NAVYkart

A PROJECT REPORT

Submitted By

Aviral Srivastava
(2000290140035)
Yash Mittal
(2000290140140)
Vishal Bhardwaj
(2000290140133)
Nikhil Upadhyay
(2000290140076)

Submitted in partial fulfillment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Dr. Shashank Bhardwaj
Assistant Professor of KIET Group Of Institutions, Ghaziabad

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(JAN 2022)

TABLE OF CONTENTS

CERTIFICATE

Certified that **Yash Mittal <200029014005825>, Aviral Srivastava <200029014005720>, Vishal Bhardwaj <200029014005818>, Nikhil Upadhyay <200029014005761>** have carried out the project work having "**NAVYkart**" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:13-01-2022

Yash Mittal (2000290140140)

Aviral Srivastava (2000290140035)

Vishal Bhardwaj(2000290140133)

Nikhil Upadhyay(2000290140076)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:13-01-2022

Dr. Shashank Bhardwaj
Assistant Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Ajay Shrivastava
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Our Project “NavyKart” is an Online Shopping Website. The objective of this online shopping website is to improve the services of customers. The main features of this project is high accuracy, design flexibility and easy availability. This website allows customers to shop and buy the items online.

This project is an attempt to provide the advantages of online shopping to the customers. This online shopping website allows users to shop from anywhere and anytime and it provides users to find more variety of products with fewer expenses. Users can compare multiple items at a time in this application. This application saves lot of time and users can avoid crowd for shopping.

This project is designed from a user point of view. The user friendly design helps the users in accomplishing their task with ease. Attempts have been made to keep the design simple and understandable. Technologies like HTML, CSS, Javascript, React-JS and firebase have been used in this project.

The user module: User should fill the registration by submitting all the details like email and address. Users can make search for the product and add the item to cart.

Now a days the life style of the people is different. People feel uncomfortable and time consuming for going crowded markets. So, E-Shopping is a boon as it saves lot of time. Online shopping is a process whereby consumers directly buy goods, services etc. from a seller without an intermediary service over the Internet. Shoppers can visit web stores from the comfort of their house and shop as by sitting in front of the computer. Online stores are usually available 24 hours a day and many consumers have internet access both at work and at home. So it is very convenient for them to shop Online. One of the most enticing factors about online shopping, particularly during holiday season is, it alleviates the need to wait in long lines or search from a store for a particular item. Variety of goods are available in online. So the researcher want to know the preference of the consumers. So fifty respondents were met and data were collected regarding their preference towards shopping online.

ACKNOWLEDGEMENTS

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Shashank Bhardwaj** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Yash Mittal

Aviral Srivastava

Vishal Bhardwaj

Nikhil Upadhyay

1. INTRODUCTION

With the advancement in technology and science, people can now do various things at the comfort of their homes and one such thing is online shopping. It has gained a lot of spotlight due to its ever-increasing demand and craze among people. Online shopping refers to the way of purchasing things online without actually going to the physical stores. People nowadays are busy earning their livelihood and they hardly get any time to go for shopping, however, with the advent of online shopping, they can now order anything be it clothes, footwear, gadgets, appliances and much more. There are numerous advantages of online shopping, let's take an insight into it. Online shopping is the best option for people who do not have much time and are busy in their office and business work. It is a convenient way for the people who cannot withstand crowded places and malls for shopping so they can sit at their home or office and can order anything anytime. Online shopping offers a vast variety of options which is not possible with physical shopping. You can browse through different websites and can choose the product according to your requirements. Online shopping does not require physical cash and you can make payments through your debit or credit cards although you have an option of cash on delivery. Well, everything comes with some disadvantages as well. Although online shopping is easy and convenient, however sometimes it disappoints you as things ordered online may not seem the same when they arrive at your door, the colour, the size or something else can be different from the actual item. Also, there are some websites that are fake and provide you with great offers to tempt you and befool you at the end. So it is important that we should do online shopping wisely and with much care in order to avoid any kind of chaos later on.

1.1 Overview

Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser or a mobile app. Consumers find a product of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping search engine, which displays the same product's availability and pricing at different e-retailers. customers can shop online using a range of different computers and devices, including desktop computers, TV, Washing Machine, laptops, tablet computers and smartphones.

1.1.1 Project Description

The main features of this project is high accuracy, design flexibility and easy availability. This website allows customers to shop and buy the items online. This project is an attempt to provide the advantages of online shopping to the customers. This online shopping website allows users to shop from anywhere and anytime and it provides users to find more variety of products with fewer expenses. Users can compare multiple items at a time in this application. This application saves lot of time and users can avoid crowd for shopping. This project is designed from a user point of view. The user friendly design helps the users in accomplishing their task with ease. Attempts have been made to keep the design simple and understandable.

1.1.2 Project Scope

Purchasing and selling products and services over the internet without the need of going physically to the market is what online shopping all about. Online shopping is just like a retail store shopping that we do by going to the market, but it is done through the internet. Online shopping has made shopping painless and added more fun. Online stores offer product description, pictures, comparisons, price and much more. Few examples of these are Amazon.com, ebay.com, framt.com and the benefits of online shopping is that by having direct access to consumer ,the online stores can offer products that cater to the needs of consumer ,cookies can be used for tracking the customer selection over the internet or what is of their interest when they visit the site again . Online shopping makes use of digital technology for managing the flow of information, products, and payment between consumer, site owners and suppliers. Online shopping can be either B2B (business to business) or B2C (business to consumer).

Shopping cart is one of the important facility provided in online shopping, this lets customer to browse different goods and services and once they select an item to purchase they can place the item in shopping cart, and continue browsing till the final selection. Customers can even remove the items from shopping cart that were selected earlier before they place the final order. It reminds us of shopping basket that we carry in departmental store.

1.1.3 Hardware / Software used in Project

Processor : Dual Core 2nd generation
RAM : 2GB or more
HARD DISK : 80GB
OPERATING SYSTEM: Windows 7 and Above
HTML

CSS
Javascript
React-JS
Firebase

2. Feasibility Study

A feasibility study assesses the operational, technical and economic merits of the proposed project. The feasibility study is intended to be a preliminary review of the facts to see if it is worthy of proceeding to the analysis phase. From the systems analyst perspective, the feasibility analysis is the primary tool for recommending whether to proceed to the next phase or to discontinue the project.

The feasibility study is a management-oriented activity. The objective of a feasibility study is to find out if an information system project can be done and to suggest possible alternative solutions.

Projects are initiated for two broad reasons:

1. Problems that lend themselves to systems solutions
2. Opportunities for improving through: (a) upgrading systems (b) altering systems (c) installing new systems

A feasibility study should provide management with enough information to decide:

- Whether the project can be done
- Whether the final product will benefit its intended users and organization
- What are the alternatives among which a solution will be chosen
- Is there a preferred alternative

2.1 Technical Feasibility

A large part of determining resources has to do with assessing technical feasibility. It considers the technical requirements of the proposed project. The technical requirements are then compared to the technical capability of the organization. The systems project is considered technically feasible if the internal technical capability is sufficient to support the project requirements.

The analyst must find out whether current technical resources can be upgraded or added to in a manner that fulfills the request under consideration. This is where the expertise of system analysts is beneficial, since using their own experience and their contact with vendors they will be able to answer the question of technical feasibility.

The essential questions that help in testing the operational feasibility of a system include the following:

Is the project feasible within the limits of current technology?

Does the technology exist at all?

Is it available within given resource constraints?

Is it a practical proposition?

Manpower- programmers, testers & debuggers

Software and hardware

Are the current technical resources sufficient for the new system?

Can they be upgraded to provide to provide the level of technology necessary for the new system?

Do we possess the necessary technical expertise, and is the schedule reasonable?

Can the technology be easily applied to current problems?

Does the technology have the capacity to handle the solution?

Do we currently possess the necessary technology?

2.2 Operational Feasibility

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented.

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

Operational feasibility reviews the willingness of the organization to support the proposed system. This is probably the most difficult of the feasibilities to gauge. In order to determine this feasibility, it is important to understand the management commitment to the proposed project. If the request was initiated by management, it is likely that there is management support and the system will be accepted and used. However, it is also important that the employee base will be accepting of the change.

The essential questions that help in testing the operational feasibility of a system include the following:

· Does current mode of operation provide adequate throughput and response time?

· Does current mode provide end users and managers with timely, pertinent, accurate and useful formatted information?

