stringify(decoded));
17     setUser({
18       name: decoded.name,
19       image: decoded.picture,
20     });
21     navigate("/");
22   };
23   return (
24     <div className="login main">
25       <div className="container">
26         <div className="login container">
27           <div className="login btn">
28             <GoogleLogin
29               onSuccess={googleResponse}
30               onError={() => console.log("err")}
31             />
32           </div>
33         </div>
34       </div>
35     </div>
36   );
37 }
38 export default Login;
```

Fig. 11& 12. Login page of the application

4. Cart page

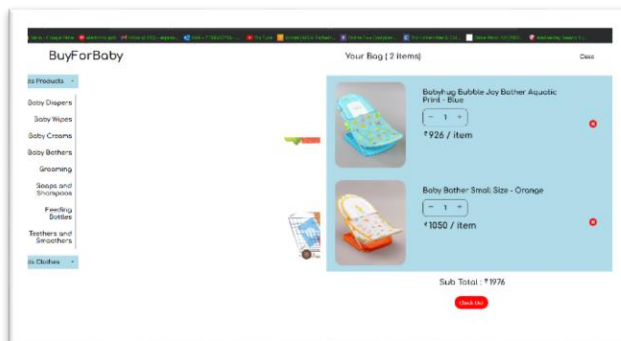


Fig. 13. Cart page of the application

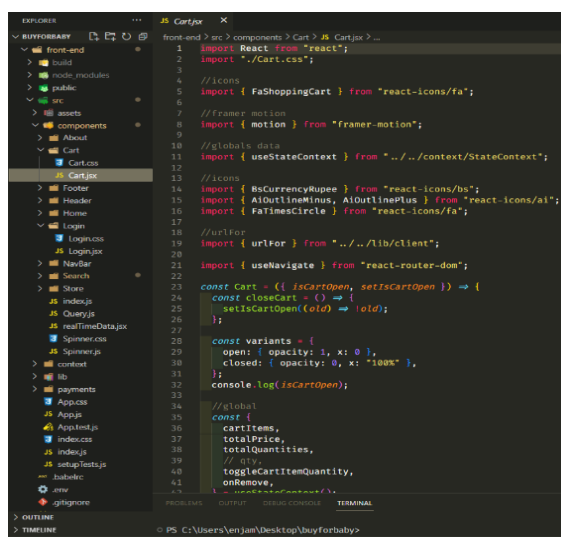


Fig. 14. Code of Cart page of the application

5. Search page

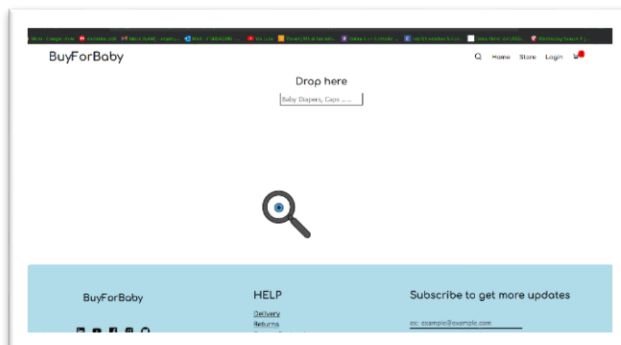
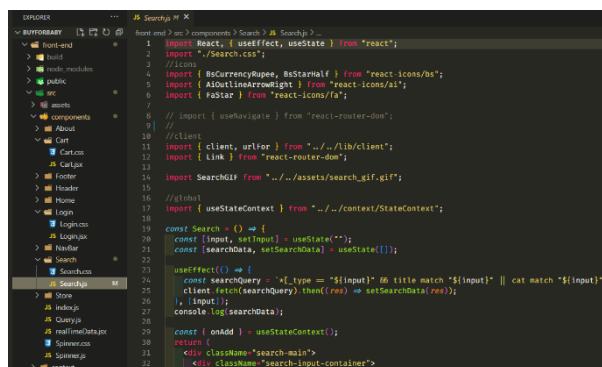


Fig. 15. Search page of the application



```

1 import React, { useEffect, useState } from "react";
2 import './Search.css';
3 //Icons
4 import { BsCurrencyRupee, BsStarHalf } from "react-icons/bs";
5 import { AiOutlineArrowRight } from "react-icons/ai";
6 import { FaStar } from "react-icons/fa";
7
8 // import { useNavigate } from "react-router-dom";
9
10 //Client
11 import { client, urlFor } from "../lib/client";
12 import { Link } from "react-router-dom";
13
14 import SearchGif from "../assets/search_gif.gif";
15
16 //Global
17 import { useStateContext } from "../context/StateContext";
18
19
20 const Search = () => {
21   const [input, setInput] = useState("");
22   const [searchData, setSearchData] = useState([]);
23
24   useEffect(() => {
25     const searchQuery = `*${input}* && title match "${input}" && cat match "${input}"
26     client.fetch(searchQuery).then((res) => setSearchData(res));
27     console.log(searchData);
28   }, [input]);
29
30   const { onAdd } = useStateContext();
31   return (
32     <div className="search-main">
33       <div className="search-input-container">

```

Fig. 16. Code of Search page of the application

5. Some Helpful Hints

1. Equations

In React.js, we don't typically use React directly to handle mathematical equations or calculations. Instead, we use JavaScript, the language React is built on, to perform such computations. React is primarily a library for building user interfaces, and it doesn't include specific functionality for mathematical operations.

2. Abbreviations and Acronyms

Some of the abbreviations used in the text, even after they have already been defined in the abstract.

JSX – JavaScript Xml.

API – Application Programmable Interface.

HTML – Hyper Text Markup Language.

CSS – Cascaded Style Sheets

JS – Java script.

