Description

Description

This document is full of designs and specifications for my current E-commerce project. This is a log of important details of the current state and steps taken up till this point.

1. Project Description

This project is a simple E-Commerce web application that simulates a basic online store. It includes user registration, login, authentication, access-controlled product display, and a logout feature. Users can view a product menu once logged in, and all data is securely handled through a backend powered by Node.js, Express, and MongoDB. The interface is built with HTML, CSS, and JavaScript.

2. Functional Requirements, Product Specifications, and Standards

Functional Requirements:

- User registration with **ENHANCED** form validation (5-character minimums)
- Secure login using JWT (JSON Web Tokens) with **REFINED** validation
- View-only product menu after authentication
- Logout functionality with confirmation prompt
- Preventing access to protected pages when not logged in

Product Specifications:

- Technologies: HTML, CSS, JavaScript (Frontend), Node.js, Express.js (Backend), MongoDB (Database)
- Authentication: JWT (stored in localStorage)
- API Endpoints: /register, /login
- Local development with Live Server and Thunder Client for API testing

Standards Followed:

- RESTful API design
- Separation of concerns (frontend/backend)
- Use of environment variables (.env) for security
- **ENHANCED** Input validation and error handling with real-time feedback
- Responsive and user-friendly interface design principles

Current Functional Requirements

- BBelow are functional requirements of my application up to this point that can be fully replicated. This represents the current state after implementing full frontend-backend integration with persistent shopping cart system, complete backend order processing system, ENHANCED LOGIN/REGISTRATION VALIDATION, and **ENHANCED CHECKOUT CONFIRMATION REFINEMENT**.

- ### User Interface Requirements

- A landing page (Home.html) with navigation links: "About", "Login", "Register"
- ENHANCED login page (login.html) with REFINED VALIDATION: o 5-character minimum username validation o 5-character minimum password validation o Real-time validation feedback with color-coded inputs o Dynamic error messages and submit button states o Loading states during authentication
- ENHANCED registration page (register.html) with COMPREHENSIVE VALIDATION: o 5-character minimum username validation o 5-character minimum password validation o Username format validation (alphanumeric and underscores only) o Character limits and duplicate username handling o Real-time validation with visual feedback o Clear requirement display and user guidance
- A menu page (shop.html) fully integrated with backend APIs showing dynamic products with complete shopping cart functionality and STREAMLINED CHECKOUT INTEGRATION
- A logout button on the shop.html that logs the user out and redirects to the login page
- ENHANCED alert and redirection behavior for unauthenticated access attempts
- Dynamic product loading from MongoDB database via GET /products API
- Interactive product grid with add-to-cart functionality connected to backend
- Shopping cart sidebar with real-time updates persisted in MongoDB
- Cart counter badge displaying total items synchronized with database
- Quantity controls and item removal options using backend APIs

- COMPLETE CHECKOUT BUTTON INTEGRATION - Fully functional "Proceed to Checkout" button with validation and seamless navigation
- Loading states and button disable functionality during API operations
- Comprehensive error handling with user-friendly messages
- Toast notifications for user actions
- **ENHANCED CHECKOUT PAGE (checkout.html) - REFINED CHECKOUT INTERFACE with enhanced validation:** o **City dropdown menu with 20 major US cities selection** o **Phone number validation with automatic formatting (XXX) XXX-XXXX** o **Numbers-only phone input with 10-digit requirement** o **Real-time validation feedback with color-coded inputs** o **Dynamic submit button states based on validation** o **Visual validation messages and progress indicators**
- ORDER CONFIRMATION PAGE (confirmation.html) - Order success and details display