· Does current mode of operation provide cost-effective information services to the business?

· Could there be a reduction in cost and or an increase in benefits?

· Does current mode of operation offer effective controls to protect against fraud and to guarantee accuracy and security of data and information?

· Does current mode of operation make maximum use of available resources, including people, time, and flow of forms?

· Does current mode of operation provide reliable services

· Are the services flexible and expandable?

- Are the current work practices and procedures adequate to support the new system?
- If the system is developed, will it be used?
- Manpower problems
- Labour objections
- Manager resistance
- Organizational conflicts and policies
- Social acceptability
- Government regulations
- Does management support the project?
- Are the users not happy with current business practices?
- Will it reduce the time (operation) considerably?
- Have the users been involved in the planning and development of the project?
- Will the proposed system really benefit the organization?
- Does the overall response increase?
- Will accessibility of information be lost?
- Will the system affect the customers in considerable way?
- Legal aspects
- How do the end-users feel about their role in the new system?
- What end-users or managers may resist or not use the system?
- How will the working environment of the end-user change?
- Can or will end-users and management adapt to the change?

2.3 Economical Feasibility

Economic analysis could also be referred to as cost/benefit analysis. It is the most frequently used method for evaluating the effectiveness of a new system. In economic analysis the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system. An entrepreneur must accurately weigh the cost versus benefits before taking an action.

Possible questions raised in economic analysis are:

- Is the system cost effective?
- Do benefits outweigh costs?
- The cost of doing full system study
- The cost of business employee time
- Estimated cost of hardware
- Estimated cost of software/software development
- Is the project possible, given the resource constraints?
- What are the savings that will result from the system?

- Cost of employees' time for study
- Cost of packaged software/software development
- Selection among alternative financing arrangements (rent/lease/purchase)

The concerned business must be able to see the value of the investment it is pondering before committing to an entire system study. If short-term costs are not overshadowed by long-term gains or produce no immediate reduction in operating costs, then the system is not economically feasible, and the project should not proceed any further. If the expected benefits equal or exceed costs, the system can be judged to be economically feasible. Economic analysis is used for evaluating the effectiveness of the proposed system.
The economical feasibility will review the expected costs to see if they are in-line with the projected budget or if the project has an acceptable return on investment. At this point, the projected costs will only be a rough estimate. The exact costs are not required to determine economic feasibility. It is only required to determine if it is feasible that the project costs will fall within the target budget or return on investment. A rough estimate of the project schedule is required to determine if it would be feasible to complete the systems project within a required timeframe. The required timeframe would need to be set by the organization.

