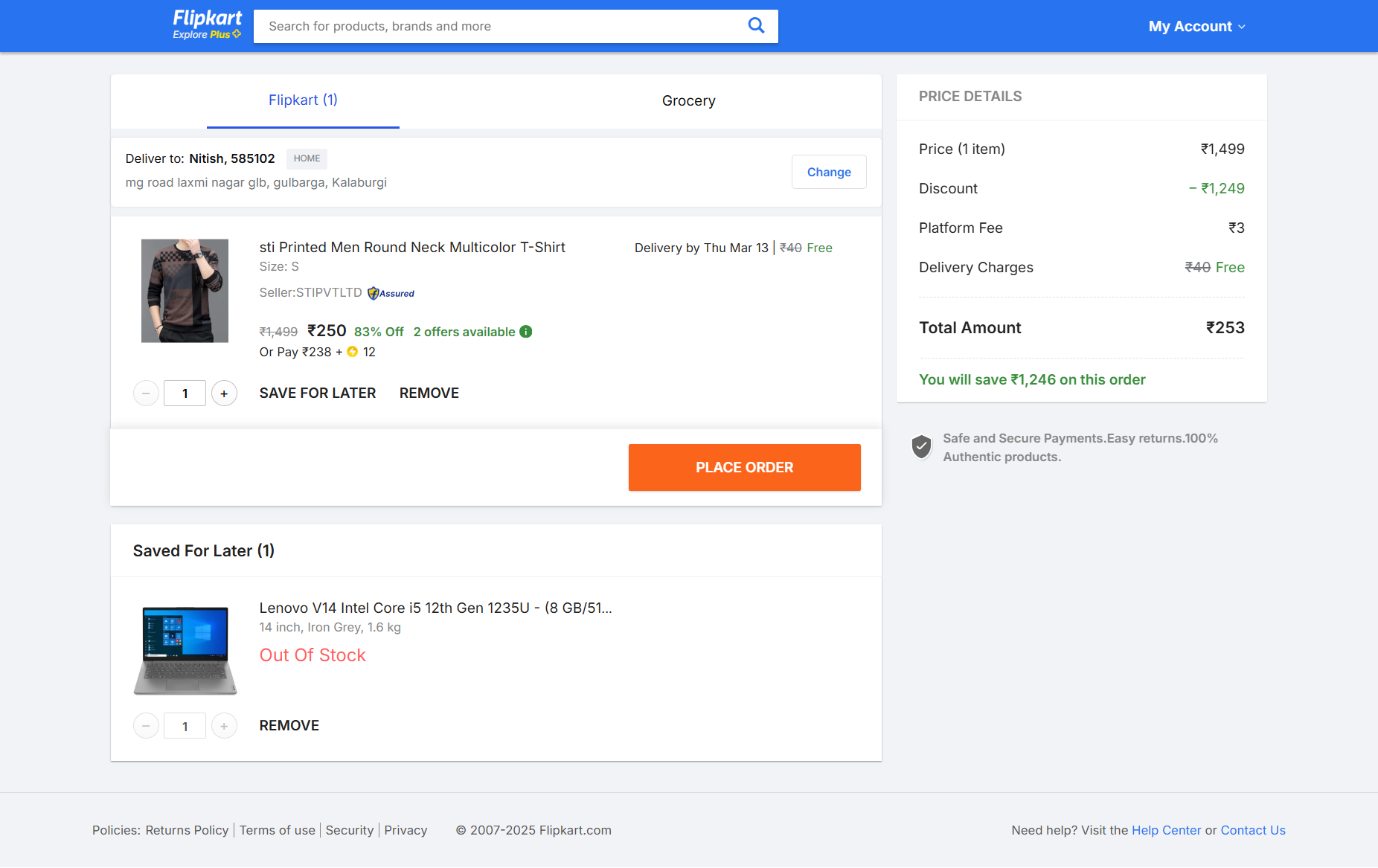
