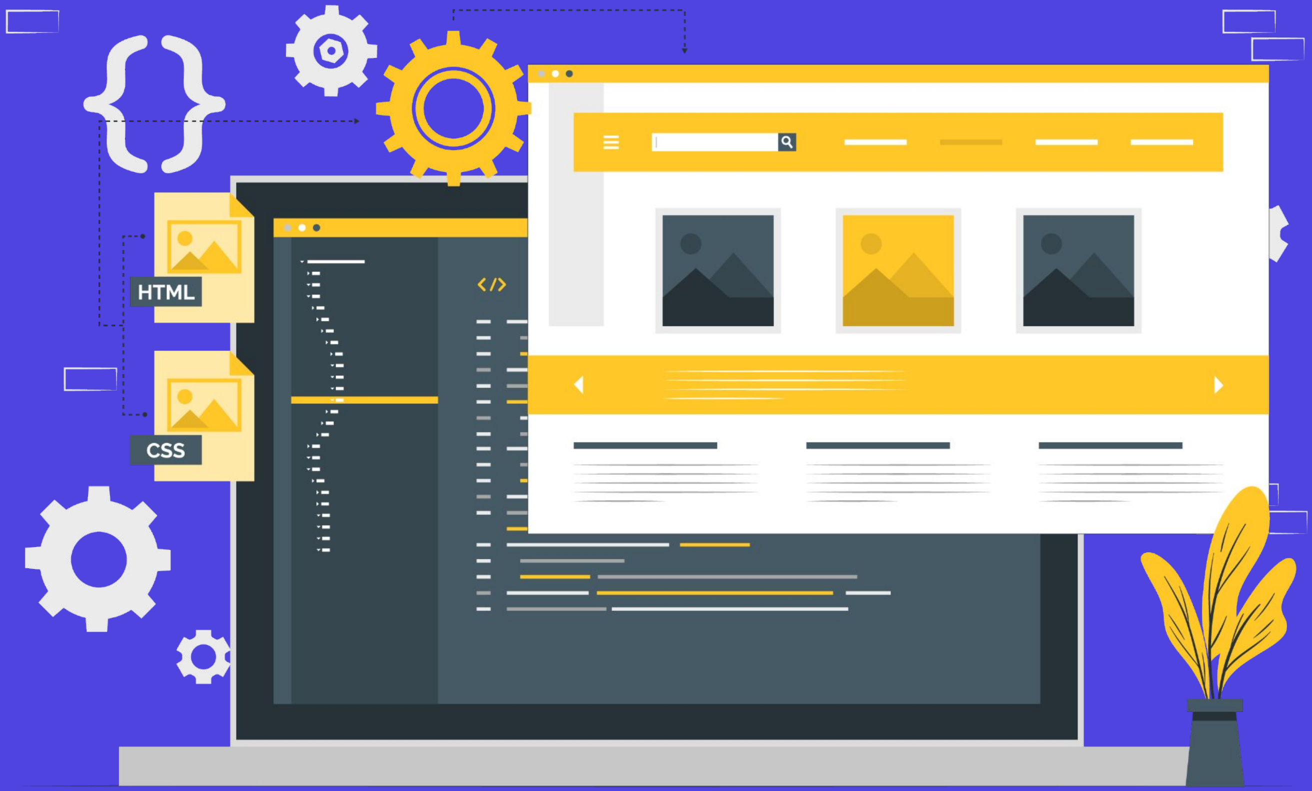
# Introduction to E-Commerce Web Application



Relevel
by Unacademy

E-commerce website plays an important role in development of any sort of business bet it small scale of large scale. In today's convenience-based society, people tend to remotely shop rather than going to local stores for buying items. E-commerce concept is rapidly growing as it has brought new ideas and opportunities in the field of business. E-commerce attracts consumers by providing services 24 hour with wide variety of product.

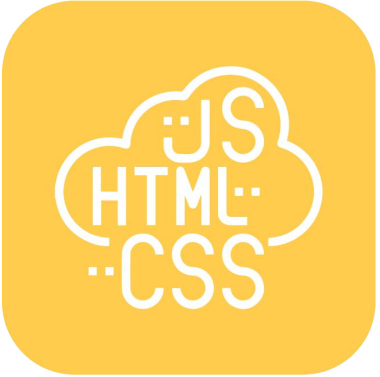
# App Introduction: (15 minutes)

We will be creating an online E-commerce application using HTML, CSS, JavaScript and some advanced tools and technologies. In which customers can place their orders from the shop of various categories such as electronic gadgets, mobiles, appliances and groceries etc.

Tools and technologies:

## HTML/HTML5

HTML means Hypertext Markup Language. This language is used in creating web pages. This language also supports other languages such CSS, PHP, JAVASCRIPT, etc. in creating interactive and responsive pages on the pages.



## CSS

CSS is simply referred to as Cascading Style Sheets.CSS is used to define styles for web pages, including the design, layout, and variations in the display for different devices and screen sizes.

## JavaScript

JavaScript is a high-level language which could be used independently or inculcated into the webpage. It can be used to, handle requests and responses and also add dynamic behaviour and also store information on a website.

# Integrating with Other Libraries:

## Bootstrap:

Bootstrap gives us the ability to create responsive designs easily. It is a front-end framework for developing easier and faster web development. Bootstrap includes HTML and CSS related design templates which are used for forms, tables, modals, carousels navigation and many other, and also as JavaScript plugins.

Downloading Bootstrap:

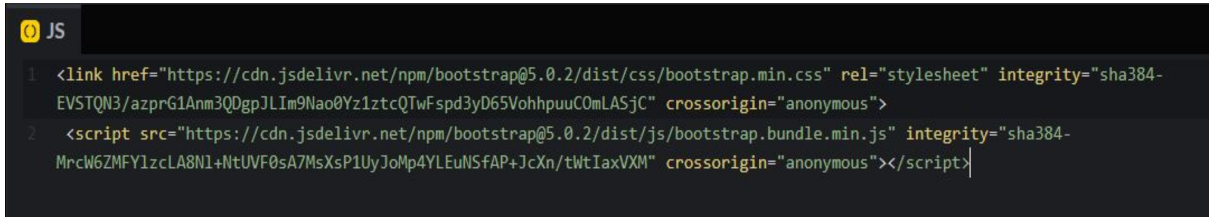
URL: https://getbootstrap.com

You can download bootstrap and add in your folder structure manually and import it in the code or alternatively you can use CDN.