3. Database Design

3.1 Flow Chart

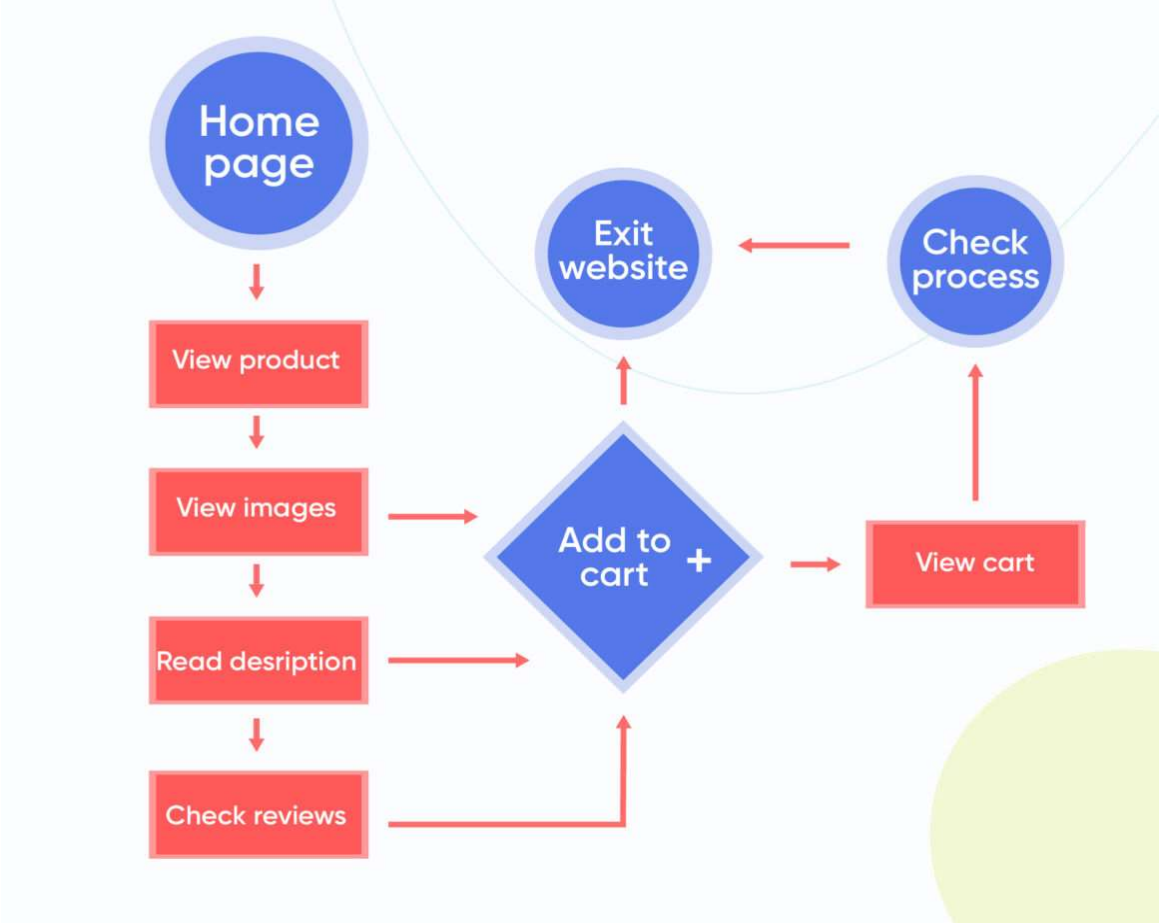


Figure 1: *Flow Chart*

3.2 Use Case Diagram

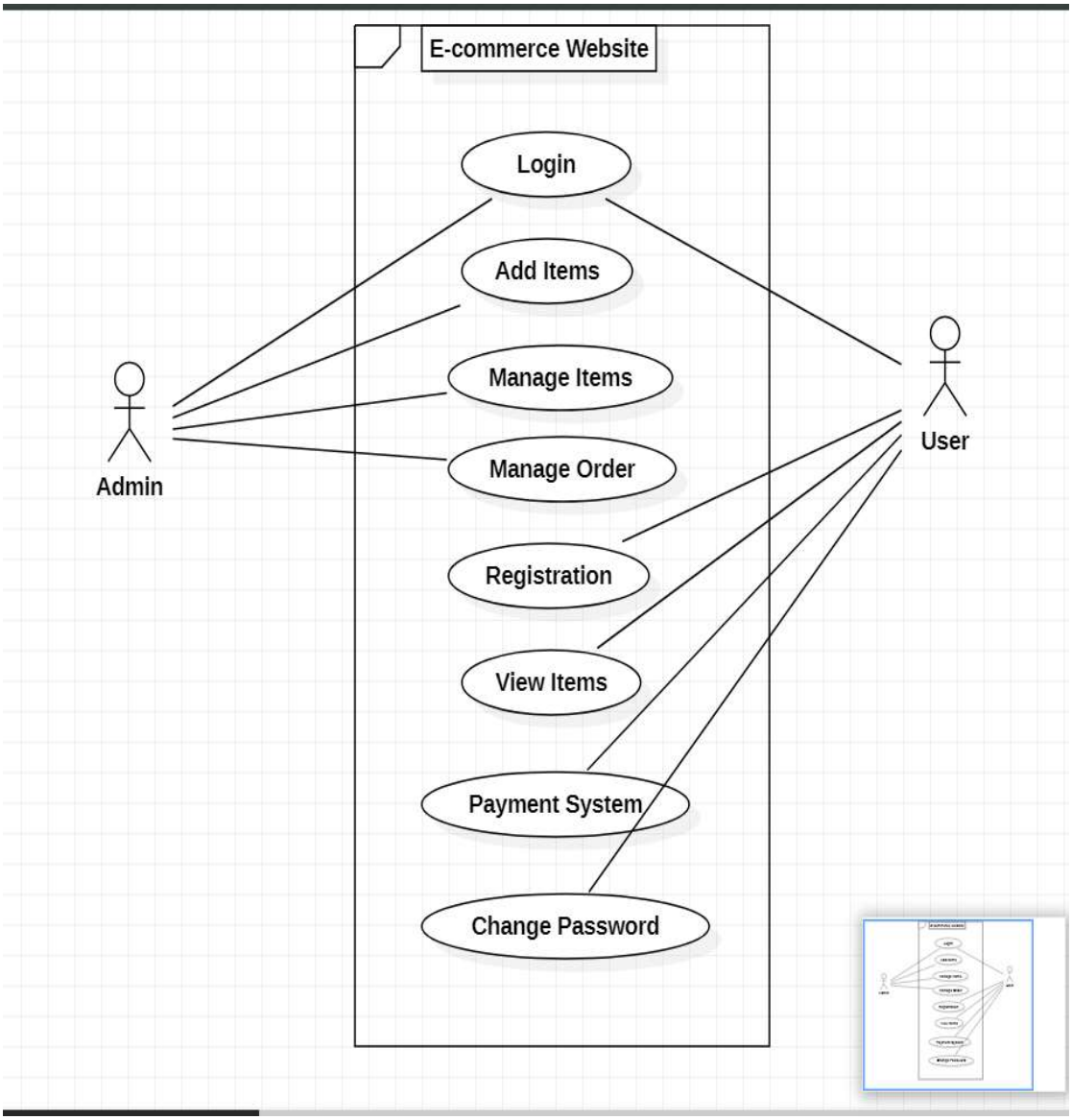
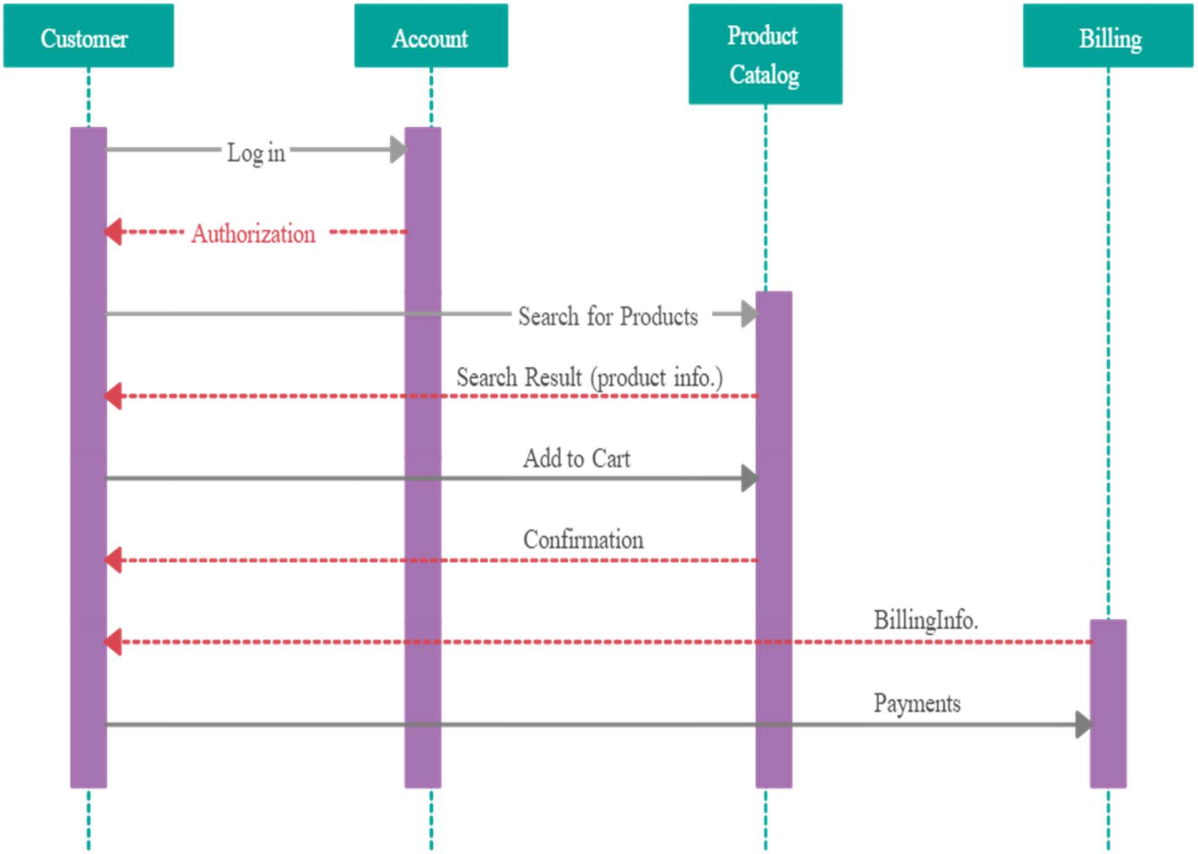


Figure 2: *Use-Case Diagram of Online Shopping Website*

3.3 Sequence Diagram



4. Form Design

4.1 Input / Output Form (Screenshot)

Login

Email Address

Password

Forgot password?

Login

By continuing, you agree to Navykart's Conditions of Use and Privacy Notice.

New to Navykart?

Create your Navykart account

Create Account

Your Name

Email

Password

At least 6 characters

Confirm Password

must be same as password

We will send you a text to verify your phone.
Message and Data rates may apply.