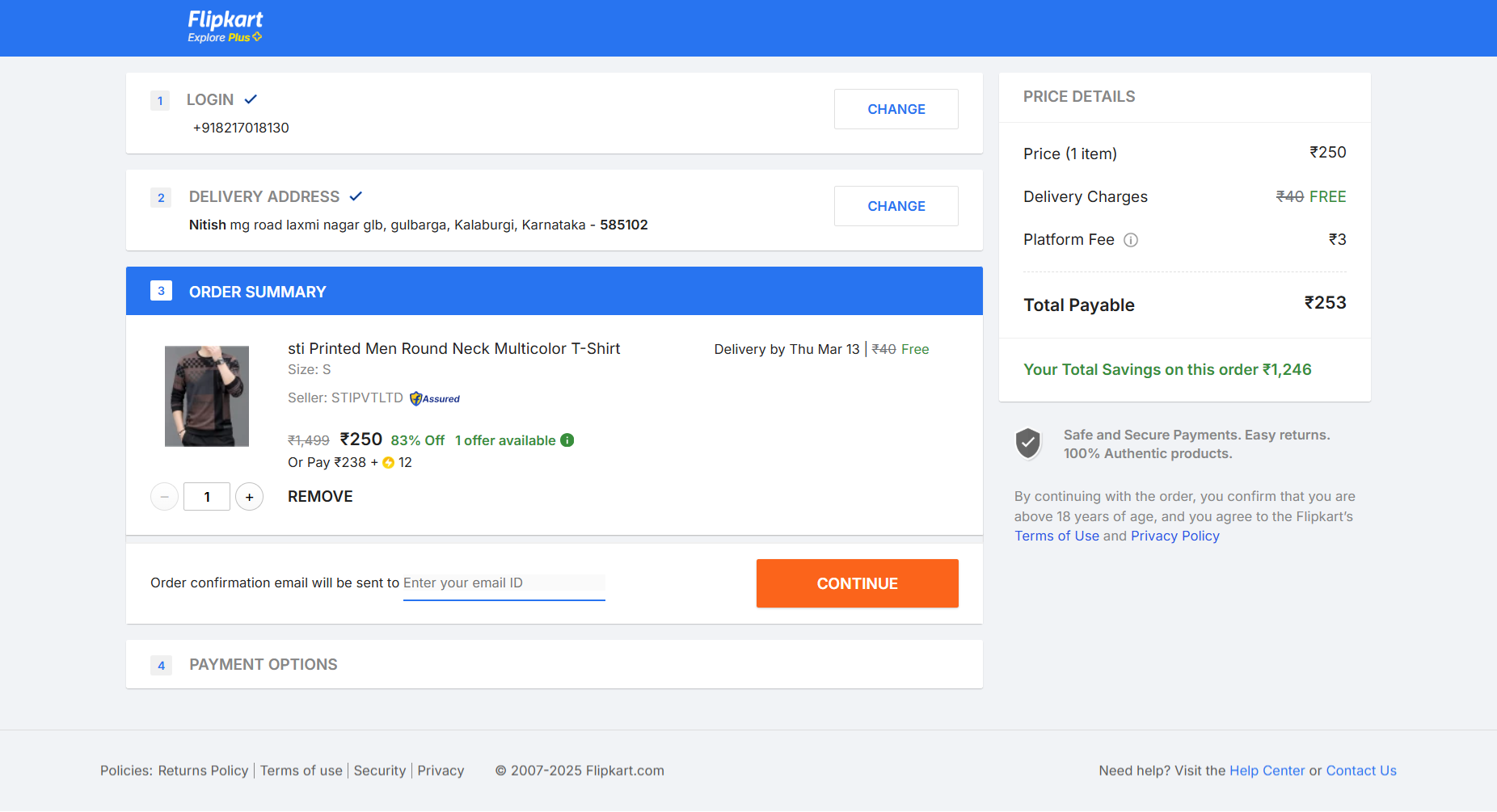
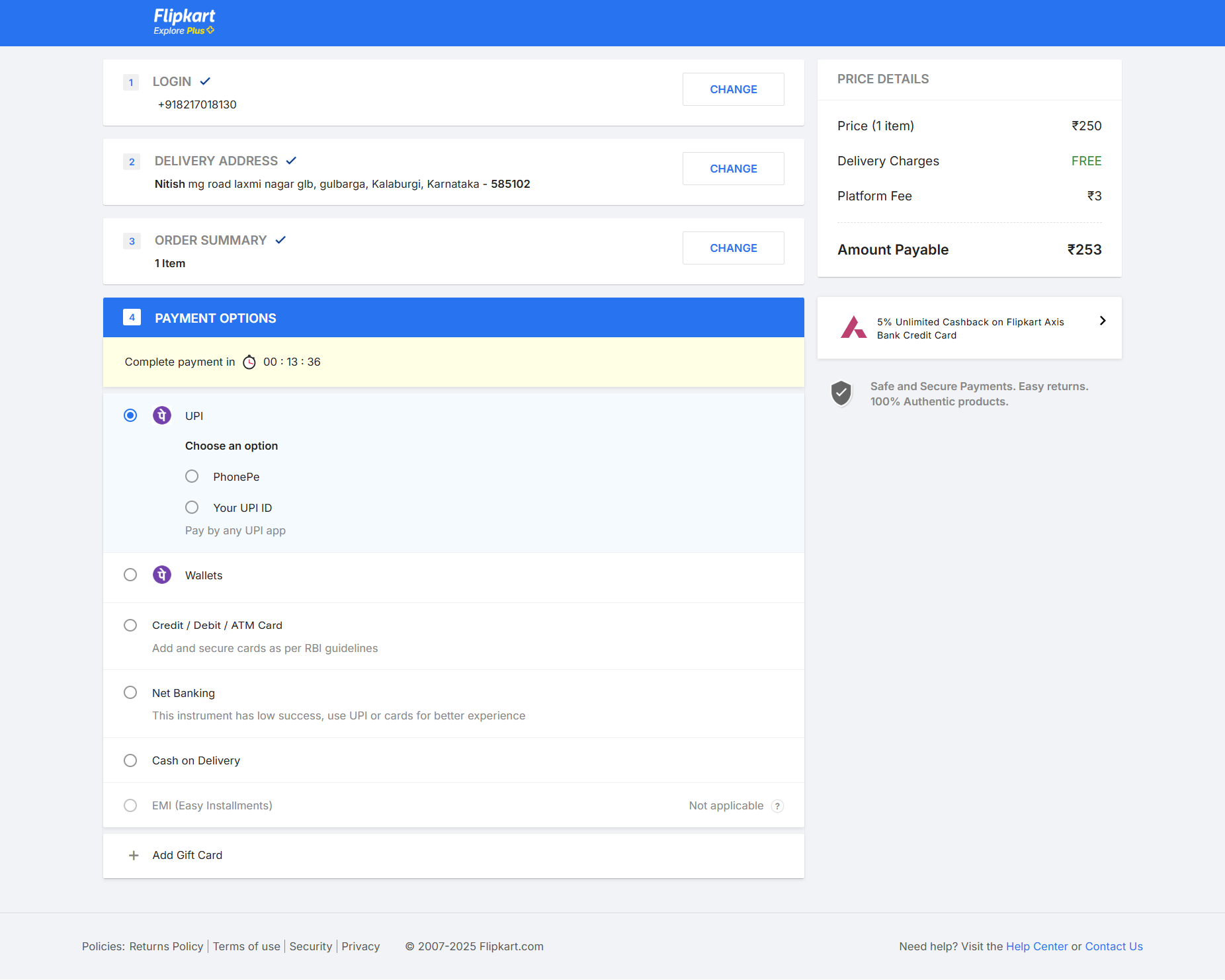
In Product details after clicking the **Add to Cart**