Authentication & Access Control

- ENHANCED New users can register with refined validation requirements: o Username minimum 5 characters, maximum 30 characters o Password minimum 5 characters, maximum 100 characters o Username format restrictions (letters, numbers, underscores only) o Real-time validation feedback and error handling
- REFINED Passwords are securely hashed using bcrypt before storage
- ENHANCED Upon login, JWT validation includes: o 5-character minimum validation for both fields o Real-time input validation with visual feedback o Enhanced error messaging for failed attempts
- IMPROVED Pages such as shop.html check for the existence of the token before loading — if absent, redirect to login.html
- ENHANCED Logout clears the JWT from localStorage and redirects to login.html
- JWT middleware protects cart API endpoints
- Token validation with proper error handling for expired/invalid tokens
- JWT tokens automatically included in all API requests with Authorization Bearer headers
- JWT authentication protection on checkout and confirmation pages

Shopping Cart Management (FULLY INTEGRATED)

- Add products to cart with backend persistence via POST /cart/add
- Real-time cart counter updates synchronized with MongoDB
- Quantity increase/decrease using PUT /cart/update API endpoint
- Remove individual items using DELETE /cart/remove API endpoint

- Automatic total price calculation from backend cart data
- Cart data persistence across browser sessions and devices
- User-specific cart isolation using JWT authentication
- Loading states during cart operations
- Error handling for failed cart operations
- Empty cart state handling with user guidance
- Responsive cart sidebar interface
- Cart loads from GET /cart API on page load
- All cart operations now use backend APIs instead of localStorage
- COMPLETE CHECKOUT INTEGRATION - "Proceed to Checkout" button with full validation and seamless navigation to checkout.html

- ### Checkout & Order Processing (ENHANCED REFINEMENT COMPLETE)

- **ENHANCED checkout interface with refined shipping information form:** o **City selection via dropdown menu (20 major US cities)** o **Phone number field with automatic formatting and validation** o **Numbers-only phone input with 10-digit requirement** o **Real-time validation feedback with visual indicators** o **Dynamic submit button states and messaging**
- Order summary display with cart items, tax calculation, and shipping costs
- **Enhanced form validation for city selection and phone number format**
- Integration with POST /orders API endpoint for streamlined order processing
- Order confirmation page with complete order details display
- Order success messaging with order ID
- Print receipt functionality
- Continue shopping navigation
- Cart clearing after successful order completion
- STREAMLINED NAVIGATION - Complete integration from shop.html cart to checkout.html with validation
- PORT ALIGNMENT - All API calls properly configured to http://localhost:5000

Project Structure

```
E-Commerce
 ├── Front End
```

```
│  ├── Home.html
│  ├── Login.html (**ENHANCED WITH REFINED VALIDATION**)
│  ├── Register.html (**ENHANCED WITH COMPREHENSIVE VALIDATION**)
│  ├── Shop.html (FULLY INTEGRATED WITH COMPLETE CHECKOUT FLOW)
│  ├── Checkout.html (COMPLETE CHECKOUT INTERFACE)
│  └── Confirmation.html (ORDER CONFIRMATION PAGE)
├── Server.js (Complete with integrated cart + ORDER PROCESSING APIs)
├── .env ← MongoDB connection string
├── Package.json
 ├── Routes/
│ └── auth.js
└── Models/
 ├── User.js
 ├── Product.js
├── CartItem.js (FULLY INTEGRATED)
└── Order.js (FULLY IMPLEMENTED AND CONNECTED)
```

User Stories and End Goal Functionality Requirements

Below are the user stories of functionality for the end goal of my project. These detail the full requirements and goal of my project when finished completely.

Authentication & Account Management

As a new visitor, I want to register for an account so that I can shop and make purchases on the site.

As a returning customer, I want to log in to my account so that I can access my personalized shopping experience.

As a logged-in user, I want to log out securely so that my account information stays protected.

As a user, I want to be redirected to login if I try to access protected pages so that the site remains secure.

e a unique order number so that I can reference it if needed.

Navigation & User Experience

As a user, I want clear navigation between pages so that I can easily move around the site.

As a customer, I want a consistent, professional design so that I trust the site with my information.

As a shopper, I want responsive design so that I can shop on any device.

As a user, I want clear feedback when actions succeed or fail so that I understand what's happening.

