**Project Structure**

The project is structured into two main parts: backend and frontend.

lua

Copy code

ecommerce-store/

|-- backend/

| |-- models/

| | |-- Product.js

| |-- routes/

| | |-- products.js

| |-- app.js

| |-- package.json

| |-- package-lock.json

|-- frontend/

| |-- css/

| | |-- checkout.css

| | |-- styles.css

| |-- js/

| | |-- checkout.js

| | |-- script.js

| |-- checkout.html

| |-- index.html

**Backend**

1. **Product.js** (Located in backend/models/)

javascript

Copy code

const mongoose = require('mongoose');

// Define the schema for a Product

const ProductSchema = new mongoose.Schema({

name: String,

description: String,

price: Number,

image: String,

});

// Export the Product model based on the schema

module.exports = mongoose.model('Product', ProductSchema);

* **const mongoose = require('mongoose');**: Imports Mongoose, a MongoDB object modeling tool.
* **const ProductSchema = new mongoose.Schema({...});**: Defines the structure of a Product document in MongoDB.
* **module.exports = mongoose.model('Product', ProductSchema);**: Exports a Mongoose model for the Product schema.

1. **products.js** (Located in backend/routes/)

javascript

Copy code

const express = require('express');

const router = express.Router();

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

// Get all products

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

try {

const products = await Product.find();

res.json(products);

} catch (err) {

res.json({ message: err });

}

});

// Add a new product

router.post('/', async (req, res) => {

const product = new Product({

name: req.body.name,

description: req.body.description,

price: req.body.price,

image: req.body.image,

});

try {

const savedProduct = await product.save();

res.json(savedProduct);

} catch (err) {

res.json({ message: err });

}

});

module.exports = router;

* **const express = require('express');**: Imports Express, a web application framework.
* **const router = express.Router();**: Creates a router object.
* **const Product = require('../models/Product');**: Imports the Product model.
* **router.get('/', async (req, res) => {...});**: Defines a route to get all products.
* **router.post('/', async (req, res) => {...});**: Defines a route to add a new product.
* **module.exports = router;**: Exports the router to be used in the main app.

1. **app.js** (Located in backend/)

javascript

Copy code

const express = require('express');

const mongoose = require('mongoose');

const bodyParser = require('body-parser');

const cors = require('cors');

const app = express();

// Middleware

app.use(bodyParser.json());

app.use(cors());

// MongoDB connection

mongoose.connect('mongodb://localhost:27017/ecommerce', {

useNewUrlParser: true,

useUnifiedTopology: true,

});

const db = mongoose.connection;

db.once('open', () => {

console.log('Connected to MongoDB');

});

// Routes

const productRoutes = require('./routes/products');

app.use('/api/products', productRoutes);

const PORT = process.env.PORT || 5000;

app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

* **const express = require('express');**: Imports Express.
* **const mongoose = require('mongoose');**: Imports Mongoose.
* **const bodyParser = require('body-parser');**: Imports body-parser to parse incoming request bodies.
* **const cors = require('cors');**: Imports cors to allow cross-origin requests.
* **const app = express();**: Creates an Express application.
* **app.use(bodyParser.json());**: Configures the app to use body-parser.
* **app.use(cors());**: Configures the app to use CORS.
* **mongoose.connect('mongodb://localhost:27017/ecommerce', {...});**: Connects to a MongoDB database.
* **db.once('open', () => {...});**: Event listener for when the database connection is open.
* **const productRoutes = require('./routes/products');**: Imports product routes.
* **app.use('/api/products', productRoutes);**: Sets up the product routes.
* **const PORT = process.env.PORT || 5000;**: Defines the port the server will run on.
* **app.listen(PORT, () => {...});**: Starts the server.