## Content Delivery Network:

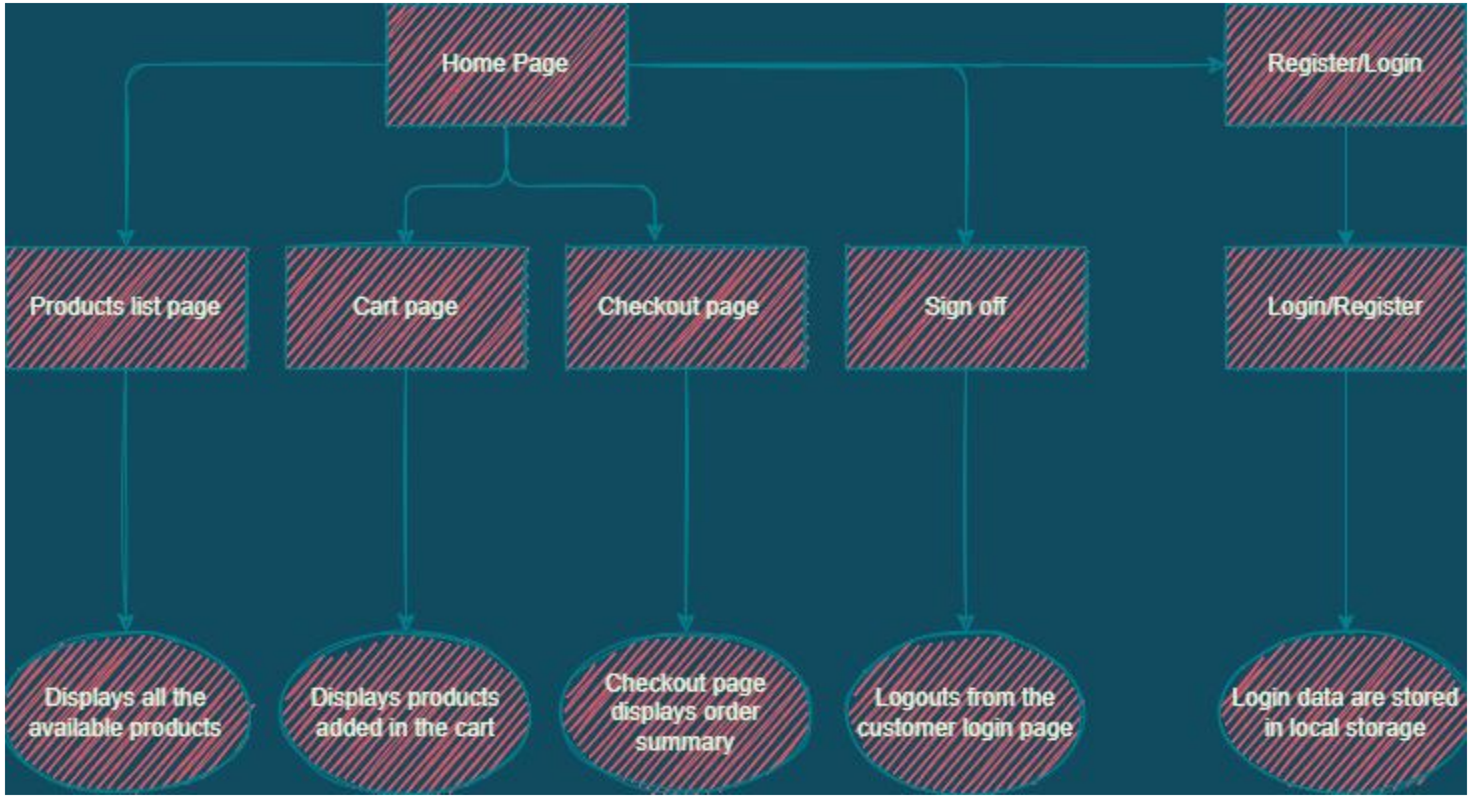
It is a distributed group of servers that work together to ensure the rapid delivery of Internet content. Helps you quickly transfer the  assets needed to load Internet content such as HTML pages, JavaScript files, stylesheets, images and videos.  Web traffic is delivered via a CDN that contains traffic from large websites such as Facebook and Amazon.

## Example:



```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>
```

## Project Structure:


Home Page
Register/Login
Products list page
Cart page
Checkout page
Sign off
Login/Register
Displays all the available products
Displays products added in the cart
Checkout page displays order summary
Logouts from the customer login page
Login data are stored in local storage

## Register/Login:

Signup: This refers to registering as a customer. The registered member has a lot of privileges associated with the shop when one becomes a customer.

Login: After the user has registered, the user becomes a customer, and he or she can log in with their personal information.

Home page: The customer can see all the products in the catalogue and able to look at the products on the homepage.

Products: Displays list of products of different categories.

Shopping cart: This refers to adding or removing products from a shopping cart.

