Updated Project Artifacts

The following artifacts have been updated to reflect the current state of the E-Commerce project, incorporating all prior instructor feedback and aligning with the completed enhancements detailed in the E-Commerce Status Report. The updates ensure that the Project Proposal, Requirements Specification, and Architecture Plan accurately represent the project's current functionality, including refined login/registration validation, enhanced checkout processes, and full backend integration.

Document History

Version 1.0 - Early April 2025 (April 1–3, 2025)

- **Description**: Initial project proposal, purpose, and designs.
- **Details**:
  - Created the initial project proposal outlining the purpose of a simple E-Commerce web application simulating an online store.
  - Included wireframes, flowcharts, and initial designs for user registration, login, authentication, product display, and logout features.
  - Defined core technologies: HTML, CSS, JavaScript (Frontend), Node.js, Express.js (Backend), MongoDB (Database).
  - Established the foundation for functional requirements and product specifications, such as user registration, secure login, and view-only product menu (aligned with Status Report Page 1: Project Description).

Version 2.0 - Mid-April 2025 (April 15, 2025)

- **Description**: Updated original specifications and designs.
- **Details**:
  - Refined the initial designs and specifications from Version 1.0.
  - Enhanced wireframes and flowcharts to include more detailed user flows for registration, login, and product display.
  - Added preliminary standards for RESTful API design and separation of concerns (frontend/backend) (aligned with Status Report Page 2: Standards Followed).

Version 3.0 - Late April 2025 (April 30, 2025)

- **Description**: Final update to initial designs.

- **Details**:
  - Finalized wireframes, flowcharts, and design specifications.
  - Incorporated enhanced input validation requirements and user-friendly interface principles.
  - Specified use of environment variables (.env) for security and local development setup with Live Server and Thunder Client for API testing (aligned with Status Report Page 2: Standards Followed and Page 1: Product Specifications).

Version 4.0 - May 15, 2025

- **Description**: Basic frontend pages implementation.
- **Details**:
  - Developed initial frontend pages: Home.html, About.html, and a hardcoded, empty product page (aligned with Status Report Page 5: Project Structure).
  - Implemented basic navigation links on the landing page (Home.html) for "About", "Login", and "Register" (aligned with Status Report Page 2: User Interface Requirements).
  - Minimal coding activity followed until July 13, 2025, focusing on design refinements and planning.

Version 5.0 - July 13, 2025

- **Description**: Initial database implementation.
- **Details**:
  - Set up MongoDB Atlas with a Users collection for login and registration functionality.
  - Implemented basic backend with Node.js and Express.js for user registration and login APIs (/register, /login) using JWT for authentication (aligned with Status Report Page 1: Product Specifications and Page 7: FR-001, FR-002).
  - Maintained hardcoded menu options for the product page, with no dynamic product loading yet.
  - Added basic authentication checks to prevent unauthorized access to protected pages (aligned with Status Report Page 3: Authentication & Access Control).

Version 6.0 - July 27, 2025

- **Description**: Expanded database with product collection.
- **Details**:
  - Added Products collection in MongoDB to replace hardcoded menu options.

    - Implemented dynamic product loading via GET /products API for the shop.html page, displaying products in a grid layout with name, price, description, and category (aligned with Status Report Page 2: User Interface Requirements and Page 9: FR-005).
    - Integrated add-to-cart functionality with localStorage for temporary cart storage (aligned with Status Report Page 9: FR-006).

Version 7.0 - August 3, 2025

- **Description**: Added order collection and enhanced functionality.
- **Details**:
    - Created Orders collection in MongoDB for order processing and storage.
    - Implemented POST /orders API endpoint for order processing and introduced the Order.js model (aligned with Status Report Page 5: Project Structure and Page 11: FR-009).
    - Added CartItem.js model for full shopping cart integration with backend persistence via POST /cart/add, PUT /cart/update, and DELETE /cart/remove APIs (aligned with Status Report Page 3: Shopping Cart Management).
    - Enhanced checkout flow with basic shipping information form and order confirmation page (confirmation.html) (aligned with Status Report Page 3: Checkout & Order Processing).