Continue

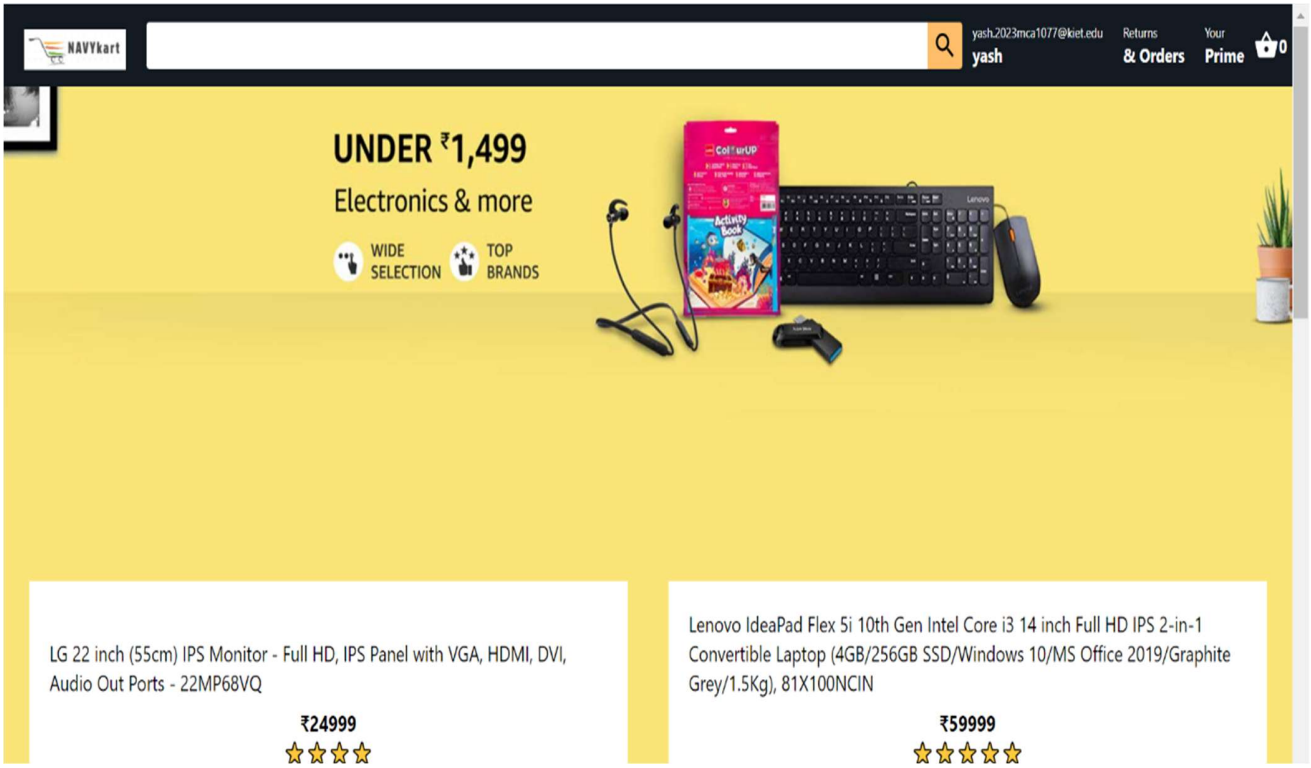
NAVYkart

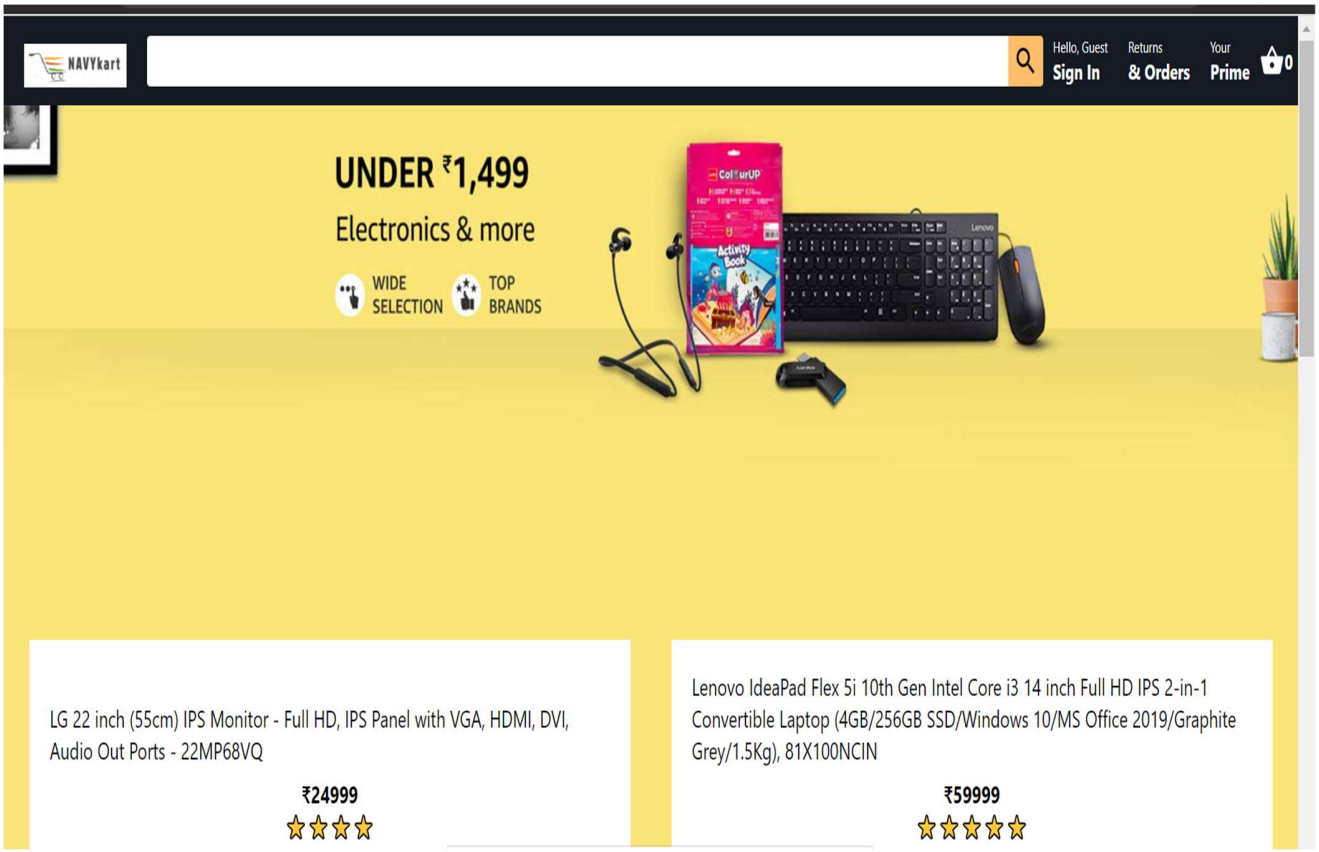
Password assistance

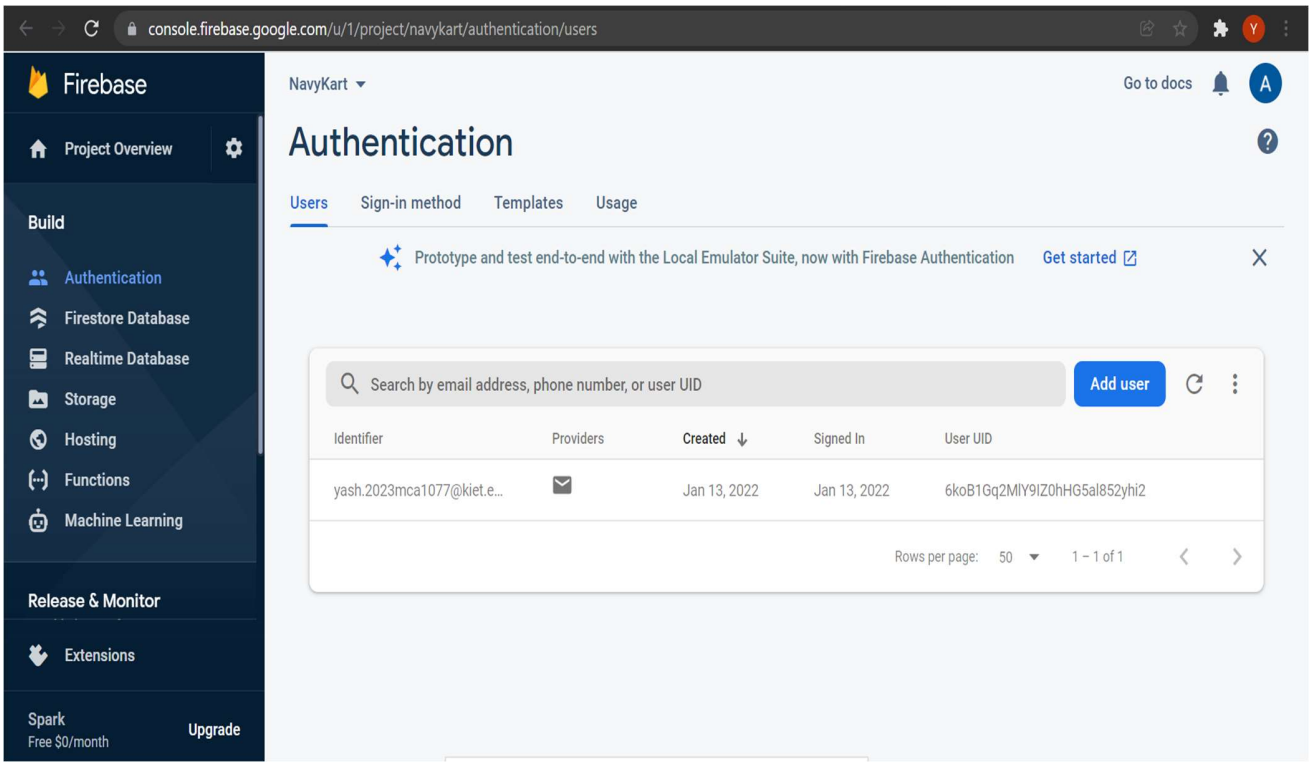
Enter the email address associated with your Navykart account.