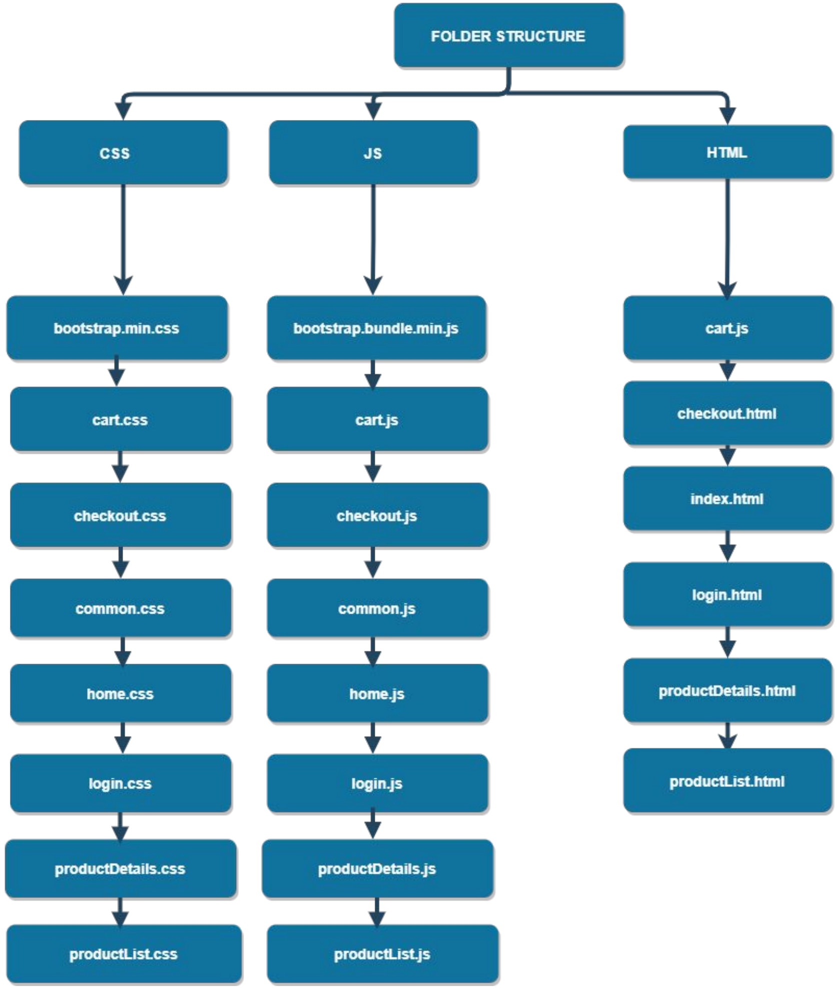
Checkout: Customer can check out by purchasing the product from the cart. Also this page displays the order summary of the products purchased by the customer.

Sign off: Customer can logout from the system.

# Application & Description:

We’ll create 5 pages – Register/Login page, home page, products page, shopping cart page and success page.

We can see the project’s folder structure:



FOLDER STRUCTURE

CSS

bootstrap.min.css

cart.css

checkout.css

common.css

home.css

login.css

productDetails.css

productList.css

JS

bootstrap.bundle.min.js

cart.js

checkout.js

common.js

home.js

login.js

productDetails.js

productList.js

HTML

cart.js

checkout.html

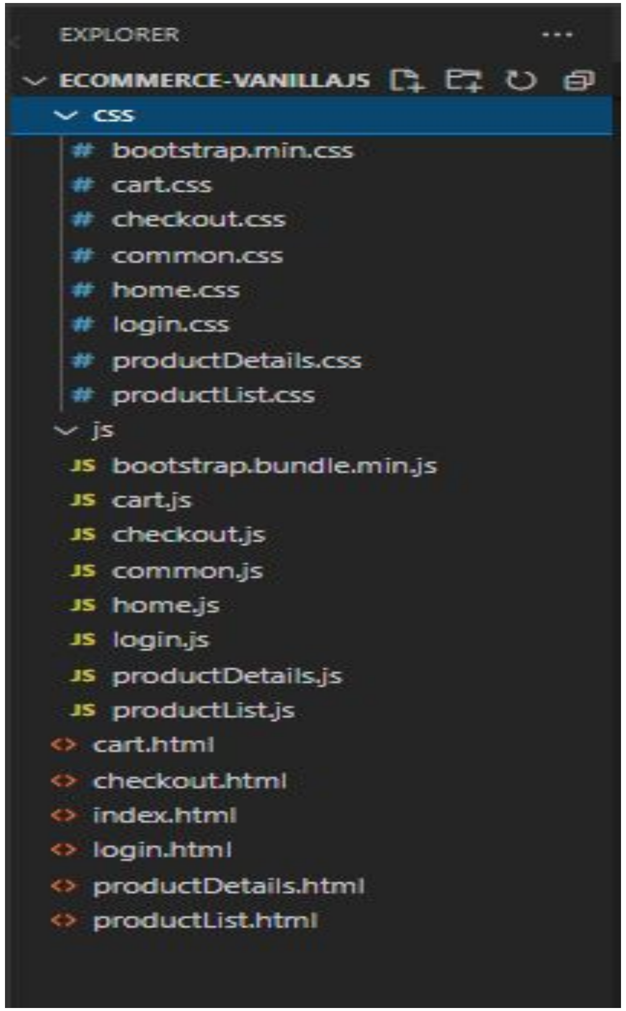
index.html

login.html

productDetails.html

productList.html

## Project’s folder structure:

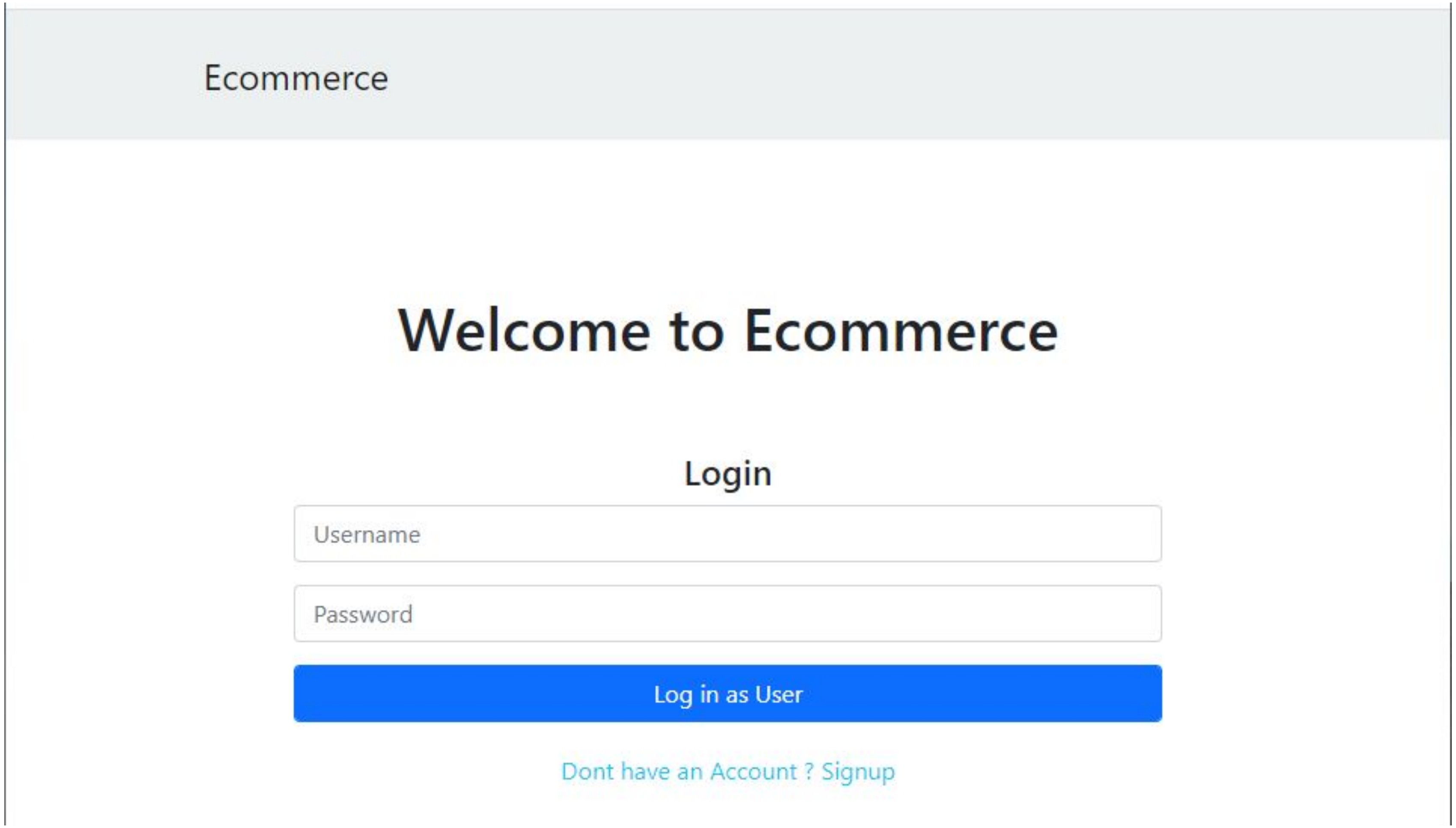


```
EXPLORER
ECOMMERCE-VANILLAJS
  css
    bootstrap.min.css
    cart.css
    checkout.css
    common.css
    home.css
    login.css
    productDetails.css
    productList.css
  js
    bootstrap.bundle.min.js
    cart.js
    checkout.js
    common.js
    home.js
    login.js
    productDetails.js
    productList.js
  cart.html
  checkout.html
  index.html
  login.html
  productDetails.html
  productList.html
```

# Implementation of functions:

## Register/Login Page:

- The web app has user signup and login page for the customers to register and login to the application. When the user registers, the information is stored to the local storage of the browser.
- The same page can be used for both login and signup functionality.
- Once the user is registered using the credentials username and password. They become customer and can able to login.



Ecommerce

Welcome to Ecommerce

Login

# Concepts Involved:

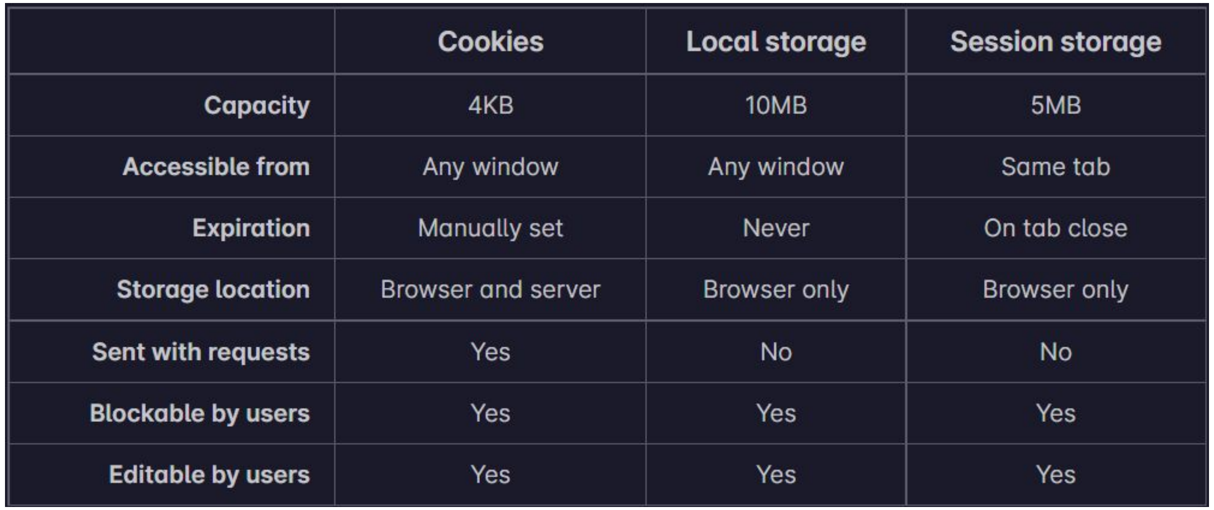
## Local Storage:

Local storage is used to store data for long time, which could be a day, a week, or even a year. The local storage only stores strings so, if you wish to store objects, lists, or another datatype, you must convert them into a string using JSON.stringify().

## Session Storage:

The session storage uses the session storage object to store data on a temporary basis, for a single browser window or tab. The data disappears when session ends that is when the user closes that browser window or tab.

Difference between local storage and session storage:



| | Cookies | Local storage | Session storage |
|---|---|---|---|
| **Capacity** | 4KB | 10MB | 5MB |
| **Accessible from** | Any window | Any window | Same tab |
| **Expiration** | Manually set | Never | On tab close |
| **Storage location** | Browser and server | Browser only | Browser only |
| **Sent with requests** | Yes | No | No |
| **Blockable by users** | Yes | Yes | Yes |
| **Editable by users** | Yes | Yes | Yes |

## LoginFn Method:

- loginFn method is defined to validate the user login credentials such as username and password.
- To select an HTML element, JavaScript most often uses the document.getElementById() method, in this case to obtain the username and password.
- The username and password is passed to the API and response is obtained using the fetch method.
- If the response is success, the username and userId generated from the backend is stored in the local storage for future use during the session.
- If the response is failure, the error message received is show in the webpage.