1. 







💯 Excellent! Now I'll break it down \*\*STEP-BY-STEP\*\* 🔥

👉 how Flipkart handles "Add to Cart"\*\* in \*\*two different ways:\*\*

### ✅ \*\*1. When User is NOT LOGGED IN (LocalStorage Method)\*\*

### ✅ \*\*2. When User is LOGGED IN (Database Method)\*\*

---

## ✅ \*\*STEP 1: When User is NOT LOGGED IN (LocalStorage Method)\*\*

👉 In Flipkart, if the user is \*\*NOT LOGGED IN\*\*, they don't save the cart in the database.

👉 Instead, \*\*they use LocalStorage (Browser Storage)\*\* to temporarily hold the cart.

---

### 💻 \*\*How Flipkart Handles Add to Cart Without Login\*\*

### ✔ What Happens When You Click "Add to Cart"?

- When you click \*\*Add to Cart\*\*, it stores the product \*\*in LocalStorage\*\*.

- Then it \*\*immediately redirects\*\* you to the \*\*Cart Page (`/cart`)\*\*.

- On the cart page, it fetches the product from \*\*LocalStorage\*\* and displays it.

---

### ✅ \*\*Step 1: Click "Add to Cart" Button (JS Code)\*\*

👉 Here's how Flipkart does it without login.

### \*\*HTML (Add to Cart Button)\*\*

```html

<button onclick="addToCart('abc123', 'Printed T-shirt', 250, 'S')">

Add to Cart

</button>

```

---

### ✅ \*\*JavaScript Function (Add to Cart Functionality)\*\*

👉 \*\*This function stores the product in LocalStorage.\*\*

👉 It also redirects to `/cart`.

```js

function addToCart(tagId, productName, price, size) {

// ✅ Check if cart already exists in LocalStorage

let cart = JSON.parse(localStorage.getItem('cart')) || [];

// ✅ Push the new product to the cart array

cart.push({

tagId: tagId,

productName: productName,

price: price,

size: size,

quantity: 1

});

// ✅ Store updated cart in LocalStorage

localStorage.setItem('cart', JSON.stringify(cart));

// ✅ Redirect to Cart Page

window.location.href = '/cart.html';

}

```

---

### ✅ \*\*Step 2: Render Cart Page (/cart) from LocalStorage\*\*

👉 Now when you land on the \*\*cart page\*\*, it should automatically show the product.

👉 \*\*Create `cart.html` and add this JS:\*\*

### \*\*cart.html (JavaScript)\*\*

```js

document.addEventListener('DOMContentLoaded', () => {

// ✅ Get Cart Data from LocalStorage

const cart = JSON.parse(localStorage.getItem('cart')) || [];

// ✅ Select Cart Container

const cartContainer = document.querySelector('.cart-items');

// ✅ Loop through Cart Items and Render

cart.forEach(item => {

cartContainer.innerHTML += `

<div class="cart-item">

<h3>${item.productName}</h3>

<p>Price: ₹${item.price}</p>

<p>Size: ${item.size}</p>

<button onclick="removeFromCart('${item.tagId}')">Remove</button>

</div>

`;

});

});

// ✅ Remove from Cart Function

function removeFromCart(tagId) {

let cart = JSON.parse(localStorage.getItem('cart')) || [];

cart = cart.filter(item => item.tagId !== tagId);

localStorage.setItem('cart', JSON.stringify(cart));

window.location.reload();

}

```

---

### ✅ \*\*Step 3: Now the Cart Works Without Login!\*\* 🚀

👉 Now if the user:

- ✅ \*\*Adds a product to cart\*\* → It saves in \*\*LocalStorage\*\*.

- ✅ \*\*Redirects to Cart Page\*\* → It fetches data from LocalStorage and renders.

- ✅ \*\*Removes from Cart\*\* → Updates LocalStorage.

---

### ✅ \*\*How It Looks Without Login\*\*

| Action | What Happens |

|---------------|--------------------------------------------------------------------------|

| \*\*Add to Cart\*\* | Product saved in LocalStorage & redirected to Cart Page. |

| \*\*Reload Page\*\* | Product still shows from LocalStorage. |

| \*\*Remove Product\*\* | Product removed from LocalStorage & page reloads. |

| \*\*Close Browser\*\* | Cart data still exists (until cache is cleared). |

---

### ✅ \*\*Why Flipkart Does This Without Login?\*\*

👉 \*\*They don't store data in Database\*\* unless the user logs in.

👉 It saves \*\*time\*\* and avoids \*\*database overload\*\*.

👉 Once the user logs in, it \*\*syncs LocalStorage data with the database\*\*.