Data Persistence

As a customer, I want my cart to persist during my session so that I don't lose items if I navigate away.

As a user, I want my account information stored securely so that I can trust the platform.

As a customer, I want my order history saved so that I can reference past purchases.

Epic-Level Stories (High-Level Features)

As a business owner, I want a complete online shopping simulation so that customers can experience the full e-commerce journey from browsing to purchase.

As a site administrator, I want user authentication and data security so that customer information is protected.

As a customer, I want a seamless shopping experience from product discovery through checkout so that buying is quick and easy.

These user stories cover your complete e-commerce vision and can guide development priorities. You could tackle them in phases - like completing all the product browsing stories first, then cart management, then checkout, etc.

Traceability Functional Requirements

Here are functional requirements/use cases specifically for my e-commerce project in traceability format.

FR-001: User Registration

Use Case: New User Registration

- **Actor:** New Customer
- **Precondition:** User has internet access and is on the registration page
- **Main Flow:**
    - User enters username and password
    - System validates input format
    - System checks if username already exists
    - System hashes password using bcrypt
    - System stores user data in MongoDB
    - System displays success message
    - System redirects to login page
- **Alternative Flow:** Username already exists → Display error message
- **Postcondition:** New user account created in database

FR-002: User Authentication

Use Case: User Login

- **Actor:** Registered Customer
- **Precondition:** User has valid account credentials
- **Main Flow:**
    - User enters username/password on login page
    - System validates credentials against database
    - System generates JWT token
    - System stores token in localStorage

    - System redirects to menu page
- **Alternative Flow:** Invalid credentials → Display error message
- **Postcondition:** User authenticated and has access to protected pages

FR-003: Session Management

Use Case: Access Control

- **Actor:** Any User
- **Precondition:** User attempts to access protected page
- **Main Flow:**
    - System checks for valid JWT token in localStorage
    - If token exists, allow access to page
    - If no token, redirect to login page
- **Postcondition:** Only authenticated users access protected content

FR-004: User Logout

Use Case: Secure Logout

- **Actor:** Authenticated User
- **Precondition:** User is logged in and on menu page
- **Main Flow:**
    - User clicks logout button
    - System displays confirmation dialog
    - User confirms logout
    - System removes JWT token from localStorage
    - System redirects to login page
- **Postcondition:** User session terminated securely

FR-005: Product Display

Use Case: View Product Catalog

- **Actor:** Authenticated Customer
- **Precondition:** User is logged in and on menu page
- **Main Flow:**
  - System loads product data from array/database
  - System displays products in grid layout
  - Each product shows: name, price, description, category
  - Products display with hover effects for better UX
- **Postcondition:** Customer can see available products

FR-006: Shopping Cart Management

Use Case: Add Items to Cart

- **Actor:** Authenticated Customer
- **Precondition:** User viewing product catalog
- **Main Flow:**
  - User clicks "Add to Cart" on desired product
  - System adds product to cart storage
  - System updates cart counter
  - System displays confirmation message
- **Alternative Flow:** Item already in cart → Update quantity
- **Postcondition:** Product added to customer's cart

FR-007: Cart Viewing

Use Case: View Shopping Cart

- **Actor:** Authenticated Customer

- **Precondition:** User has items in cart
- **Main Flow:**
  - User navigates to cart page
  - System displays all cart items
  - System shows: product name, quantity, unit price, total price
  - System displays cart subtotal
- **Postcondition:** Customer sees complete cart contents

FR-008: Cart Modification

Use Case: Modify Cart Contents

- **Actor:** Authenticated Customer
- **Precondition:** User has items in cart and is viewing cart
- **Main Flow:**
  - User changes item quantity OR clicks remove
  - System updates cart storage
  - System recalculates totals
  - System updates display
- **Alternative Flow:** Quantity set to 0 → Remove item completely
- **Postcondition:** Cart reflects user's changes