### LocalStorage method:

- The Window interface features a wide range of functions, constructors, objects, and namespaces. Window.localStorage is a read-only property that returns a reference to the local storage object used to store data that is only accessible to the origin that created it.

### Methods:

- setItem(): Add key and value to the  localStorage.
- getItem(): to get items from localStorage.
- removeItem(): Remove an item using a key from localStorage.
- clear(): Clear all the content in localStorage.
- key(): Pass a number(index) to retrieve the key of a localStorage.

# SignUpFn Method:

- signUpFn method is used to register a new user to the application.
- The username and password obtained similar to login function and passed to the API and response is obtained using the fetch method.
- If the response is success, the username and userId generated from the backend is stored in the local storage for future use during the session.
- If the response is failure, the error message received is show in the webpage.

# Homepage:

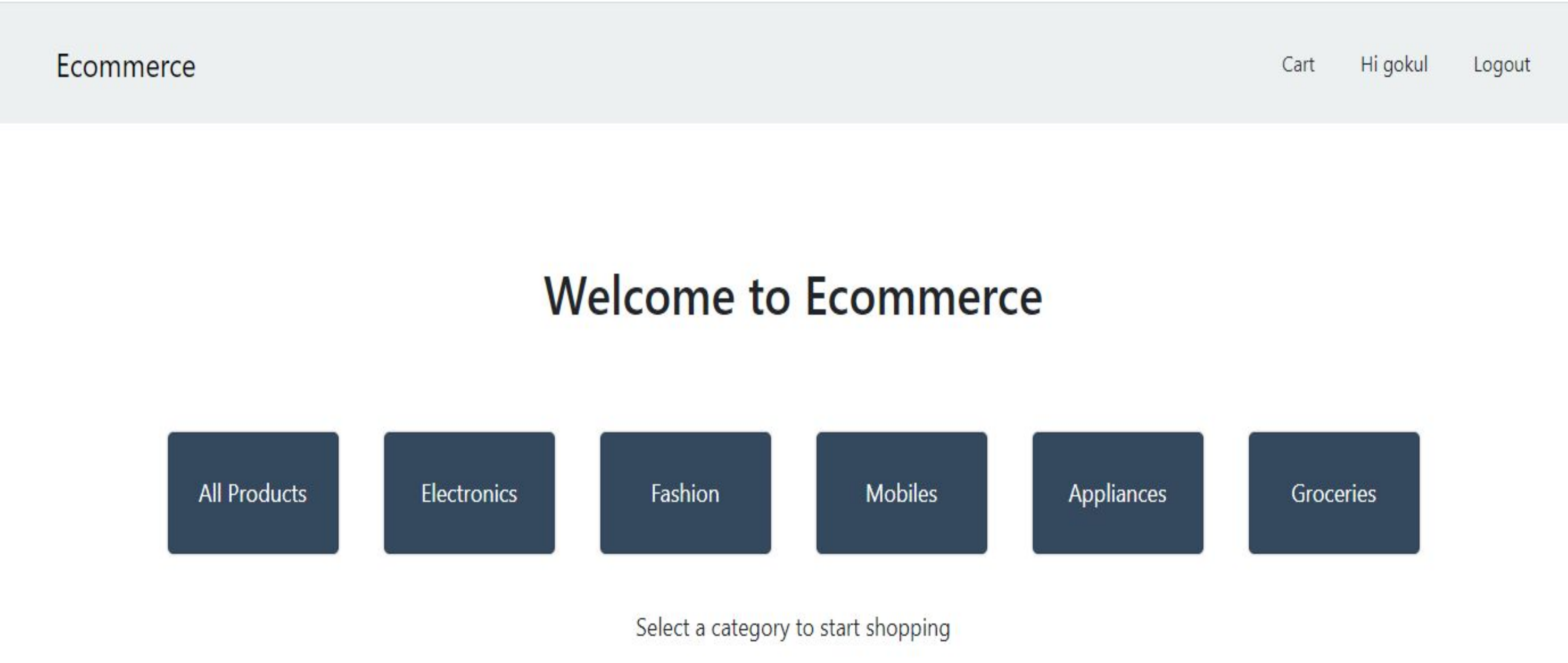
### home.js:

- This JavaScript code contains the code to display the category of products that are available in the E-commerce website.
- The categories available are fetched from the API call which made on load of the page, once the data is returned it is passed to renderCategories method.

### Render Categories Method:

- The categories to be displayed it is constructed dynamically using the data received in the page load API call.
- The category Id is embedded with anchor tag URL in the card, for future use.
- After construction of card, it displayed in the webpage using innerHtml method.
- Once the customer clicks on the category, the page is redirected to the productList.html using the id of the category embedded in the URL.

## Output:



Ecommerce

Welcome to Ecommerce

Select a category to start shopping

# ProductList page:

## ProductList.js:

This JavaScript code is to display the products that are available by providing filter options to filter out the products based on user preference in the E-commerce website.

- Initially page gets loaded with category selected in the previous page that is obtained from category id embedded in the URL.
- loadCategories method is used to fetch the categories from API and display the category as filter option to the user using renderCategory Method .
- User can select the category displayed in the filter section, so the products are displayed based on selection.
- The loadProducts method is to fetch the products based on the category. The category id is passed as a request parameter in the API.
- Initially, category id is extracted from the URL which is coming from previous page. Once the data is returned it is rendered as a card using the renderCategories method.

**Search Filter –**

when the user enters the product name the API call is triggered using the data and category id obtained, once the data is returned the UI is populated with relevant products by reusing rendering methods.

**Window.location.Search-**

The search property of the Location interface is a search string, also called a query string; that is, a USVString containing a '?' followed by the parameters of the URL.