**Frontend**

1. **checkout.css** (Located in frontend/css/)

css

Copy code

body {

font-family: Arial, sans-serif;

background-color: #f9f9f9;

margin: 0;

padding: 0;

}

header {

background-color: #00796b;

color: white;

padding: 1em;

text-align: center;

}

main {

display: flex;

justify-content: space-between;

padding: 2em;

}

#checkout-cart {

flex: 3;

margin-right: 2em;

}

#checkout-cart div {

border: 1px solid #ccc;

background-color: #fff;

padding: 1em;

margin-bottom: 1em;

}

#summary {

flex: 1;

border: 1px solid #ccc;

background-color: #fff;

padding: 1em;

position: sticky;

top: 0;

}

#summary h2 {

margin-top: 0;

}

#summary p {

font-size: 1.2em;

}

* **body {...}**: Styles for the body of the document.
* **header {...}**: Styles for the header section.
* **main {...}**: Styles for the main content area.
* **#checkout-cart {...}**: Styles for the checkout cart section.
* **#checkout-cart div {...}**: Styles for each item in the checkout cart.
* **#summary {...}**: Styles for the order summary section.

1. **styles.css** (Located in frontend/css/)

css

Copy code

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: aqua;

}

main {

padding: 2rem;

width: 50%;

margin: 50px auto;

text-align: center;

}

table {

margin: 50px auto;

}

table, th, td {

border-collapse: collapse;

border: 1px solid black;

}

* **body {...}**: Styles for the body of the document.
* **main {...}**: Styles for the main content area.
* **table {...}**: Styles for the table and its elements.

1. **checkout.js** (Located in frontend/js/)

javascript

Copy code

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

const checkoutCartContainer = document.getElementById('checkout-cart');

const subtotalElement = document.getElementById('subtotal');

const proceedButton = document.getElementById('proceed');

// Simulated cart data (in practice, fetch from a backend or local storage)

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

let subtotal = 0;

cart.forEach(item => {

const cartItem = document.createElement('div');

cartItem.innerHTML = `

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

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

<p>Quantity: 1</p>

`;

checkoutCartContainer.appendChild(cartItem);

subtotal += item.price;

});

subtotalElement.textContent = `Subtotal: $${subtotal.toFixed(2)}`;

proceedButton.addEventListener('click', () => {

alert('Proceed to buy - Not Implemented');

});

});

* **document.addEventListener('DOMContentLoaded', () => {...});**: Runs the code inside once the DOM is fully loaded.
* **const checkoutCartContainer = document.getElementById('checkout-cart');**: Gets the element to display the checkout cart.
* **const subtotalElement = document.getElementById('subtotal');**: Gets the element to display the subtotal.
* **const proceedButton = document.getElementById('proceed');**: Gets the proceed button element.
* **const cart = JSON.parse(localStorage.getItem('cart')) || [];**: Retrieves the cart data from local storage.
* **cart.forEach(item => {...});**: Loops through each item in the cart and displays it.
* **subtotalElement.textContent = Subtotal: $${subtotal.toFixed(2)};**: Updates the subtotal.
* **proceedButton.addEventListener('click', () => {...});**: Adds an event listener to the proceed button.

1. **script.js** (Located in frontend/js/)

javascript

Copy code

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

const productsContainer = document.getElementById('products');

const cartItemsContainer = document.getElementById('cart-items');

const checkoutButton = document.getElementById('checkout');

let cart = [];

// Fetch products from the backend

fetch('http://localhost:5000/api/products')

.then(response => response.json())

.then(products => {

products.forEach(product => {

const productElement = document.createElement('div');

productElement.innerHTML = `

<h3>${product.name}</h3>

<p>${product.description}</p>

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

<button data-id="${product.\_id}">Add to Cart</button>

`;

productsContainer.appendChild(productElement);

});

// Add event listeners to "Add to Cart" buttons

productsContainer.querySelectorAll('button').forEach(button => {

button.addEventListener('click', addToCart);

});

});

// Add product to cart

function addToCart(event) {

const productId = event.target.getAttribute('data-id');

fetch(`http://localhost:5000/api/products`)

.then(response => response.json())

.then(products => {

const product = products.find(p => p.\_id === productId);

cart.push(product);

renderCart();

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

});

}

// Render the shopping cart

function renderCart() {

cartItemsContainer.innerHTML = '';

cart.forEach(item => {

const cartItem = document.createElement('li');

cartItem.textContent = `${item.name} - $${item.price}`;

cartItemsContainer.appendChild(cartItem);

