

```

File: cart.html
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Cart Page</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <nav id="navbar">
    <a href="home.html">
      
    </a>

    <div class="cart-button-container">
      <button id="view-cart">View Cart</button>
      <a href="orders.html" class="orders-link">View Orders</a>
    </div>
  </nav>

  <div id="cart-items-container"></div>
  <div id="order-total"></div>
  <button id="checkout">Check Out</button>
  <button id="continue-shopping">Continue Shopping</button>

  <script src="cart.js"></script>
</body>
</html>

```

```

File: home.html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Product Page</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <nav id="navbar">
    <a href="home.html">
      
    </a>
    <div class="cart-button-container">
      <button id="view-cart">View Cart</button>
      <a href="orders.html" class="orders-link">View
Orders</a>

```

```
        </div>
    </nav>

    <div id="product-container"></div>

    <script src="script.js"></script>
</body>
</html>
File: orders.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Order History</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <nav id="navbar">
        <a href="home.html">
            
        </a>
        <div class="cart-button-container">
            <button id="view-cart">View Cart</button>
            <a href="orders.html" class="orders-link">View Orders</a>
        </div>
    </nav>

    <h1>Order History</h1>
    <div id="orders-container"></div>
    <script src="orders.js"></script>
</body>
</html>
File: thank-you.html
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Thank You</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <nav id="navbar">
        <a href="home.html">
            
        </a>
```

```

        <div class="cart-button-container">
            <button id="view-cart">View Cart</button>
            <a href="orders.html" class="orders-link">View Orders</a>
        </div>
    </nav>

    <h1>Thank you for your order!</h1>
    <p>Your order has been placed successfully.</p>
    <p>Order Total: $<span id="order-total"></span></p>
    <p>Expected Ship Date: <span id="expected-ship-date"></span></p>
    <script src="thank-you.js"></script>
</body>
</html>
File: cart.js
let cart = [];

document.addEventListener('DOMContentLoaded', () => {
    cart = JSON.parse(localStorage.getItem('cart')) || [];
    document.getElementById('checkout').addEventListener('click',
checkout);
    document.getElementById('continue-
shopping').addEventListener('click', continueShopping);
    renderCartItems();
    fetchOrders();
});

function renderCartItems() {
    const cartItemsContainer = document.getElementById('cart-items-
container');
    cartItemsContainer.innerHTML = '';
    let total = 0;

    if (cart.length === 0) {
        cartItemsContainer.innerHTML = '<p>Your cart is empty.</p>';
        return;
    }

    cart.forEach((item, index) => {
        const itemCost = item.price * item.quantity;
        total += itemCost;

        const itemDiv = document.createElement('div');
        itemDiv.innerHTML = `
            <p>${item.name} - Quantity: ${item.quantity} - Price: $$
${item.price.toFixed(2)} - Total: $$${itemCost.toFixed(2)}</p>
            <button onclick="removeFromCart(${index})">Remove from
cart</button>
        `;
        cartItemsContainer.appendChild(itemDiv);
    });
}