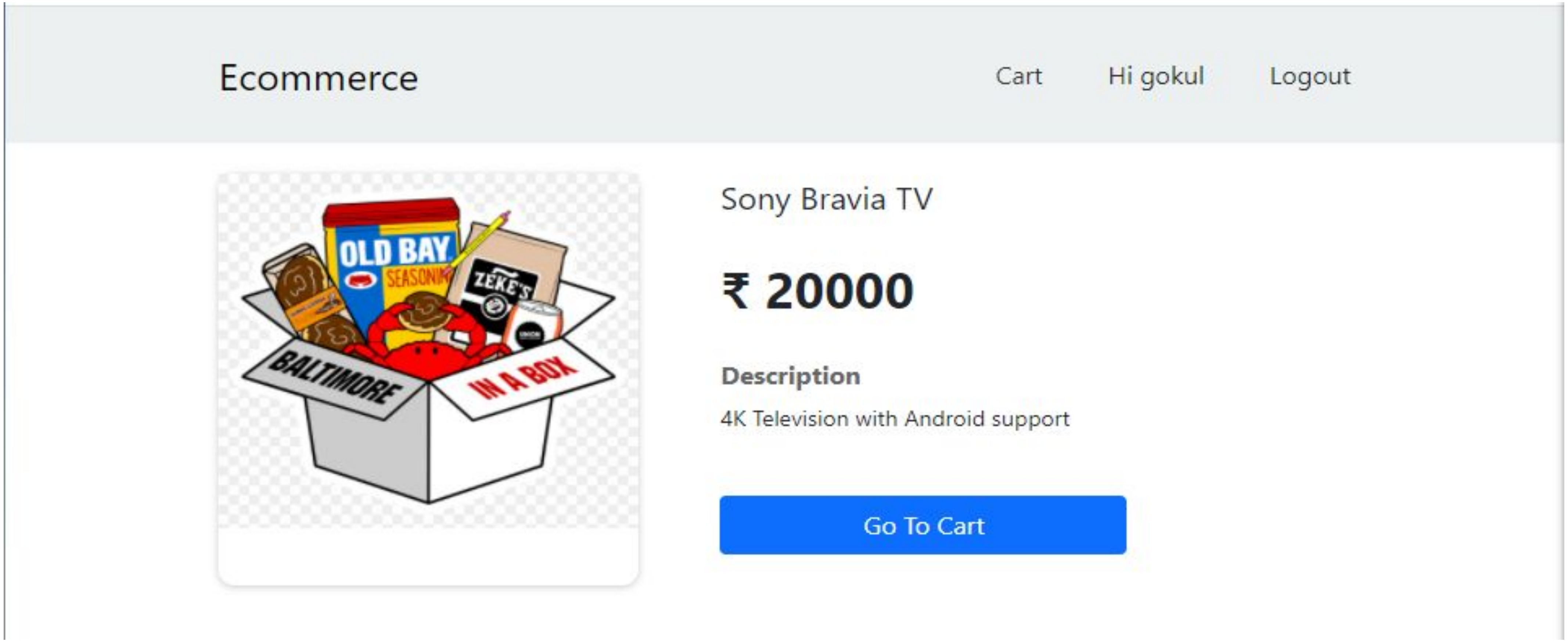
Modern browsers provide URLSearchParams and URL.searchParams to make it easy to parse out the parameters from the querystring.

## ProductDetails page:

### ProductDetails.js:

- After the customer clicks on the product, one can view the detailed product description on this page.
- On page load, the loadproductdetails method is called by passing the product id which is obtained from URL. The data is returned from API call the product details is constructed in renderProductDetails method and displayed in the UI.
- When the user clicks AddToCart button, the addToCartFn method is called the productId and userId which is in local storage is passed to API as request parameters.
- On success of the API call, the ‘addtocart’ button is replaced with ‘gotocart ‘button which link to cart.html.

