

## 1. Frontend (HTML/CSS/JavaScript)

We'll create a simple store with product listings and a shopping cart.

### HTML

html

Copy code

```
<!DOCTYPE html><html lang="en"><head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Basic E-Commerce Store</title>
  <link rel="stylesheet" href="styles.css"></head><body>
  <header>
    <h1>My E-Commerce Store</h1>
    <div id="cart">
      <span>Cart: <span id="cart-count">0</span>
items</span>
      <button onclick="checkout()">Checkout</button>
    </div>
  </header>

  <main>
    <div class="product-list">
      <!-- Products will be loaded here -->
    </div>
  </main>

  <script src="scripts.js"></script></body></html>
```

## CSS (styles.css)

```
css
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
}
header {
  background-color: #333;
  color: white;
  padding: 15px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}
.product-list {
  display: flex;
  justify-content: space-around;
  margin-top: 20px;
}
.product {
  border: 1px solid #ddd;
  padding: 20px;
  text-align: center;
}
button {
  background-color: #28a745;
  color: white;
  padding: 10px;
  border: none;
  cursor: pointer;
}
button:hover {
  background-color: #218838;
}
```

## JavaScript (scripts.js)

javascript

Copy code

```
let cart = [];  
const products = [  
  { id: 1, name: 'Product 1', price: 100 },  
  { id: 2, name: 'Product 2', price: 150 },  
  { id: 3, name: 'Product 3', price: 200 },  
];  
document.addEventListener('DOMContentLoaded', () => {  
  loadProducts();  
});  
function loadProducts() {  
  const productList = document.querySelector('.product-list');  
  products.forEach(product => {  
    const productDiv = document.createElement('div');  
    productDiv.classList.add('product');  
    productDiv.innerHTML = `  
      <h3>${product.name}</h3>  
      <p>Price: $$${product.price}</p>  
      <button onclick="addToCart(${product.id})">Add to  
Cart</button>  
    `;  
    productList.appendChild(productDiv);  
  });  
}  
function addToCart(productId) {  
  const product = products.find(p => p.id === productId);  
  cart.push(product);  
  document.getElementById('cart-count').innerText =  
cart.length;  
}  
function checkout() {  
  alert(`You have ${cart.length} items in your cart`);  
  // Add further checkout logic (like sending data to the  
backend)  
}
```

---

## 2. Backend Setup

You can choose either Django or Express.js for the backend. Both approaches involve setting up a simple API to handle product management and orders.

### Django Backend

#### 1.Install Django

```
pip install django
```

#### 2.Set Up Project and App:

```
django-admin startproject ecommercecd ecommerce  
django-admin startapp store
```

#### 3.Models for Products and Orders (store/models.py):

```
from django.db import models  
class Product(models.Model):  
    name = models.CharField(max_length=255)  
    price = models.DecimalField(max_digits=10,  
decimal_places=2)  
    description = models.TextField(blank=True, null=True)  
class Order(models.Model):  
    product = models.ForeignKey(Product,  
on_delete=models.CASCADE)  
    quantity = models.PositiveIntegerField(default=1)  
    created_at = models.DateTimeField(auto_now_add=True)
```

#### 4.Views to Handle Product and Cart Logic (store/views.py):

```

from django.http import JsonResponse
from .models import Product, Order
def product_list(request):
    products = Product.objects.all()
    product_data = [{"id": p.id, "name": p.name, "price":
str(p.price)} for p in products]
    return JsonResponse(product_data, safe=False)
def place_order(request):
    if request.method == "POST":
        # Process the order here (handle product IDs and quantities)
        # For simplicity, return a success response
        return JsonResponse({"status": "Order placed"})

```

## 5. URL

```

from django.contrib import admin
from django.urls import path
from store import views

```

```

urlpatterns = [
    path('admin/', admin.site.urls),
    path('products/', views.product_list),
    path('checkout/', views.place_order),
]

```

## 6. Migrate and Run:

```

python manage.py migrate
python manage.py runserver

```

---

## Express.js Backend

### 1. Install Dependencies:

```

npm init -y
npm install express mongoose body-parser

```

## 2.Set Up Express (server.js):

```
const express = require('express');const mongoose =  
require('mongoose');const bodyParser = require('body-parser');  
const app = express();  
app.use(bodyParser.json());
```

```
mongoose.connect('mongodb://localhost:27017/ecommerce',  
{ useNewUrlParser: true });  
const Product = mongoose.model('Product', new  
mongoose.Schema({  
  name: String,  
  price: Number  
}));
```

```
app.get('/products', async (req, res) => {  
  const products = await Product.find();  
  res.json(products);  
});
```

```
app.post('/checkout', (req, res) => {  
  const order = req.body;  
  // Save order logic  
  res.json({ status: 'Order placed' });  
});
```

```
app.listen(3000, () => {  
  console.log('Server is running on port 3000');  
});
```

## 3.Database Models (Product):

```
const mongoose = require('mongoose');  
const ProductSchema = new mongoose.Schema({  
  name: String,  
  price: Number  
});  
module.exports = mongoose.model('Product', ProductSchema);
```

Run the Server:

```
node server.js
```

---

### 3. Integration

In the frontend's `scripts.js`, update the product and checkout functionality to interact with the backend using `fetch` API.

For example:

javascript

Copy code

```
function loadProducts() {
  fetch('/products')
    .then(response => response.json())
    .then(products => {
      const productList = document.querySelector('.product-
list');
      products.forEach(product => {
        const productDiv = document.createElement('div');
        productDiv.classList.add('product');
        productDiv.innerHTML = `
          <h3>${product.name}</h3>
          <p>Price: $${product.price}</p>
          <button onclick="addToCart(${product.id})">Add
to Cart</button>
        `;
        productList.appendChild(productDiv);
      });
    });
}

function checkout() {
  fetch('/checkout', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ cart })
  })
}
```

```
}).then(response => response.json())  
  .then(data => {  
    alert(data.status);  
  });  
}
```