Registered email

Continue




NAVYkart
Hello, Guest
Sign In
Returns
& Orders
Your
Prime
0
UNDER ₹1,499
Electronics & more
WIDE SELECTION
TOP BRANDS
LG 22 inch (55cm) IPS Monitor - Full HD, IPS Panel with VGA, HDMI, DVI, Audio Out Ports - 22MP68VQ
₹24999
Lenovo IdeaPad Flex 5i 10th Gen Intel Core i3 14 inch Full HD IPS 2-in-1 Convertible Laptop (4GB/256GB SSD/Windows 10/MS Office 2019/Graphite Grey/1.5Kg), 81X100NCIN
₹59999


console.firebase.google.com/u/1/project/navykart/authentication/users
Firebase
Project Overview
Build
Authentication
Firestore Database
Realtime Database
Storage
Hosting
Functions
Machine Learning
Release & Monitor
Extensions
Spark
Free $0/month
Upgrade
NavyKart
Go to docs
Authentication
Users
Sign-in method
Templates
Usage
Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication
Get started
Search by email address, phone number, or user UID
Add user
Identifier
Providers
Created
Signed In
User UID
yash.2023mca1077@kiet.e...
Jan 13, 2022
Jan 13, 2022
6koB1Gq2MlY9IZ0hHG5al852yhi2
Rows per page:
50
1 – 1 of 1

5. Coding

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="Web site created using create-react-app"
    />
    <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />

    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <title>NAVYkart</title>
  </head>
  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
      </body>
</html>
```

Cart.css

```
.cart {
  display: flex;
  padding: 20px;
  height: fit-content;
}

.cart_left > img {
  width: 100%;
}

.cart_left > h2 {
  border-bottom: 1px solid lightgray;
  padding-bottom: 8px;
  margin-right: 10px;
}

.cart_items {
  height: fit-content;
  display: flex;
  flex-direction: column;
```

```
}
```

Cart.js

```
import React from 'react'
import './Cart.css'
import Ad from './Assets/Ad.jpg'
import Subtotal from './Subtotal'
import { useStateValue } from './StateProvider'
import CartProduct from './CartProduct'


function Cart() {

    const[{basket},dispatch]=useStateValue();

    return (
        <div className="cart">
            <div className="cart_left">
                <img src={Ad} alt="" />
                <h2>Your shopping Cart</h2>

                <div className="cart_items">
                    { basket?.map(item=>(
                            <CartProduct
                            img={item.image}
                            description={item.Description}
                            price={item.price}
                            rating={item.rating}
                            />))
                    }
                </div>
            </div>

            <div className="cart_right">
                <Subtotal/>
            </div>
        </div>
    )
}

export default Cart
```

CartProduct.css

```css
.cartProduct {
    display: flex;
    padding: 15px;
    border: 1px solid #dddddd;
    border-radius: 3px;
    background-color: #f6f6f6;
    margin-bottom: 20px;
}

.cartProduct > img {
  object-fit: contain;
  width: 150px;
}
.cartProduct > div {
  margin-left: 10px;
}
.cartProduct > div > button {
  cursor: pointer;
  color: #111;
  background-color: #f0c14b;
  border: 1px solid #846a29;
  border-radius: 2px;
  margin-top: 3px;
  padding: 2px;
}
```

CartProduct.js

```jsx
import React from 'react'
import './CartProduct.css'

function CartProduct(props) {

    const {img,description,price,rating}=props

    const removeFromBasket=()=>{

    }

    return (
        <div className="cartProduct">
            <img src={img} alt=""/>
            <div>
                <div>{description}</div>
                <strong>₹{price}</strong>
                <div>{Array(rating).fill().map((_,i)=>(
                    <span>☆</span>))}
                </div>
                <button onClick={removeFromBasket}>Remove from Cart</button>
```

```
            </div>
        </div>
    )
}

export default CartProduct
```

CheckOut.js

```
import React from 'react'

function CheckOut() {
    return (
        <div>
            <h1>CheckOut page</h1>
        </div>
    )
}

export default CheckOut
```

Header.css

```
.header {
  display: flex;
  align-items: center;
  height: 60px;
  background-color: #131a22;
  position: sticky;
  top: 0px; /*learn Remember*/
  z-index: 100; /*learn Remember*/
  width: 100%;
}

/* logo */
.header_logo {
  object-fit: contain;
  width: 100px;
  margin: 0 20px;
  margin-top: 11px;
}

/* search_bar */
.header_search {
  flex: 1; /*learn */
  display: flex;
```

```
  align-items: center;
}

.header_input {
  width: 100%;
  border-radius: 3px 0px 0px 3px;
  padding: 10px;
  height: 14px;
  border: none;
  outline: none;
}

.header_input[type="text"] {
  font-size: 18px;
}

.header_icon {
  background-color: #febd69;
  padding: 5px; /*learn Remember //to increase the outer area of icon*/
  border-radius: 0px 3px 3px 0px;
}

/* navigation */
.header_nav {
  display: flex;
  justify-content: space-evenly;
  align-items: center;

  text-decoration: none;
}

.header_options {
  display: flex;
  flex-direction: column;
  margin-left: 10px;
  margin-right: 10px;
  color: white;
}

.header_options:hover {
  outline: 1px solid white;
  outline-offset: 5px;
}

.header_lineOne {
  font-size: 10px;
  cursor: pointer;
}

.header_lineTwo {
  font-weight: 700;
  font-size: 14px;
```

```
  cursor: pointer;
}

.header_optionCart {
  color: white;
  display: flex;
  align-items: center;
  margin-right: 5px;
  cursor: pointer;
}

.header_optionCart:hover {
  outline: 1px solid white;
  outline-offset: 5px;
}

/* link text decoration */

.header_linkToLogin {
  text-decoration: none;
}

.header_linkToCart {
  text-decoration: none;
}

/* sign in dropdown menu */

.header_signIn {
  position: relative;
  display: flex;
  flex-direction: column;
  align-items: center;
}

/* for the smaller triangle near sign in*/
.header_arrowup {
  display: none;
  position: absolute;
  top: 30px;
  height: 0px;
  width: 0px;
  border-bottom: 8px solid white;
  border-right: 8px solid transparent;
  border-left: 8px solid transparent;
}
.header_signIn:hover .header_arrowup{
  display: block;
}

.header_SignInMenu {
  display: none;
```

```
  position: absolute;
  top: 36px;
  background-color: #ffffff;
  width: 200px;
  height: 80px;
}

.header_signIn:hover .header_SignInMenu {
  display: block;
}

.header_signInmenuBtn {
  width: 100px;
  padding: 5px;
  border-radius: 2px;
  background-color: #febd69;
  border: 1px solid #846a29;
  cursor: pointer;
  color: #111;
  font-size: 14px;
  outline: none;
  margin: 10px 0px;
  margin-left: 45px;
}

.header_signInmenuBtn:hover {
  background-color: #eb9525;
}

.header_signInmenuBtn:focus {
  background-color: #eb9525;
}

.header_SignInMenuText {
  font-weight: 400;
  color: rgb(99, 95, 95);
  font-size: 12px;
  margin-left: 25px;
}

.header_SignInMenu_signup {
  color: #0066c0;
  cursor: pointer;
  font-size: 12px;
}

.header_SignInMenu_signup:hover {
  color: red;
  text-decoration: underline;
}

.header_SignInMenu_signupLink {
```

```
  text-decoration: none;
}
```

Header.js

```
import React from 'react'
import './Header.css'
import navykartLogo from './Assets/navykart-logo.jpg'
import SearchIcon from '@material-ui/icons/Search';
import ShoppingBasketIcon from '@material-ui/icons/ShoppingBasket';
import {Link} from 'react-router-dom'
import fire from '../config'
import { useStateValue } from './StateProvider';

function Header({user}) {

    const [{basket},dispatch]=useStateValue();

    const handleLogout=()=>{
        if(user)
        fire.auth().signOut();
    }

    return (
        <div className="header">
            <Link to="/">
            <img className="header_logo" src={navykartLogo} alt="Logo" />
            </Link>

            <div className="header_search">
                <input className="header_input" type="text" />
                <SearchIcon className="header_icon" />
            </div>
            <div className="header_nav">

                <div className="header_signIn">            {/* Conditional
operator */}

                    <div className="header_signIn">
                        <Link to={!user && '/login'}
className="header_linkToLogin ">
                            <div className="header_options"
onClick={handleLogout} >
                                <span className="header_lineOne">{user ?
fire.auth().currentUser?.email : 'Hello, Guest'}</span>
                                <span className="header_lineTwo">{user ?
fire.auth().currentUser?.displayName : 'Sign In'}</span>
                            </div>
                        </Link>
```

```
                        {/*the '?' in "auth().currentUser.displayname" is because
initially it will show error that "display name can not be empty"
                            while it have data but it takes times to fetch it so
'?' makes it async fuction which means wait for some time till it fetch the
data.  }
                        </Link>
                        */}
                        {user ?
                            <div className="header_SignInMenu">
                                <button className="header_signInmenuBtn"
onClick={handleLogout}>Logout</button>
                            </div> :
                            <div className="header_SignInMenu">
                                <Link to="/login">
                                    <button className="header_signInmenuBtn">Sign
in</button>
                                </Link>
                                <div className="header_SignInMenuText">New
Customer?
                                    <Link to="/signup"
className="header_SignInMenu_signupLink">
                                        <span
className="header_SignInMenu_signup"> Start here.</span>
                                    </Link>
                                </div>
                            </div>}
                    </div>