});

}

// Handle checkout

checkoutButton.addEventListener('click', () => {

window.location.href = 'checkout.html';

});

});

* **document.addEventListener('DOMContentLoaded', () => {...});**: Runs the code inside once the DOM is fully loaded.
* **const productsContainer = document.getElementById('products');**: Gets the element to display the products.
* **const cartItemsContainer = document.getElementById('cart-items');**: Gets the element to display the cart items.
* **const checkoutButton = document.getElementById('checkout');**: Gets the checkout button element.
* **fetch('http://localhost:5000/api/products')...**: Fetches the products from the backend.
* **products.forEach(product => {...});**: Loops through each product and displays it.
* **productsContainer.querySelectorAll('button').forEach(button => {...});**: Adds event listeners to the "Add to Cart" buttons.
* **function addToCart(event) {...}**: Adds a product to the cart when the "Add to Cart" button is clicked.
* **function renderCart() {...}**: Renders the cart items.
* **checkoutButton.addEventListener('click', () => {...});**: Redirects to the checkout page when the checkout button is clicked.

1. **checkout.html** (Located in frontend/)

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>Checkout Page</title>

<link rel="stylesheet" href="css/checkout.css">

</head>

<body>

<header>

<h1>Shopping Cart</h1>

</header>

<main>

<section id="checkout-cart"></section>

<aside id="summary">

<h2>Order Summary</h2>

<p id="subtotal">Subtotal: $0.00</p>

<button id="proceed">Proceed to Buy</button>

</aside>

</main>

<script src="js/checkout.js"></script>

</body>

</html>

* **<!DOCTYPE html>**: Declares the document type and version.
* **<html lang="en">**: Sets the language of the document to English.
* **<meta charset="UTF-8">**: Sets the character encoding of the document.
* **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Sets the viewport settings for responsive design.
* **<title>Checkout Page</title>**: Sets the title of the document.
* **<link rel="stylesheet" href="css/checkout.css">**: Links to the CSS stylesheet for the checkout page.
* **<header>...</header>**: Contains the header of the document.
* **<main>...</main>**: Contains the main content of the document.
* **<section id="checkout-cart"></section>**: Section for displaying the checkout cart.
* **<aside id="summary">...</aside>**: Section for displaying the order summary.
* **<script src="js/checkout.js"></script>**: Links to the JavaScript file for the checkout page.

1. **index.html** (Located in frontend/)

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>Simple E-commerce Store</title>

<link rel="stylesheet" href="css/styles.css">

</head>

<body>

<header>

<h1>Simple E-commerce Store</h1>

</header>

<main>

<section id="products"></section>

<section id="cart">

<h2>Shopping Cart</h2>

<ul id="cart-items"></ul>

<button id="checkout">Checkout</button>

</section>

</main>

<script src="js/script.js"></script>

</body>

</html>

* **<!DOCTYPE html>**: Declares the document type and version.
* **<html lang="en">**: Sets the language of the document to English.
* **<meta charset="UTF-8">**: Sets the character encoding of the document.
* **<meta name="viewport" content="width=device-width, initial-scale=1.0">**: Sets the viewport settings for responsive design.
* **<title>Simple E-commerce Store</title>**: Sets the title of the document.
* **<link rel="stylesheet" href="css/styles.css">**: Links to the CSS stylesheet for the main page.
* **<header>...</header>**: Contains the header of the document.
* **<main>...</main>**: Contains the main content of the document.
* **<section id="products"></section>**: Section for displaying products.
* **<section id="cart">...</section>**: Section for displaying the shopping cart.
* **<script src="js/script.js"></script>**: Links to the JavaScript file for the main page.

This project represents a simple e-commerce store with a backend built using Node.js, Express, and MongoDB, and a frontend built with plain HTML, CSS, and JavaScript. The backend handles product data and serves it through a RESTful API, while the frontend fetches and displays this data, allowing users to add products to a cart and proceed to checkout.