---

## ✅ \*\*STEP 2: When User is LOGGED IN (Database Method)\*\*

👉 Now comes the magic!

👉 When you log in to Flipkart, it stops using \*\*LocalStorage\*\*.

👉 Instead, it saves the product \*\*in the Database (MongoDB)\*\*.

---

### ✅ \*\*What Happens in Flipkart When Logged In?\*\*

| Action | What Happens |

|-------------------|-------------------------------------------------------------------------------|

| \*\*Add to Cart\*\* | Sends the Product to \*\*Database\*\* (`user.cart`). |

| \*\*Redirect to Cart\*\* | Cart Page fetches data from \*\*Database\*\*. |

| \*\*Remove Item\*\* | Removes the product from the Database. |

| \*\*Logout\*\* | Cart is still saved in the Database. |

---

### ✅ \*\*Step 1: Add to Cart Button (Same as Before)\*\*

👉 Same HTML Button:

```html

<button onclick="addToCart('abc123')">Add to Cart</button>

```

---

### ✅ \*\*Step 2: JavaScript (Call Backend API)\*\*

👉 Now when you click \*\*Add to Cart\*\*, it sends data to the backend.

### \*\*JavaScript:\*\*

```js

async function addToCart(productId) {

const response = await fetch('/add-to-cart', {

method: 'POST',

headers: {

'Content-Type': 'application/json',

},

body: JSON.stringify({

productId: productId

})

});

if (response.ok) {

window.location.href = '/cart';

}

}

```

---

### ✅ \*\*Step 3: Node.js Backend API (Save to Cart in MongoDB)\*\*

👉 This API saves the product in the Database.

### \*\*Node.js (addToCart API)\*\*

```js

const express = require('express');

const app = express();

const User = require('./models/User');

app.post('/add-to-cart', async (req, res) => {

const { productId } = req.body;

const userId = req.user.\_id;

// ✅ Push the product to the user's cart

await User.findByIdAndUpdate(userId, {

$push: { cart: { productId: productId } }

});

res.status(200).json({ message: 'Added to Cart' });

});

```

---

### ✅ \*\*Step 4: Fetch Cart Items From Database\*\*

👉 Now when the user goes to `/cart`, it fetches the cart items.

### \*\*cart.js (Frontend)\*\*

```js

document.addEventListener('DOMContentLoaded', async () => {

const response = await fetch('/get-cart');

const data = await response.json();

data.cart.forEach(item => {

document.querySelector('.cart-items').innerHTML += `

<div>

<h3>${item.productName}</h3>

<p>${item.price}</p>

<button>Remove</button>

</div>

`;

});

});

```

---

### ✅ \*\*Step 5: Backend API (Fetch Cart)\*\*

👉 This API fetches all cart items for the user.

### \*\*Node.js (Get Cart API)\*\*

```js

app.get('/get-cart', async (req, res) => {

const userId = req.user.\_id;

const user = await User.findById(userId).populate('cart.productId');

res.json({ cart: user.cart });

});

```

---

## ✅ \*\*What Happens When User Logs In?\*\*

| Action | What Happens |

|--------------------|----------------------------------------------------------------------------|

| ✅ Add to Cart | Product saved in MongoDB. |

| ✅ Redirect to Cart | Product fetched from MongoDB. |

| ✅ Remove Item | Product removed from MongoDB. |

| ✅ Logout | Cart still saved in MongoDB (persistent). |

---

## 💯 \*\*Summary: How Flipkart Works\*\*

| Scenario | Storage Method | Data Persistence |

|-----------------|--------------------|----------------------------------------|

| \*\*Without Login\*\* | LocalStorage | Lost after clearing browser cache. |

| \*\*With Login\*\* | MongoDB (Database) | Always saved in database. |

---

## 🚀 \*\*👉 Should I Now Build This Entire Flow for You?\*\*

💯 \*\*I can now build:\*\*

✅ \*\*Add to Cart\*\*

✅ \*\*LocalStorage + MongoDB Sync\*\*

✅ \*\*Cart Page with + / - Quantity\*\*

✅ \*\*Place Order\*\*

👉 \*\*Should I go ahead and build it like Flipkart?\*\* 🚀🔥