FR-009: Checkout Process

Use Case: Complete Purchase

- **Actor:** Authenticated Customer
- **Precondition:** User has items in cart
- **Main Flow:**
  - User clicks "Checkout" button
  - System displays checkout form
  - User enters shipping information

    - User enters payment information (simulated)
    - User reviews order summary
    - User confirms purchase
    - System processes order (simulation)
    - System generates unique order ID
    - System stores order in database
    - System displays confirmation page
    - System clears cart
- **Postcondition:** Order completed and stored, cart emptied

FR-010: Order History

Use Case: View Past Orders

- **Actor:** Authenticated Customer
- **Precondition:** User has completed at least one order
- **Main Flow:**
    - User navigates to order history page
    - System retrieves user's orders from database
    - System displays orders with: order ID, date, items, total
    - User can click to view detailed order information
- **Postcondition:** Customer sees complete purchase history

FR-011: Navigation System

Use Case: Site Navigation

- **Actor:** Any User
- **Precondition:** User is on any page of the site
- **Main Flow:**
    - User sees consistent navigation menu
    - User clicks navigation link

    - System routes to appropriate page
    - System maintains authentication state during navigation
- **Postcondition:** User successfully navigates to desired page

FR-012: Data Persistence

Use Case: Maintain User Data

- **Actor:** System
- **Precondition:** User interactions occur
- **Main Flow:**
    - System stores user accounts in MongoDB
    - System maintains cart data during session
    - System persists order history permanently
    - System ensures data integrity and security
- **Postcondition:** All user data properly stored and retrievable

% Complete

Complete User Journey Now Available:

1. ENHANCED Registration/Login → User creates account and logs in WITH REFINED VALIDATION
2. Product Browsing → User views products loaded from database
3. Cart Management → User adds/modifies/removes items with real-time updates
4. **ENHANCED Checkout Flow → User clicks "Proceed to Checkout" with refined city dropdown and phone validation**
5. **REFINED Order Processing → User completes enhanced checkout form (city dropdown & validated phone) and places order**
6. Order Confirmation → User receives confirmation with order details

All Core Features Complete:

✅ ENHANCED User authentication and security with refined validation

✅ Product catalog with database integration

✅ Shopping cart with full backend integration

✅ **ENHANCED checkout process (city dropdown + phone validation)**

✅ Order confirmation and management

✅ Seamless navigation between all components

✅ ENHANCED error handling and user feedback

✅ Responsive design across all pages

✅ FIXED port alignment across all components

Recent Improvements Completed:

ENHANCEMENT #1 COMPLETE - Login/Registration Refinement:

- 5-character minimum validation for username and password
- Real-time validation with visual feedback
- Enhanced error messaging and user guidance
- Loading states and improved UX
- Additional security validations (character limits, format checking)
- Port Mismatch Resolution - All API calls aligned to localhost:5000
- Checkout Simplification - Removed payment processing, kept only shipping info
- Backend Streamlining - Simplified order processing without payment handling
- Database Model Updates - Made paymentInfo optional, simplified shippingInfo
- Enhanced Error Handling - Better API response handling and data structure flexibility

ENHANCEMENT #2 COMPLETE - Checkout Confirmation Refinement:

- **City dropdown menu with 20 major US cities (replacing freeform text input)**
- **Phone number validation with automatic formatting to (XXX) XXX-XXXX pattern**
- **Numbers-only phone input with exactly 10-digit requirement**
- **Real-time validation feedback with color-coded input borders**
- **Dynamic submit button states showing validation progress**
- **Enhanced UX with visual validation messages and progress indicators**

- **Consistent validation patterns matching login/registration enhancements**

Current Status: 100% complete - Enhanced Core Functionality Ready!

The e-commerce application now has refined, fully functional online store capabilities with complete end-to-end shopping flow (registration through order confirmation) using **ENHANCED CHECKOUT PROCESS WITH CITY DROPDOWN AND PHONE VALIDATION** plus **REFINED LOGIN/REGISTRATION VALIDATION**.