                <div className="header_arrowup"></div>     {/* for little
triangle near the signin button*/}
                </div>

                <div className="header_options">
                    <span className="header_lineOne">Returns</span>
                    <span className="header_lineTwo">& Orders</span>
                </div>

                <div className="header_options">
                    <span className="header_lineOne">Your</span>
                    <span className="header_lineTwo">Prime</span>
                </div>

                <Link to="/cart" className="header_linkToCart">
                <div className="header_optionCart">
                    <ShoppingBasketIcon className="header_cartLogo" />
                    <span className="header_lineTwo
header_cartCount">{basket?.length}</span>
                </div>
                </Link>
            </div>
        </div>
```

```
    )
}

export default Header
```

Home.css

```
.home {
  background-color: rgb(234, 237, 237);
}

.home_banner {
  width: 100%;
  mask-image: linear-gradient(
    to bottom,
    rgba(0, 0, 0, 1),
    rgba(0, 0, 0, 1),
    rgba(0, 0, 0, 0)
  );
  z-index: -1;
  margin-bottom: -150px;
}

.home_product {
  display: flex;
  z-index: 1;
  justify-content: center;
  margin-left: 5px;
  margin-right: 5px;
  margin-bottom: 20px;
}

@media only screen and (max-width: 550px) {
  .home_product {
    flex-direction: column;
  }
}
```

Home.js

```
import React from 'react'
import './Home.css'
import Keyboard from './Assets/Keyboard.jpg'
import Product from './Product'

import Samsung from './product/Samsung.jpg'
```

```
import Bag from './product/Bag.jpg'
import  Laptop from './product/Laptop.jpg'
import Earbud from './product/Earbud.jpg'
import AC from './product/AC.jpg'
import Display from './product/Display.jpg'

function Home() {

    return (
        <div className="home">

            <img className="home_banner" src={Keyboard} alt="Banner"/>

            <div className="home_product">
                <Product id={1} Description="LG 22 inch (55cm) IPS Monitor - Full
HD, IPS Panel with VGA, HDMI, DVI, Audio Out Ports - 22MP68VQ" price={24999}
rating={4} image={Display} />
                <Product id={2} Description=" Lenovo IdeaPad Flex 5i 10th Gen
Intel Core i3 14 inch Full HD IPS 2-in-1 Convertible Laptop (4GB/256GB
SSD/Windows 10/MS Office 2019/Graphite Grey/1.5Kg), 81X100NCIN " price={59999}
rating={5} image={Laptop}/>
            </div>

            <div className="home_product">
                <Product id={3} Description="POLESTAR XPLORE 55 ltrs with Rain
Cover Rucksack Hiking Backpack" price={4999} rating={4} image={Bag}/>
                <Product id={4} Description="boAt Airpodes 121v2 TWS Earbuds with
Bluetooth V5.0, Immersive Audio, Up to 14H Total Playback, Instant Voice
Assistant,(Active Black)" price={1999} rating={3} image={Earbud}/>
                <Product id={5} Description="Samsung Galaxy M31s (Space Black,
8GB RAM, 128GB Storage) " price={21999} rating={4} image={Samsung}/>
            </div>

            <div className="home_product">
                  <Product id={6} Description="Blue Star 1.5 Ton 3 Star Split AC
(Copper, 2018 Model, FS318AATX, White)" price={34999} rating={3} image={AC}/>
            </div>

        </div>
    )
}

export default Home
```

Login.css

```
.login_form{
  margin-top: 50px;
}
```

```css
.login_img {
  object-fit: contain;
  width: 170px;
  margin-top: -30px;
  margin-bottom: -30px;
}

.login {
  display: flex;
  flex-direction: column;
  align-items: center;
}

.login_form {
  display: flex;
  flex-direction: column;
  width: 300px;
  height: fit-content;
  border: 1px solid lightgray;
  padding: 0px 20px;
  background-color: white;
}

.login_form > h2 {
  font-weight: 500;
  font-size: 27px;
  margin-bottom: 7px;
}

.login_formOption {
  font-weight: 600;
  font-size: 13px;
  margin-top: 7px;
  margin-bottom: 5px;
}

.login_formText {
  width: 95%;
  padding: 7px;
  border-radius: 2px;
  border: 1px solid gray;
  margin-bottom: 5px;
  border-radius: 2px;
  outline: none;
}

.login_formText:focus {
  border: 1px solid #febd69;
  outline: 2px solid #febd69;
}

.login_formText[type="text"] {
```

```
  font-weight: lighter;
  font-size: 13px;
}

.login_login {
  margin-top: 15px;
  padding: 7px;
  border-radius: 2px;
  background-color: #febd69;
  border: 1px solid #846a29;
  cursor: pointer;
  color: #111;
  font-size: 15px;
  outline: none;
  text-align: center;
}

.login_login:hover,
.login_login:focus {
  outline: 2px solid #febd69;
  background-color: #eeb160;
}

.login_form > p {
  font-size: 12px;
  font-weight: 400;
  margin: 25px 0px;
  margin-bottom: 40px;
}

.login_form_pSpan {
  color: #0066c0;
  cursor: pointer;
}

.login_form_pSpan:hover {
  text-decoration: underline;
}

.login_line {
  margin-top: 40px;
  border-bottom: 1px solid rgb(192, 188, 188);
  width: 350px;
  margin-bottom: -11px;
}

.login_linePara {
  font-size: small;
  font-weight: 300;
  background-color: white;
  margin-bottom: 15px;
  color: rgb(99, 95, 95);
```

```css
}

.login_register {
  width: 350px;
  padding: 7px;
  text-align: center;
  border-radius: 2px;
  border: 1px solid rgb(71, 69, 69);
  color: rgb(31, 29, 29);
  outline: none;
  margin-bottom: 30px;
}

.login_register:hover,
.login_register:focus {
  background-color: rgb(224, 218, 218);
  outline: 2px solid rgb(211, 204, 204);
}

/* forget pasword */
.login_forgot:hover {
  text-decoration: underline;
}

.login_forgot {
  text-decoration: none;
  color: #0066c0;
}

.forgot {
  margin-top: -4px;
  align-self: flex-end;
  font-weight: 400;
}
```

Login.js

```jsx
import React ,{useState} from 'react'
import logo from './Assets/navykart-logo.jpg'
import './Login.css'
import {Link , useHistory} from 'react-router-dom'
import fire from '../config'

function Login() {

    const [email,setEmail]=useState("")
    const [password,setPassword]=useState("")
```

```
    const history = useHistory();

    const handleLogin=(e)=>{
        e.preventDefault();
        fire.auth().signInWithEmailAndPassword(email,password)
        .then((res)=>{
            if(res.user.emailVerified){
                history.push("/")
            }else{
                fire.auth().currentUser.sendEmailVerification().catch(err=>
alert(err))
                alert("Email not verified, Check Your email to verify & Login")
            }
        })
        .catch((err)=> alert(err))
    }

    return (
        <div className="login">
             <Link to="/">
            <img className="login_img" src={logo} alt=""/>
            </Link>
            <div className="login_form" >
                <h2>Login</h2>
                <div className="login_formOption ">Email Address</div>
                <input type="text" value={email} className="login_formText"
onChange={(e)=> setEmail(e.target.value)} />

                <div className="login_formOption">Password</div>
                <input type="password" value={password}
className="login_formText" onChange={(e)=> setPassword(e.target.value)} />

                <div className="login_formOption forgot">
                    <Link to="/resetpassword" className="login_forgot">Forgot
password?</Link>
                </div>

                <button className="login_login"
onClick={handleLogin}>Login</button>

                <p>By continuing, you agree to Navykart's <span
className="login_form_pSpan">Conditions of Use</span> and <span
className="login_form_pSpan">Privacy Notice.</span></p>
            </div>

            <span className="login_line"></span>
            <span className="login_linePara"> New to Navykart?</span>

            <Link to="/signup">
            <button className="login_register">Create your Navykart
account</button>
```

```
            </Link>

        </div>
    )
}

export default Login
```

Product.css

```
.product {
  display: flex;
  flex-direction: column;
  justify-content: flex-end;
  align-items: center;
  background-color: white;
  z-index: 1;
  margin: 0px 20px;
  padding: 20px;
  width: 100%;
  max-height: 400px;
  min-width: 100px;
}