```

```
});

const orderTotalDiv = document.getElementById('order-total');
orderTotalDiv.textContent = `Order Total: ${total.toFixed(2)}`;
}

function removeFromCart(index) {
  if (cart[index].quantity > 1) {
    cart[index].quantity -= 1;
  } else {
    cart.splice(index, 1);
  }
  localStorage.setItem('cart', JSON.stringify(cart));
  renderCartItems();
}

function checkout() {
  submitOrderToServer(cart).then(response => {
    const orderTotal = response.orderTotal;
    const expectedShipDate = new Date();
    expectedShipDate.setDate(expectedShipDate.getDate() + 2);

    localStorage.setItem('orderTotal',
JSON.stringify(orderTotal));
    localStorage.setItem('expectedShipDate',
expectedShipDate.toISOString());

    cart = [];
    localStorage.removeItem('cart');

    window.location.href = 'thank-you.html';
  }).catch(error => {
    console.error('Checkout error:', error);
  });
}

function submitOrderToServer(cart) {
  return fetch('http://localhost:3001/api/orders', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ items: cart.map(item => ({ itemId:
item._id, quantity: item.quantity })) })
  }).then(response => {
    if (!response.ok) {
      throw new Error('Network response was not ok');
    }
    return response.json();
  });
}
```

```
function continueShopping() {
    window.location.href = 'home.html';
}
File: orders.js
document.addEventListener('DOMContentLoaded', () => {
    const viewCartButton = document.getElementById('view-cart');
    if (viewCartButton) {
        viewCartButton.addEventListener('click', viewCart);
    }
    fetchOrders();
});

function fetchOrders() {
    fetch('http://localhost:3001/api/orders')
        .then(response => {
            console.log(response);
            return response.json();
        })
        .then(orders => displayOrders(orders))
        .catch(error => console.error('Error fetching orders:',
error));
}

function viewCart() {
    window.location.href = 'cart.html';
}

function displayOrders(orders) {
    const ordersContainer = document.getElementById('orders-
container');
    ordersContainer.innerHTML = '';

    orders.forEach(order => {
        const orderDiv = document.createElement('div');
        orderDiv.className = 'order';
        let orderContent = `<h3>Order ID: ${order._id}</h3>`;
        orderContent += `<p>Date: ${new
Date(order.dateOrdered).toLocaleDateString()}</p>`;
        orderContent += `<p>Order Total: ${order.items.reduce((total,
item) => total + item.itemId.price * item.quantity, 0).toFixed(2)}</
p>`;
        orderContent += '<ul>';

        order.items.forEach(item => {
            orderContent += `<li>${item.itemId.name} - Quantity: $
${item.quantity} - Cost: ${((item.itemId.price *
item.quantity).toFixed(2))}</li>`;
        });
    });
}
```

```
        orderContent += '</ul>';
        orderDiv.innerHTML = orderContent;
        ordersContainer.appendChild(orderDiv);
    });
}
File: routes.js
const express = require('express');
const router = express.Router();
const Product = require('./models/Product');
const Order = require('./models/Order');

// Get all products
router.get('/api/products', async (req, res) => {
    try {
        const products = await Product.find();
        res.json(products);
    } catch (error) {
        res.status(500).json({ message: error.message });
    }
});

// Create a new order
router.post('/api/orders', async (req, res) => {
    try {
        let orderTotal = 0;
        for (const item of req.body.items) {
            const product = await Product.findById(item.itemId);
            if (!product) {
                throw new Error(`Product with ID ${item.itemId} not
found`);
            }
            orderTotal += product.price * item.quantity;
        }

        const newOrder = new Order({
            items: req.body.items,
            dateOrdered: new Date(),
        });

        await newOrder.save();

        res.status(201).json({ orderTotal: orderTotal, newOrder:
newOrder });
    } catch (error) {
        console.error("Error saving order:", error);
        res.status(500).json({ error: "Error saving order" });
    }
});

router.get('/api/orders', async (req, res) => {
```

```
    try {
      const orders = await Order.find().sort({ dateOrdered:
-1 }).populate('items.itemId');
      res.json(orders);
    } catch (error) {
      res.status(500).json({ message: error.message });
    }
  });

module.exports = router;
File: script.js
let cart = [];
let products = [];

document.addEventListener('DOMContentLoaded', () => {
  fetch('http://localhost:3001/api/products')
    .then(response => response.json())
    .then(data => {
      products = data;
      displayProducts(products);
    })
    .catch(error => console.error('Fetching error:', error));

  const viewCartButton = document.getElementById('view-cart');
  if (viewCartButton) {
    viewCartButton.addEventListener('click', () => {
      window.location.href = 'cart.html';
    });
  }
});

function toggleDescription(productId) {
  const description = document.getElementById(`description-${productId}`);
  if (description) {
    description.style.display = description.style.display ===
'block' ? 'none' : 'block';
  }
}

function addToCart(productId) {
  const product = products.find(p => p._id === productId);
  const quantitySelect = document.getElementById(`quantity-${productId}`);
  const quantity = quantitySelect ? parseInt(quantitySelect.value) :
1;

  if (product) {
    const cartItem = cart.find(item => item._id === productId);
```

```
        if (cartItem) {
            cartItem.quantity += quantity;
        } else {
            cart.push({ ...product, quantity });
        }

        console.log(cart);
        alert('Product added to cart successfully!');
    }
    localStorage.setItem('cart', JSON.stringify(cart));
}

function viewCart() {
    const cartContainer = document.getElementById('cart-container');
    const cartItemsDiv = document.getElementById('cart-items');

    cartItemsDiv.innerHTML = '';

    cart.forEach(product => {
        const itemDiv = document.createElement('div');
        itemDiv.className = 'cart-item';
        itemDiv.innerHTML = `
            <p>${product.name} - Quantity: ${product.quantity} -
Total: $$${(product.price * product.quantity).toFixed(2)}</p>
            `;
        itemDiv.innerHTML += `<button onclick="removeFromCart('${product._id}')">Remove from cart</button>`;
        cartItemsDiv.appendChild(itemDiv);
    });

    cartContainer.style.display = cartContainer.style.display ===
'block' ? 'none' : 'block';
    document.getElementById('checkout').addEventListener('click',
checkout);
}

function removeFromCart(productId) {
    cart = cart.filter(item => item._id !== productId);
    viewCart();
}

function displayProducts(products) {
    const productContainer = document.getElementById('product-
container');
```



```

    console.log(products);
    products.forEach(product => {
      const productDiv = document.createElement('div');
      productDiv.className = 'product-item';
      const imageUrl = `http://localhost:3001${product.imageUrl}`;
      productDiv.innerHTML = `
        <img src='${imageUrl}' alt='${product.name}'
class="product-image">
        <div class="product-name">${product.name}</div>
        <div class="product-price">${product.price.toFixed(2)}</
div>
        <p class="product-description" id="description-${
product._id}" style="display: none;">${product.description}</p>
        <select class="quantity-select" id="quantity-${
product._id}">
          <option value="1">1</option>
          <option value="2">2</option>
          <option value="3">3</option>
          <option value="4">4</option>
          <option value="5">5</option>
        </select>
        <button onclick="addToCart('${product._id}')">Add to
Cart</button>
        <button onclick="toggleDescription('${
product._id}')">More</button>
      `;
      productContainer.appendChild(productDiv);
    });

    const viewCartButton = document.getElementById('view-cart');
    if (viewCartButton) {
      viewCartButton.addEventListener('click', viewCart);
    }
  }
}