## Output:



Ecommerce

Cart Hi gokul Logout

Sony Bravia TV

₹ 20000

Description

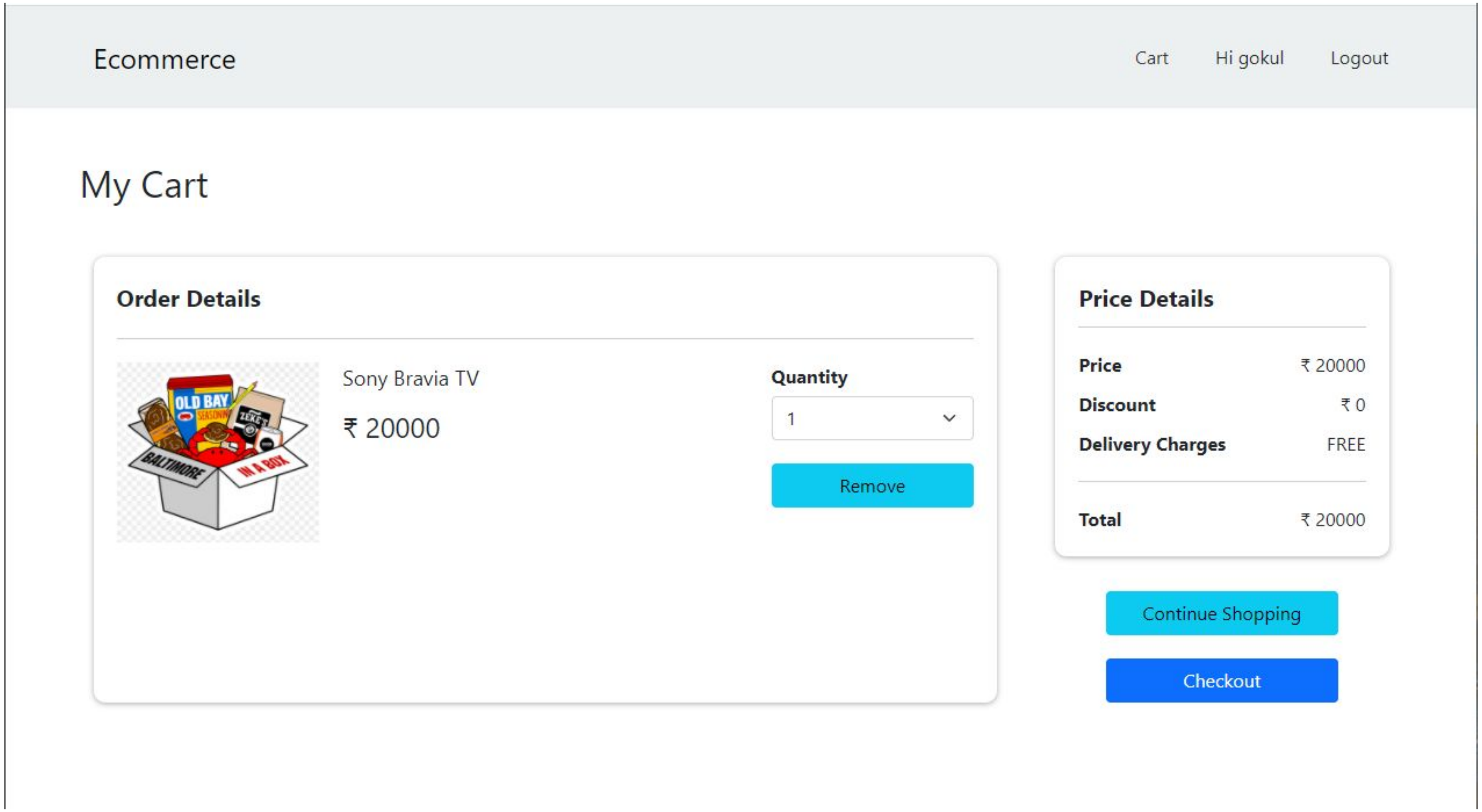
4K Television with Android support

# Shopping Cart:

## Cart.js:

- This page displays the products added in the cart with calculated price for each of the products along with the quantity.
- The initial cart details are obtained by calling the API on page load which can be seen in loadOrderDetails method.
- On Success the data is passed to renderOrderDetails method (it can be seen below) to construct the order details along with the price details and it is displayed in the UI.
- The customer can also remove the product from the cart using the remove button
- The products added in the cart can be incremented or decremented in the quantity in the cart using the quantity dropdown.
- The customer can add products to the shopping cart. They can also update the shopping cart from time to time until they are done shopping.
- After that, they can check out by clicking on the checkout button, or continue shopping by clicking continue shopping button.

## Output:



Ecommerce

My Cart

**Order Details**

Sony Bravia TV

₹ 20000

**Quantity**

1

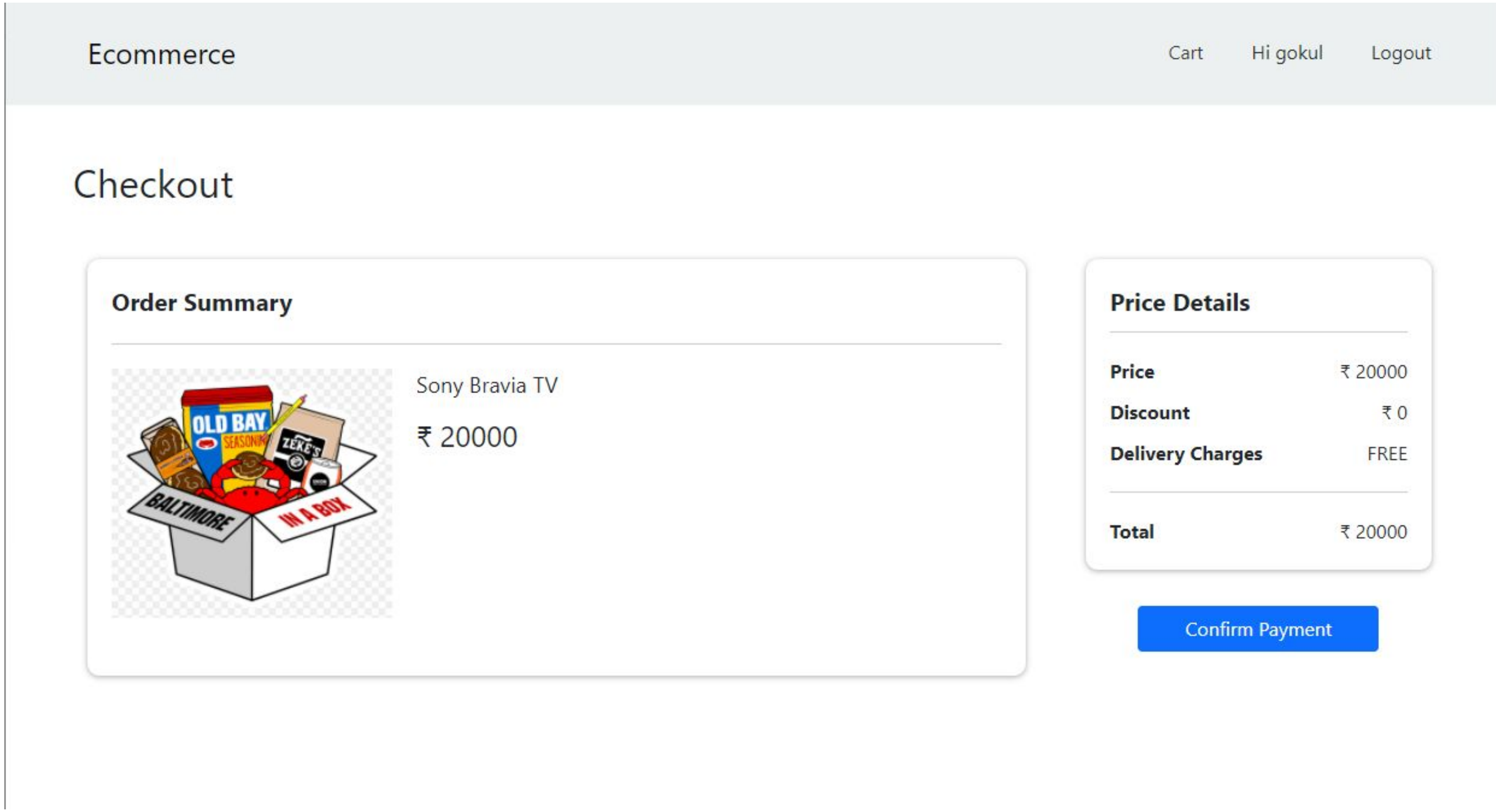
**Price Details**

| | |
|---|---|
| **Price** | ₹ 20000 |
| **Discount** | ₹ 0 |
| **Delivery Charges** | FREE |
| **Total** | ₹ 20000 |

## Checkout page:

- This page displays the order summary of the user.
- The initial order summary details of the user is fetched from the API call by passing the user id in the method loadOrderDetails on page Load.
- On success the returned data is passed to renderOrderDetails method to construct the order summary table as shown below.
- The order id is also received from the API response.
- Finally, the order is placed by the customer. And the customer can view the order details like products, delivery details and the pricing of the order.
- On clicking confirm payment button in the checkout page, the order along with the userId is passed to confirmPayment method
- On Success, the order which is confirmed will be displayed in the checkout page.