Props – Properties.

DOM – Document Object Model.

3. Links and Bookmarks

Project Live Link: <https://buy-for-baby.netlify.app/>

6. Conclusion

In conclusion, React offers immense potential for the development of dynamic and interactive web applications. With its component-based architecture and efficient virtual DOM rendering, React has gained widespread popularity and support from the developer community. By harnessing React's capabilities, you have created a powerful and scalable application that provides an engaging user experience. The modular nature of React allows for easy maintenance and future enhancements, ensuring that your app can adapt to evolving user needs and technological advancements. Furthermore, the future scope of your React app is vast. You can continue to refine and expand its functionality, explore mobile app development, optimize performance, transform it into a PWA, or integrate it with backend technologies. Overall, your React app lays a solid foundation for building upon, and with ongoing improvements and innovation, it has the potential to become even more successful and impactful in the future.



References

1. "Designing e-commerce Websites: 60 UX Design Tips" by Stefan Mancevski
2. E-commerce User Experience: A Practical Handbook" by Vivek Sharma.
3. The E-Commerce Book: Building the E-Empire" by Steffano Korper and Juanita Ellis This book provides a comprehensive overview of e-commerce, covering topics such as online business models, website design, marketing strategies, payment systems, and logistics.
4. Siddegowda C J, Gitu mani Borah, Chandru KA, "Reactframework (creating a web application with reactnative)," International Journal of Recent Trends in Engineering & Research (IJRTER) vol. 04, Issue 03, March 2018.
5. Ritwik C and Anitha Sandeep, "React.js and frontend development," IRJET Volume: 07 Issue: 04, Apr 2020.
6. Minati Biswal, "React lifecycle methods," Researchgate Article, December 2019.
7. Ch Rajesh, K S V Krishna Srikanth, "Research on HTML5 in Web Development", International Journal of Computer Science and Information Technologies, Vol.5 (2), 2014, 2408-2412.
8. ReactJS.org, 'ReactJS official'. [Online].
9. How to use axios with React(2018).
10. firstcry.com (<https://www.firstcry.com/>)
11. sanity.io (<https://sanity.io/>)
12. npmjs (<https://www.npmjs.com>)
13. stripe.com
14. E-commerce Platforms (<https://ecommerce-platforms.com/>)
15. Shopify Blog (www.shopify.com/blog): Covers various aspects of e-commerce website design, including marketing, optimization, and user experience.
16. Smashing Magazine (www.smashingmagazine.com): Offers articles, tutorials, and case studies on web design, e-commerce, and user experience.