```

```

File: server.js
require('dotenv').config();
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');
const app = express();

app.use(cors());
app.use(express.json());
app.use('/images', express.static('images'));

// Connect to MongoDB
mongoose.connect(process.env.MONGODB_URI, {

```

```
    useUrlParser: true,
    useUnifiedTopology: true
  }).then(() => {
    console.log('Connected to the MongoDB database.');
```

```
  }).catch((error) => {
    console.error('MongoDB connection error:', error);
  });

const productRoutes = require('./routes');
app.use(productRoutes);

// Start the server
const PORT = process.env.PORT || 3001;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

```
File: thank-you.js
document.addEventListener('DOMContentLoaded', () => {
  const orderTotal = localStorage.getItem('orderTotal');
  const expectedShipDate = new
Date(localStorage.getItem('expectedShipDate'));

  document.getElementById('order-total').textContent = orderTotal;
  document.getElementById('expected-ship-date').textContent =
expectedShipDate.toLocaleDateString();

  localStorage.removeItem('orderTotal');
  localStorage.removeItem('expectedShipDate');
});
File: styles.css
#navbar {
  background-color: #2c3e50;
  color: white;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 10px 20px;
}

@media screen and (max-width: 600px) {
  #navbar {
    flex-direction: column;
    align-items: flex-start;
  }
}

#navbar #logo {
```

```
    max-height: 50px;
    width: auto;
}

#product-container {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    justify-content: center;
}

.product-item {
    border: 1px solid #ddd;
    padding: 10px;
    text-align: center;
    flex: 1 1 200px;
    max-width: calc(33.333% - 20px);
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
    transition: transform 0.3s;
}

.product-item:hover {
    transform: scale(1.05);
}

@media screen and (max-width: 600px) {
    .product-item {
        max-width: calc(50% - 20px);
    }
}

.product-image {
    max-width: 80%;
    height: auto;
    background-color: #f3f3f3;
    margin-bottom: 10px;
}

#cart-container {
    position: fixed;
    right: 0;
    top: 0;
    width: 300px;
    max-width: 70%;
    height: 100%;
    background-color: white;
    box-shadow: -2px 0px 5px rgba(0, 0, 0, 0.5);
    padding: 20px;
    display: none;
    overflow-y: auto;
}
```

```
}

body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 0;
}

header {
  background-color: #333;
  color: white;
  padding: 10px 0;
  text-align: center;
}

#cart-items-container {
  width: 80%;
  margin: 20px auto;
  background: white;
  padding: 20px;
  box-shadow: 0 2px 5px rgba(0,0,0,0.2);
}

.cart-item {
  border-bottom: 1px solid #ddd;
  padding: 10px 0;
}

.cart-item p {
  margin: 0;
  padding: 5px 0;
}

button {
  background-color: #5cb85c;
  color: white;
  border: none;
  padding: 10px 20px;
  margin: 10px 0;
  cursor: pointer;
  border-radius: 5px;
}

button:hover {
  background-color: #4cae4c;
}

#continue-shopping {
```

```
        background-color: #0275d8;
    }

    #continue-shopping:hover {
        background-color: #025aa5;
    }

    #order-total {
        font-size: 1.2em;
        padding: 10px 0;
    }

    .order {
        border: 1px solid #ccc;
        margin-bottom: 10px;
        padding: 10px;
    }

    .order ul {
        list-style-type: none;
        padding: 0;
    }

    .orders-link {
        display: inline-block;
        margin-left: 10px;
        padding: 10px 15px;
        background-color: #4CAF50;
        color: white;
        text-decoration: none;
        border-radius: 5px;
        font-size: 1em;
    }

    .orders-link:hover {
        background-color: #45a049;
    }

    @media screen and (max-width: 600px) {
        #cart-items-container {
            width: 95%;
        }

        button {
            width: 100%;
        }
    }

    @media screen and (max-width: 600px) {
```

```
#cart-container {  
  width: 100%;  
}  
}
```