.product_description {
  margin-bottom: 5px;
  overflow: hidden;
}

.product_image {
  max-height: 200px;
  object-fit: contain;
  width: 100%;
  margin-bottom: 10px;
  margin-top: 5px;
}
.product_price {
  margin-top: 3px;
}

.product > button {
  cursor: pointer;
  color: #111;
  background-color: #f0c14b;
  border: 1px solid #846a29;
  border-radius: 2px;
  margin-top: 3px;
  padding: 5px;
}
```

```css
@media only screen and (max-width: 550px) {
  .product {
    margin: 20px auto;
    margin-top: 20px;
    width: 80%;
  }
}
```

Product.js

```jsx
import React from 'react'
import './Product.css'
import { useStateValue } from './StateProvider';

function Product(props) {
    const {id,Description ,price,rating,image}=props ;

    const [{basket},dispatch]=useStateValue();

    const addToBasket=()=>{
        //dispatch the data item into data layer
        dispatch({
            type: 'ADD_TO_BASKET',
            item : {
                id:id,
                Description:Description ,
                price:price,
                rating:rating,
                image:image
            }
        })
    }

    return (
        <div className="product ">

            <div className="product_description">
                {Description}
            </div>

            <strong className="product_price">
                ₹{price}
            </strong>


             <div className="product_rating">
                 {Array(rating).fill().map((_,i)=>(
                    <span>☆</span>
```

```
                ))}
            </div>

            <img className="product_image" src={image} alt="product"/>

            <button onClick={addToBasket} >Add To Cart</button>
        </div>
    )
}

export default Product
```

Reducer.js

```
export const initialState ={
    basket : [],
}

const reducer = (state,action) => {
    switch(action.type){
        case 'ADD_TO_BASKET' :
            return {
                ...state,
                basket:[...state.basket,action.item],
            };
        case 'REMOVE_FROM_BASKET':
            return {

            }
        default :
            return state;
    }
};
export default reducer;
```

ResetPassword.css

```
.resetPassword_form{
  margin-top: 50px;
}
.resetPassword_img {
  object-fit: contain;
  width: 170px;
  margin-top: -35px;
  margin-bottom: -35px;
}

.resetPassword {
```

```
  display: flex;
  flex-direction: column;
  align-items: center;
}

.resetPassword_form {
  display: flex;
  flex-direction: column;
  width: 300px;
  height: fit-content;
  border: 1px solid lightgray;
  padding: 0px 20px;
  background-color: white;
}

.resetPassword_form > h2 {
  font-weight: 500;
  font-size: 27px;
  margin-bottom: 5px;
}

.resetPassword_para {
  font-size: 13px;
  font-weight: 400;
  margin-top: 8px;
}

.resetPassword_formOption {
  font-weight: 600;
  font-size: 13px;
  margin-top: 7px;
  margin-bottom: 5px;
}

.resetPassword_formText {
  width: 95%;
  padding: 7px;
  border-radius: 2px;
  border: 1px solid gray;
  margin-bottom: 5px;
  border-radius: 2px;
  outline: none;
}

.resetPassword_formText:focus {
  border: 1px solid #febd69;
  outline: 2px solid #febd69;
}

.resetPassword_formText[type="email"] {
  font-weight: lighter;
  font-size: 13px;
```

```
}

.resetPassword_resetPassword {
  margin: 15px 0px;
  margin-bottom: 45px;
  padding: 7px;
  border-radius: 2px;
  background-color: #febd69;
  border: 1px solid #846a29;
  cursor: pointer;
  color: #111;
  font-size: 15px;
  outline: none;
  text-align: center;
}

.resetPassword_resetPassword:hover,.resetPassword_resetPassword:focus {
  outline: 2px solid #febd69;
  background-color: #eeb160;
}
```

ResetPassword.js

```
import React ,{useState}from 'react'
import logo from './Assets/navykart-logo.jpg'
import './ResetPassword.css'
import {Link,useHistory} from 'react-router-dom'
import fire from '../config'

function ResetPassword() {

    const [email,setEmail]=useState('')
    const history =useHistory();

    const handleResetPassword=()=>{
        fire.auth().sendPasswordResetEmail(email).then
                  (()=>
                   alert("Check your email to reset password & Sign IN "),
                   history.push("/login")
                   )
                  .catch(err=> alert(err))
    }

    return (
        <div className="resetPassword">
            <Link to="/">
            <img className="resetPassword_img" src={logo} alt=""/>
            </Link>

            <div className="resetPassword_form" >
```

```
                <h2>Password assistance</h2>

                <p className="resetPassword_para">Enter the email address
associated with your Navykart account.</p>

                <div className="resetPassword_formOption">Registered email</div>
                <input type="email"  className="resetPassword_formText"
onChange={(e)=> setEmail(e.target.value)} />

                <button className="resetPassword_resetPassword"
onClick={handleResetPassword}>Continue</button>
            </div>

        </div>
    )
}

export default ResetPassword
```

signup.css

```
.signup_form{
  margin-top: 50px;
}
.signup_img {
  object-fit: contain;
  width: 170px;
  margin-top: -35px;
  margin-bottom: -35px;
}

.signup {
  display: flex;
  flex-direction: column;
  align-items: center;
  background-color: white;
}

.signup_form {
  display: flex;
  flex-direction: column;
  width: 300px;
  height: fit-content;
  border: 1px solid lightgray;
  padding: 0px 20px;
  margin-bottom: 30px;
}

.signup_form > h2 {
  font-weight: 500;
```

```
  font-size: 27px;
  margin-bottom: 5px;
}

.signup_formOption {
  font-weight: 600;
  font-size: 13px;
  margin-top: 7px;
  margin-bottom: 5px;
}

.signup_formText {
  width: 95%;
  padding: 7px;
  border-radius: 2px;
  border: 1px solid gray;
  margin-bottom: 5px;
  border-radius: 2px;
  outline: none;
}

.signup_formText:focus {
  border: 1px solid #febd69;
  outline: 2px solid #febd69;
}

.signup_formText[type="text"] {
  font-weight: lighter;
  font-size: 13px;
}

.signup_signup {
  margin-top: 2px;
  padding: 7px;
  border-radius: 2px;
  background-color: #febd69;
  border: 1px solid #846a29;
  cursor: pointer;
  color: #111;
  font-size: 15px;
  outline: none;
  text-align: center;
}

.signup_signup:hover,
.signup_signup:focus {
  outline: 2px solid #febd69;
  background-color: #eeb160;
}

.signup_para {
  margin: 20px 0px;
```

```
}

.signup_para > p {
  font-size: 12px;
  font-weight: 400;
  margin: 0px;
}

.signup_borderline {
  border: 1px solid rgb(202, 196, 196);
  width: 300px;
  margin-top: 25px;
}

.signup_login_line {
  margin-top: 20px;
  font-size: 13px;
  font-weight: 400;
  margin-bottom: 15px;
}

.signup_login {
  color: #0066c0;
  cursor: pointer;
}

.signup_login:hover {
  color: red;
  text-decoration: underline;
}

.signup_linkToLogin {
  text-decoration: none;
}
```

SignUp.js

```
import React ,{useState} from 'react'
import logo from './Assets/navykart-logo.jpg'
import './signup.css'
import {Link, useHistory} from 'react-router-dom'
import fire from '../config'

function SignUp() {

    const [name,setName]=useState('')
    const [email,setEmail]=useState('')
    const [password,setPassword]=useState('')
    const [confirmPassword,setConfirmPassword]=useState('')
```

```
    const history = useHistory();

    const handleSignUp=(e)=>{
        e.preventDefault();
        if(name!==''){
            if(password===confirmPassword){
                fire.auth().createUserWithEmailAndPassword( email,password)
                    .then(() => {
                        const userData = fire.auth().currentUser;
                        userData.updateProfile({ displayName: name })    /*the
name we type in input tag can be passed to firebase user data*/
                        userData.sendEmailVerification().catch(err=>alert(err))
                        alert("Check Your email")
                        history.push("/login")
                    })
                    .catch((err)=> alert(err))
            }else{ alert("Password doesn't match")}
        }else{ alert("Please enter Name")}
    }

    return (
        <div className="signup">

            <Link to="/">
            <img className="signup_img" src={logo} alt=""/>
            </Link>

