

Asian Boba Drink Deliveroo App -> A responsive Full-Stack Asian-Boba-delivering website using Next.js and MongoDB. This project contains a frontend-side, which are a homepage, product pages, cart total & a payment selection page to submit name & address & phone number details & checkout, with data fetched from MongoDB integration, and a backend-side Admin Dashboard as a merchant to receive payments. The user can choose any size, additional toppings and quantities into carts. The JS Redux library helps increase quantity numbers, and the PayPal library helps generate fake transactions. The amount increases immediately as the user makes a purchase. I have learned Next.js, MongoDB integration, server-side rendering, and tremendous knowledge of how to use cookies for authentication from implementing this project.

Functionalities/Demo:

- A nice slider on top to show the list of boba drinks and their features.
- In the cart page, the user can view what products they have chosen.
- There are two checkout options when the user clicks to pay: Cash on delivery or PayPal.
- The customer will be redirected to the order status page to track after submitting payment.
- The user can see all their previous orders and their payment statuses: paid/unpaid.
- As the merchant/Admin, we can always add new items(new boba drinks) to the homepage.

Developing Languages, Tools, and Techniques Needed:

Vscode 1.73.1 https://code.visualstudio.com/updates/v1_73

Next.JS <https://nextjs.org/>

Yarn 1.22.19 <https://classic.yarnpkg.com/lang/en/docs/install/#mac-stable>

npm package <https://www.npmjs.com/>

React JSX JavaScript XML <https://reactjs.org/docs/introducing-jsx.html>

JavaScript www.javascript.com

MongoDB Cloud <https://account.mongodb.com/account/login>

Postman API Platform <https://www.postman.com/downloads/>

Prerequisites & Setups:

In Vscode Console:

```
npx create-next-app
```

Install all Next.js properties using .

Install Yarn globally using sudo npm:

```
sudo npm install -g yarn
```

Check if yarn installed and at latest version:

```
yarn --version
```

Start the localhost server at www.localhost:3000

```
yarn dev
```

Synchronous Developing Notes:

NavBar Design:

Create a basic template n NavBar.jsx.

Create NavBar.module.css:

```
.container{  
  background-color: red;
```

Test if navbar CSS styling works:

red navbar slider test.PNG

In Navbar.jsx:

```
import Image from "next/image";  
...  
<Image src="/img/telephone.png" alt="" width="32" height="32" />
```

Order Now telephone symbol is imported:

order now telephone symbol imported.PNG

Import and add more CSS stylings designs to the list items and cart:

```
.counter{  
  position: absolute;  
  top: -10px;  
  right: -10px;  
  width: 20px;  
  height: 20px;  
  border-radius: 50%;  
  background-color: white;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  font-weight: bold;  
  color: #ff00aa;
```

navigation bar design is done.PNG

Slider Design:

Import popular featured items with customized copyrighted images and set layout as fill:

```
<div className={styles.wrapper}>  
  <div className={styles.imgContainer}>  
    {images.map((img, i) => (  
      <Image src="/img/featured.png" key={i} alt="" layout="fill" />  
    ))}  
  </div>  
</div>
```

Now imported featured items images displayed and size filled out the background properly:

imported featured item picture layout properly.PNG

Add horizontal slider to allow user to slide to view featured items horizontally:

```
.wrapper{  
  width: 300vw;  
  height: 100%;  
  display: flex;  
}  
.imgContainer{  
  width: 100vw;  
  height: 100%;  
  position: relative;
```

featured items can be slided horizontally.MOV

To enable arrow left and arrow right to be clickable, add transform effect to translateX:

```
<div className={styles.wrapper} style={{transform:
`translateX(${ -100*index}vw)`}}>
```

left right click arrow to switch the featured items.MOV

Listing products:

Import boba cards template image and template details, and add CSS stylings to the details:

```
.title {
  font-size: 18px;
  font-weight: bold;
  color: #d1411e;
}
.price{
  font-size: 18px;
  font-weight: bold;
  color: #666;
}
.desc{
  text-align: center;
  color: #777;
}
```

template products listings displayed.PNG

Design Footer:

Import footer contact infos in Footer.jsx:

```
<div className={styles.item}>
  <div className={styles.card}></div>
  <h2 className={styles.motto}>
    OH YES, WE DID. THE KRYSTAL BOBA,
    WELL BLENDED REFRESHING CUP OF BOBA.
  </h2>
  <div className={styles.card}>
    <h1 className={styles.title}>FIND OUR BOBA
SHOP</h1>
    <p className={styles.text}>
      3638 R. Nob Hill Street #304.
      <br /> San Francisco, 94105
      <br /> (415) 555-7372
    </p>...
```

Add corresponding CSS stylings and footer design is done:

footer design is done.PNG

Product Page Design:

CSS ERROR: Product not showing in localhost:3000/product/213 page. DEBUGGING: In Product.module.css. Leave space for height: calc(100vh - 100px);

Import different sizes, additional ingredients(toppings) for boba drinks as options in html checkbox format and also import button for add to cart in [id].jsx:

```
return (
  <div className={styles.container}>
    <div className={styles.left}>
      <div className={styles.imgContainer}>
        <Image src={boba.img} objectFit="contain"
layout="fill" alt="" />
      </div>
    </div>
    <div className={styles.right}>
      <h1 className = {styles.title}>{boba.name}</h1>
      <span className =
{styles.price}>${boba.price[size]}</span>
      <p className = {styles.desc}>{boba.desc}</p>
      <h3 className = {styles.choose}>Choose the size</h3>
      <div className = {styles.sizes}>
        <div className = {styles.size} onClick = {()=>
setSize(0)}>
          <Image src = "/img/size.png" layout = "fill"
alt = ""/>
          <span className = {styles.number}>Small</span>
        </div>
        <div className = {styles.size} onClick = {()=>
setSize(1)}>
          <Image src = "/img/size.png" layout = "fill"
alt = ""/>
          <span className =
{styles.number}>Medium</span>
        </div>
        <div className = {styles.size} onClick = {()=>
setSize(2)}>
          <Image src = "/img/size.png" layout = "fill"
alt = ""/>
          <span className = {styles.number}>Large</span>
        </div>
      </div>
      <h3 className = {styles.choose}> Choose additional
toppings!</h3>
      <div className = {styles.ingredients}>
        <div className = {styles.option}>
          <input...
```

Modify all icons, buttons and checkboxes stylings then we got:

product page is done designing.PNG

Shopping Cart Design:

WEB SERVER ERROR: Unhandled Runtime Error: Error: Hydration failed because the initial UI does not match what was rendered on the server. See more info here:

<https://nextjs.org/docs/messages/react-hydration-error>.

DEBUGGING: Latest React server not comfortable with using table without having tbody. So replace `<table><tr>...</tr></table>` with

`<table><tbody><tr>...</tr></tbody></table>`

Import shopping cart items in `cart.jsx`:

```
<div className={styles.container}>
  <div className={styles.left}>
    <table className={styles.table}>
      <tbody className={styles.tbody}>
        <tr className={styles.tr}>
          <th>Product</th>
          <th>Name</th>
          <th>Extras</th>
          <th>Price</th>
          <th>Quantity</th>
          <th>Total</th>
        </tr>
```

Adjust shopping cart icon and its button CSS stylings in `cart.module.css`:

```
.wrapper{
  width: 90%;
  max-height: 300px;
  background-color: #333;
  padding: 50px;
  padding-top: 10px;
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  color: white;
}
.totalTextTitle{
  margin-right: 10px;
}
```

Now we have a well-designed shopping cart page at `localhost:3000/cart`:

shopping cart total page for checkout.PNG

NOTE: Use `display: flex` in CSS container to make items align horizontally.

Order Tracking Pages:

Create different statuses in `id.jsx` and import corresponding images:

```
const Order = () => {  
  const status = 0  
  const statusClass = (index) => {  
    if (index - status < 1) return styles.done;  
    if (index - status === 1) return styles.inProgress;  
    if (index - status > 1) return styles.undone;  
  };  
};
```

Add CSS animation stylings to the order status in `Order.module.css`:

```
.row:last-child{  
  width: 80%;  
  display: flex;  
  justify-content: space-between;  
}  
.done{  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
}  
.inProgress{  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  animation: inProgress 1s ease infinite alternate;  
}
```

order status page design is done.PNG

Next.js responsive page design:

Use `@media screen and (max-width: 480px) { ... }` to make responsive effects.

Add multiple boba cards in the homepage.

overview links:

<http://localhost:3000/>

<http://localhost:3000/product/213>

<http://localhost:3000/cart>

<http://localhost:3000/orders/21>