## Output:



# Checkout

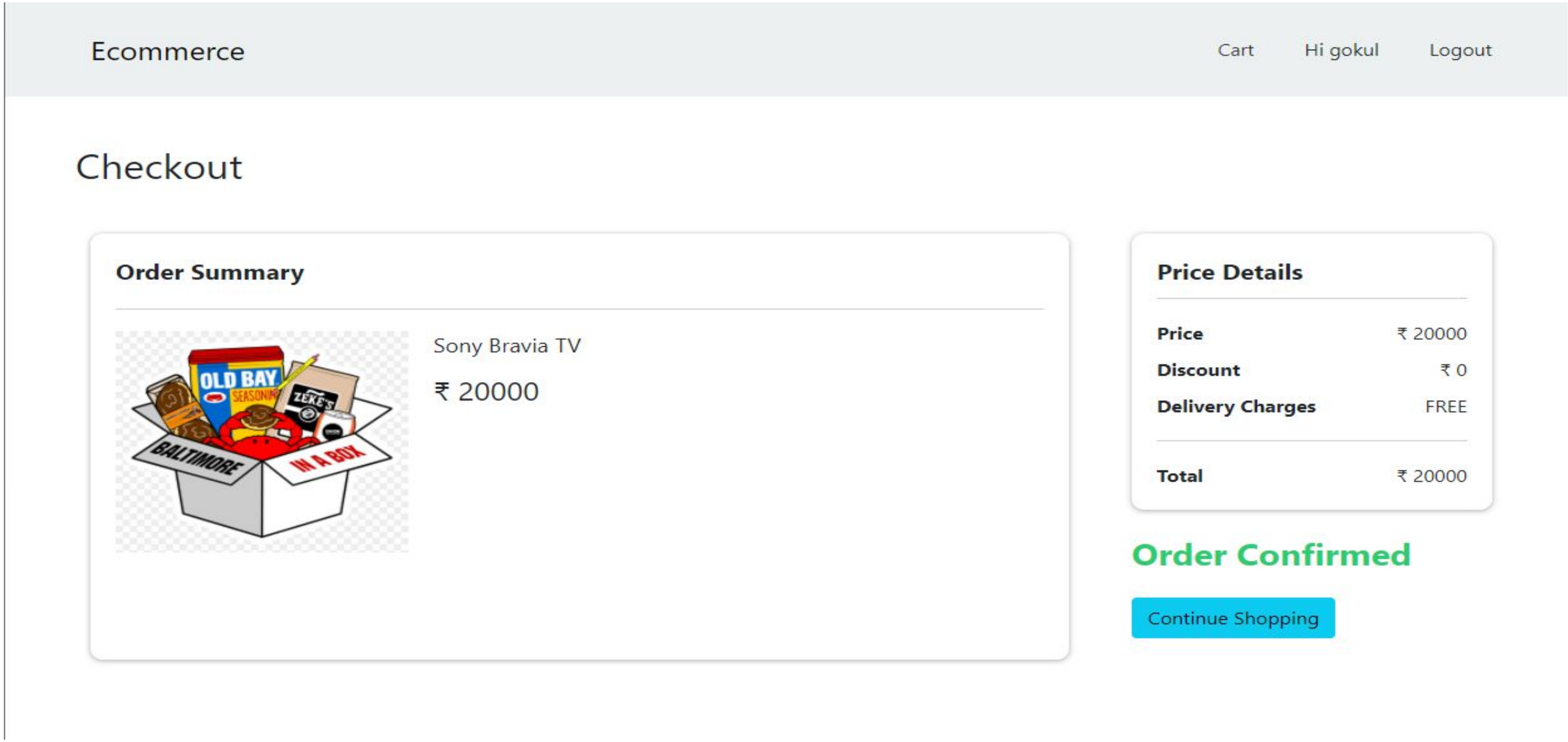
### Order Summary

Sony Bravia TV

₹ 20000

### Price Details

| | |
|---|---|
| **Price** | ₹ 20000 |
| **Discount** | ₹ 0 |
| **Delivery Charges** | FREE |
| **Total** | ₹ 20000 |



## Checkout

**Order Summary**

Sony Bravia TV

₹ 20000

**Price Details**

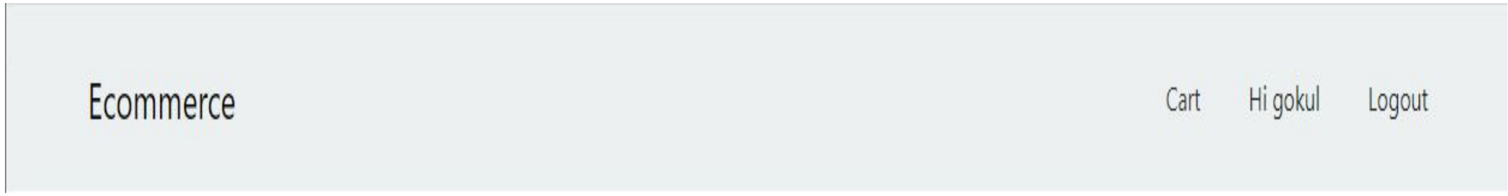
| | |
|---|---|
| **Price** | ₹ 20000 |
| **Discount** | ₹ 0 |
| **Delivery Charges** | FREE |
| **Total** | ₹ 20000 |

**Order Confirmed**

# Sign out:

- Once the customer sign offs from his profile, the page is redirected to the login page.

### Output:



## Conclusion:

The E-commerce shop was developed using HTML, CSS and JavaScript technology. Any consumer can browse products, add, replace or delete a product from the cart. The consumer can log in, with his information such as username and password. After login, the user can see the product in the cart and proceed onwards to checkout.

**Code link:** https://github.com/singh-atul/ecommerce-vanillajs-master

## Practice Ques/ HW

Create a replica of amazon or myntra shopping portal where a user must login to place an order successfully.

Thankyou