            <div className="signup_form" >
                <h2>Create Account</h2>

                <div className="signup_formOption">Your Name</div>
                <input type="text" value={name} className="signup_formText"
onChange={(e)=> setName(e.target.value)} required/>

                <div className="signup_formOption">Email</div>
                <input type="email" value={email} className="signup_formText"
onChange={(e)=> setEmail(e.target.value)} />

                <div className="signup_formOption">Password</div>
                <input type="password" value={password} placeholder="At least 6
characters" className="signup_formText" onChange={(e)=>
setPassword(e.target.value)} />

                <div className="signup_formOption">Confirm Password</div>
                <input type="password" value={confirmPassword} placeholder="must
be same as password" className="signup_formText" onChange={(e)=>
setConfirmPassword(e.target.value)} />

                <div className="signup_para">
                    <p>We will send you a text to verify your phone.</p>
                    <p>Message and Data rates may apply.</p>
```

```
                </div>

                <button className="signup_signup"
onClick={handleSignUp}>Continue</button>

                <div className="signup_borderline"></div>

                <div className="signup_login_line">Already have an account?
                    <Link to="/login" className="signup_linkToLogin">
                        <span className="signup_login">Sign in</span>
                    </Link>
                </div>

            </div>

        </div>
    )
}

export default SignUp
```

StateProvide.js

```
import React,{createContext,useReducer,useContext} from 'react'

export const StateContext = createContext();   //prepares the dataLayer

//wrap our app and provide the data layer
export const StateProvider = ({ reducer, initialState,children})=> {
    return (
        <StateContext.Provider value={useReducer(reducer,initialState)}>
            {children}
        </StateContext.Provider>
    )
}

//pull information from data layer
export const useStateValue=()=> useContext(StateContext);
```

SubTotal.css

```
.subtotal {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  width: 300px;
  height: 100px;
```

```
  padding: 20px;
  background-color: #f3f3f3;
  border: 1px solid #dddddd;
  border-radius: 3px;
  margin-left: 20px;
}

.subtotal > small {
  display: flex;
  align-items: center;
}
.subtotal > button {
  background-color: #febd69;
  border: 1px solid #846a29;
  outline: none;
  color: #111;
  border-radius: 2px;
  padding: 5px;
  width: 100%;
  cursor: pointer;
  margin-top: 10px;
}
.subtotal > button:hover,
.subtotal > button:focus {
  outline: 2px solid #febd69;
  background-color: #eeb160;
}

@media only screen and (max-width: 550px) {
  .subtotal {
    width: 100px;
    height: 200px;
  }
}
```

SubTotal.js

```
import React from 'react'
import "./Subtotal.css";
import Currencyformat from 'react-currency-format'
import { useStateValue } from './StateProvider';
import {useHistory} from 'react-router-dom'

function Subtotal() {

    const [{basket}]=useStateValue();
    const history=useHistory();

    const getBasketTotal=(basket)=>{
```

```
        let sum=0;
        basket.map(item=>{
            sum=sum+item.price;
            })
        return sum;
    }

    return (
        <div className="subtotal">
            <Currencyformat
                renderText={(value)=>(
                    <>
                        <p>Subtotal ({basket?.length}
items):<strong>{value}</strong></p>
                        <small>
                            <input type="checkbox" />
                            This order contains a gift
                        </small>
                        <button onClick={(e)=> history.push("/checkout")}>Proceed
to Checkout</button>
                    </>
                )}
                decimalScale={2}
                value={getBasketTotal(basket)}
                displayType={"text"}
                thousandSeparator={true}
                prefix={"₹"}
                suffix={"/-"}
            />
        </div>
    )
}

export default Subtotal
```

App.css

```
.App {
  min-width: 550px;
}
```

App.js

```
import React ,{useState,useEffect} from 'react'
import "./App.css";
import Header from './components/Header'
```

```
import Home from './components/Home'
import Login from './components/Login'
import {BrowserRouter as Router, Route ,Switch} from "react-router-dom"
import SignUp from './components/SignUp'
import Cart from './components/Cart'
import fire from './config'
import ResetPassword from './components/ResetPassword'
import CheckOut from './components/CheckOut'

function App() {
const [user,setUser]=useState(null)

useEffect(() => {
  fire.auth().onAuthStateChanged(user=>{
    if(user){
      if(user?.emailVerified){
        setUser(user)
      }else{
        setUser(null)
      }
    }else{
      setUser(null)
    }
  })
}, [])

  return (

      <Router>
        <div className="App">
        <Switch >

          <Route path="/cart">
            <Header user={user}/>
            <Cart/>
          </Route>

          <Route path="/login">
            <Login />
          </Route>

          <Route path="/signup">
            <SignUp  />
          </Route>

          <Route path="/resetpassword">
            <ResetPassword/>
          </Route>

          <Route path="/checkout">
            <Header user={user}/>
            <CheckOut/>
```

```
          </Route>

          <Route path="/">
            <Header user={user}/>            {/* for sending the data to header
page*/}
            <Home />
          </Route>

        </Switch>

      </div>
      </Router>
  );
}

export default App;
```

Config.js

```
import firebase from 'firebase/app'
import 'firebase/auth'

const firebaseConfig = {
  apiKey: "AIzaSyDy_jAyI6LOiFBqCa11EOnhlJ0K_pc6TTk",
  authDomain: "navykart.firebaseapp.com",
  projectId: "navykart",
  storageBucket: "navykart.appspot.com",
  messagingSenderId: "698305334130",
  appId: "1:698305334130:web:bde026ef6457c994d72beb"
};

const fire = firebase.initializeApp(firebaseConfig);

export default fire;
```

Index.css

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", "Roboto", "Oxygen",
    "Ubuntu", "Cantarell", "Fira Sans", "Droid Sans", "Helvetica Neue",
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

code {
```

```
  font-family: source-code-pro, Menlo, Monaco, Consolas, "Courier New",
    monospace;
}
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
//added for context api
import { StateProvider } from './components/StateProvider';
import reducer ,{initialState} from './components/Reducer'

ReactDOM.render(
  <React.StrictMode>
    <StateProvider initialState={initialState} reducer={reducer}>
      <App />
    </StateProvider>
  </React.StrictMode>,
  document.getElementById('root')
);
```

6. Test Cases

Test Cases for Home page:

- Verify that home page is displayed after login or not.
- Verify that User name is displayed on homepage or not.
- Verify that featured products are present on home page or not.
- Verify that Search functionality is present on home page or not.
- Verify the home page of application on different browsers.
- Verify the alignment on the home page.
- Verify that products displayed on home page are clickable or not.

- Verify that Products displayed on home page are categorised or not.

Test Cases for Product Search Functionality:

- Verify that the search field accepts alphabets numbers or symbols.

- Verify that after entering search text and clicking on search icon, the search should work.

- Verify that the search results should be as per the search query.
- Verify that user should be able to search based on product name, brand name or product
- specification.

- Verify that filter should be present for filtering the search results bases on Brand, Price, reviews or ratings.

- Verify that sorting options should be present on search results page.

Test Cases for Product Details page:

- Verify that the images of product are displayed correctly or not.

- Verify that the price of product is displayed or not.

- Verify that product reviews are mentioned or not.

Test Cases for Cart page:

- Verify that when user clicks on Add to Cart, then the product should be moved to cart.

- Verify that user is able to continue shopping after adding items to cart.

- Verify that the quantity of item should be incremented if user adds same item in cart again.

- Verify that the items in cart should be present if user logs out and logs in again.

Bibliography

https://www.google.co.in/
https://webandcrafts.com/blog/scope-of-ecommerce/
https://astischool.com/technology-and-computing/feasibility-study-of-online-shopping-article/
https://www.google.com/search?q=add+to+cart+flowchart&tbm=isch&ved=2ahUKEwiOl-Hmha31AhWHyKACHffYBTsQ2-cCegQIABAA&oq=add+to+cart+flow&gs_lcp=CgNpbWcQARgAMgUIABCABDIECAAQGDIECAAQGDoECAAQQzoICAAQgAQQsQM6CwgAEIAEELEDEIMBOgYIABAFEB46BggAEAgQHlCYDFi0N2DkSWgAcAB4AIABkwGIAf0KkgEEMC4xMZgBAKABAaoBC2d3cy13aXotaW1nwAEB&sclient=img&ei=FjnfYY60IYeRg8UP97GX2AM&bih=609&biw=1280#imgrc=jNRMeS4wpHFL7M