Final Project Demonstration Transcript

Purpose: Due to issues with my PC, my screencast demo does not possess sound, therefore, it will only contain a walkthrough of my applications functionality from beginning to end while also displaying my database which utilizes MongoDB Atlas.

E-Commerce Project Summary

Project Overview

This project is a fully functional e-commerce web application simulating an online store. It features user registration, secure login with JWT authentication, a dynamic product menu, a persistent shopping cart, and a streamlined checkout process. The application is built with a modern tech stack including HTML, CSS, and JavaScript for the frontend, Node.js and Express.js for the backend, and MongoDB for data storage. Key features include enhanced form validation, real-time user feedback, responsive design, and secure data handling, ensuring a seamless user journey from registration to order confirmation.

GUI Elements + Key Functionality

- **Home Page (Home.html)**: A landing page with navigation links to "About", "Login", and "Register".
- **Login Page (login.html)**: Enhanced with 5-character minimum validation for username and password, real-time color-coded feedback, dynamic error messages, and loading states during authentication.
- **Registration Page (register.html)**: Comprehensive validation including 5-character minimums, alphanumeric username restrictions, character limits, and real-time feedback.
- **Shop Page (shop.html)**: Displays a dynamic product grid loaded from MongoDB via GET /products API, with add-to-cart functionality, a cart counter badge, and a responsive cart sidebar for real-time updates.
- **Checkout Page (checkout.html)**: Features a refined interface with a dropdown for 20 major US cities, phone number validation with automatic (XXX) XXX-XXXX formatting, and real-time validation feedback.
- **Confirmation Page (confirmation.html)**: Displays order success details, order ID, and options to print receipts or continue shopping.
- **Key Functionality**:
    - Secure JWT-based authentication with tokens stored in localStorage.
    - Persistent shopping cart with backend integration (POST /cart/add, PUT /cart/update, DELETE /cart/remove).
    - Real-time cart updates and total price calculations.
    - Protected API endpoints and pages with JWT middleware.
    - Streamlined checkout process with enhanced validation and order processing via POST /orders API.

- Responsive design and comprehensive error handling with toast notifications.

Interesting Feature That Was Hard to Implement

The **order processing system** following the checkout process was the most challenging feature to implement, presenting significant errors and roadblocks during development. This feature required seamless integration of the shopping cart, checkout form, and order confirmation, all while ensuring data persistence and user feedback.

- **First Issue: Inability to Load Cart Items**
  - **Problem**: Initially, cart items failed to load correctly during checkout due to mismatches between the frontend and backend data structures.
  - **Resolution**: The database schema was remade multiple times to align the cart data structure with the MongoDB backend. This involved restructuring the CartItem.js model to ensure consistent data formats and updating API endpoints to handle cart data correctly.
- **Second Issue: Failure to Load Order Information**
  - **Problem**: Order details were not populating correctly on the confirmation page, often resulting in incomplete or missing data.
  - **Resolution**: Extensive debugging using browser console network logs identified issues in the POST /orders and GET /orders API routes. The authentication middleware and route handlers were reworked to ensure proper JWT validation and data retrieval. Real-time validation scripts were also refined to handle dynamic updates and prevent data loss during order submission.
- **Additional Challenges**: Other issues, though not fully recalled, included intermittent API failures and validation inconsistencies, which required iterative testing and refinement. The complexity of synchronizing cart data, validating user input, and ensuring order persistence nearly led to abandoning this feature. However, persistent debugging and schema adjustments ultimately enabled a robust, user-friendly order processing system.

Possible Enhancements and Extra Features

- **Order History Page**: Implement an order history page (as outlined in FR-010) allowing users to view past orders with details like order ID, date, items, and totals.
- **Product Search and Filtering**: Add search functionality and category filters to the shop page to enhance product discovery.
- **User Profile Management**: Introduce a user profile page where customers can update their account details, such as email or password.
- **Wishlist Feature**: Allow users to save products to a wishlist for future purchases, stored in MongoDB and accessible via a new API endpoint.
- **Email Notifications**: Integrate email confirmations for order placement and account registration using a service like Nodemailer.
- **Advanced Analytics**: Add tracking for user behavior (e.g., popular products, cart abandonment) to provide insights for the business owner.

Features Replaced/Tweaked/Removed

- **Registration and Login Validation**:
  - **Original**: Allowed single-character usernames and passwords, which posed security risks and lacked robustness.
  - **Change**: Enhanced to require a minimum of 5 characters for both fields, with alphanumeric restrictions for usernames, character limits (30 for usernames, 100 for passwords), and real-time validation feedback.
- **Payment Processing Simulation**:
  - **Original Plan**: Included a simulated payment processing step in the checkout flow.
  - **Change**: Removed due to time constraints and complexity, as it was not feasible to implement realistically in this application. The checkout process was simplified to focus on shipping information (city dropdown and phone validation), with paymentInfo made optional in the database schema.
- **Current Status**: No known bugs remain after these refinements.

Limitations

- **No Payment Processing**: The application lacks a payment processing system, limiting it to a simulated shopping experience rather than a fully operational store.
- **Limited City Options**: The checkout form restricts city selection to 20 major US cities, which may exclude some users.
- **No Order History**: While planned (FR-010), the order history feature is not yet implemented, preventing users from viewing past purchases.
- **Local Development Dependency**: The application relies on local development tools (Live Server, Thunder Client) and a fixed port (http://localhost:5000), which may complicate deployment.
- **No Multi-Language Support**: The interface and validation messages are English-only, potentially limiting accessibility for non-English-speaking users.
- **Scalability Constraints**: The current MongoDB setup and localStorage-based JWT storage may face performance issues with a large user base or high traffic.

Screencast Link: https://github.com/Darnae/Senior-Project-/tree/main

Github Portfolio Link: https://www.loom.com/share/f3a06be0a0214fe8bce16320c333674e?sid=cd3db3a2-929e-4799-a5a1-a2a13895303c