Version 8.0 - August 9, 2025

- **Description**: Final enhancements and testing.
- **Details**:
    - Completed enhanced login/registration validation with 5-character minimums, real-time feedback, color-coded inputs, and dynamic error messages (aligned with Status Report Page 2: User Interface Requirements and Page 13: ENHANCEMENT #1).
    - Refined checkout interface with a city dropdown menu (20 major US cities), phone number validation with automatic (XXX) XXX-XXXX formatting, and real-time validation feedback (aligned with Status Report Page 3: Checkout & Order Processing and Page 13: ENHANCEMENT #2).
    - Streamlined navigation and port alignment to http://localhost:5000 for all API calls (aligned with Status Report Page 4: PORT ALIGNMENT).
    - Implemented full cart persistence, user-specific cart isolation, and comprehensive error handling with toast notifications (aligned with Status Report Page 3: Shopping Cart Management and Page 4).
    - Finalized responsive design, enhanced error handling, and user feedback across all pages (aligned with Status Report Page 12: % Complete and Page 13: Recent Improvements).

    - Conducted end-to-end testing to ensure a complete user journey from registration to order confirmation (aligned with Status Report Page 12: Complete User Journey).

Current Status

As of August 9, 2025, the project is 100% complete with all core features fully implemented, including enhanced user authentication, dynamic product catalog, shopping cart management, refined checkout process, and order confirmation, as detailed in the status report (Page 14: Current Status).

Project Proposal

Overview

My project aims to develop a fully functional e-commerce web application simulating an online store. The application will provide a seamless user journey, including user registration, secure login, product browsing, shopping cart management, checkout, and order confirmation. Built with modern web technologies, the system ensures secure data handling, responsive design, and an intuitive user experience, meeting the needs of both customers and administrators.

Objectives

- Deliver a complete e-commerce experience with user authentication, product catalog, cart management, and streamlined checkout.
- Ensure robust security through JWT-based authentication and bcrypt password hashing.
- Provide a responsive, user-friendly interface with real-time validation feedback and comprehensive error handling.
- Persist user data (accounts, carts, orders) securely in MongoDB with full backend integration.

- Achieve 100% completion of core features as outlined in the function

Scope

  - The project encompasses the development of a web-based e-commerce platform with the following core components:
- **Frontend**: Responsive pages built with HTML, CSS, and JavaScript for user interaction.
- **Backend**: Node.js and Express.js APIs with MongoDB for data management.
- **Key Features**:
  - Enhanced user registration and login with 5-character minimum validation, real-time feedback, and format restrictions.
  - Dynamic product catalog loaded from MongoDB with add-to-cart functionality.
  - Persistent shopping cart with real-time updates, quantity controls, and item removal.
  - Streamlined checkout with city dropdown (20 major US cities), phone number validation ((XXX) XXX-XXXX), and order confirmation.
  - Secure access control with JWT middleware and redirects for unauthenticated users.
  - Comprehensive error handling with toast notifications and user-friendly messages.
- **Out of Scope**: Advanced product filtering, order history page, and payment processing (simulated checkout only).

Milestones

- **Fully Completed**: Full frontend-backend integration, enhanced login/registration validation, shopping cart persistence, streamlined checkout with city dropdown and phone validation, and order confirmation.
- **Future Enhancements**:
  - Potential addition of order history page (FR-010) and advanced product filtering (out of current scope).
  - Variety in shop products and category filtering and formatting per product type
  - More advanced security (more secure passwords and stronger usernames. Also possible email validation as well).

Requirements Specification

Final Traceability

  - The following functional requirements define the core capabilities of the e-commerce application, based on the provided status report:

- **FR-001: User Registration**
  - Users can register with a username (5-30 characters, alphanumeric and underscores only) and password (5-100 characters).
  - Real-time validation with color-coded feedback and dynamic submit button states.
  - Passwords hashed with bcrypt and stored in MongoDB.
  - Duplicate username handling with clear error messages.
- **FR-002: User Authentication**
  - Login requires valid credentials with 5-character minimum validation for username and password.
  - Real-time validation feedback and enhanced error messaging for failed attempts.
  - Generates JWT token stored in localStorage, redirecting to shop.html upon success.
- **FR-003: Session Management**
  - Protected pages (shop.html, checkout.html, confirmation.html) check for valid JWT token; redirect to login.html if absent.
  - JWT middleware secures cart and order APIs.
- **FR-004: User Logout**
  - Logout button on shop.html clears JWT from localStorage and redirects to login.html with confirmation prompt.
- **FR-005: Product Display**
  - Authenticated users view products in a grid layout (name, price, description, category) loaded via GET /products API.
  - Hover effects enhance user experience.
- **FR-006: Shopping Cart Management**
  - Users add products to cart via POST /cart/add API.
  - Real-time cart counter updates and persistence in MongoDB.
  - Quantity updates via PUT /cart/update and item removal via DELETE /cart/remove.
- **FR-007: Cart Viewing**
  - Cart sidebar displays items, quantities, unit prices, and subtotal, loaded via GET /cart API.
  - Responsive design with empty cart state handling.
- **FR-008: Cart Modification**
  - Users modify quantities or remove items, with real-time updates and total recalculation.
  - Zero quantity removes item from cart.
- **FR-009: Checkout Process**
  - "Proceed to Checkout" button navigates to checkout.html with validation.
  - Shipping form includes city dropdown (20 major US cities) and phone number validation ((XXX) XXX-XXXX, 10 digits).
  - Real-time feedback with color-coded inputs and dynamic submit button states.

  - Order processed via POST /orders API, generating unique order ID and clearing cart.
- **FR-010: Order History** (Future Scope)
  - Users view past orders with order ID, date, items, and total (not yet implemented).
- **FR-011: Navigation System**
  - Consistent navigation menu across all pages, maintaining authentication state.
  - Seamless transitions between Home.html, Login.html, Register.html, Shop.html, Checkout.html, and Confirmation.html.
- **FR-012: Data Persistence**
  - User accounts, cart data, and orders stored securely in MongoDB.
  - Cart persists across sessions; orders saved permanently.

Non-Functional Requirements

- **Security**:
  - JWT-based authentication for secure access control.
  - Bcrypt for password hashing.
  - Environment variables (.env) for sensitive data like MongoDB connection strings.
- **Usability**:
  - Responsive design for compatibility across devices (desktop, tablet, mobile).
  - Real-time validation feedback with color-coded inputs and dynamic submit button states.
  - Toast notifications and user-friendly error messages for all user actions.
- **Performance**:
  - Efficient database queries for product and cart data retrieval.
  - Loading states during API calls to enhance user experience.
- **Standards**:
  - RESTful API design for scalability and maintainability.
  - Separation of concerns between frontend and backend logic.
  - Adherence to responsive and user-friendly interface design principles.

Architecture Plan

System Architecture

  - 4.1 Technology Stack

- **Frontend**: HTML, CSS, JavaScript
  - Tools: Live Server for local development, Thunder Client for API testing.
- **Backend**: Node.js, Express.js
  - APIs: RESTful endpoints for registration, login, product retrieval, cart management, and order processing.
- **Database**: MongoDB
  - Models: User.js, Product.js, CartItem.js, Order.js.
- **Security**: JWT for authentication, bcrypt for password hashing, environment variables for configuration.

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 cart and checkout)
│   ├── Checkout.html (Refined with city dropdown and phone validation)
│   └── Confirmation.html (Order confirmation display)
├── Server.js (Integrated cart and 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)
```

4.3 Architecture Diagram

- **Client Layer**: Browser-based frontend (HTML, CSS, JavaScript) for user interaction.
- **Application Layer**: Node.js/Express.js server handling API requests and business logic.
- **Data Layer**: MongoDB for persistent storage of user accounts, products, carts, and orders.
- **Security Layer**: JWT middleware for protected routes, bcrypt for password security.

User Stories

The following user stories outline the end-goal functionality from the user's perspective:

- **Authentication & Account Management**:
  - As a new visitor, I want to register for an account to shop and make purchases.
  - As a returning customer, I want to log in to access my personalized shopping experience.
  - As a logged-in user, I want to log out securely to protect my account.
  - As a user, I want to be redirected to login if I try to access protected pages for security.
- **Shopping & Cart Management**:
  - As a shopper, I want to browse products with details (name, price, description, category).
  - As a customer, I want to add, modify, or remove items from my cart with real-time updates.
  - As a customer, I want my cart to persist across sessions to avoid losing items.
- **Checkout & Order Processing**:
  - As a customer, I want a streamlined checkout process with validated shipping information.
  - As a buyer, I want to receive order confirmation with a unique order number.
  - As a customer, I want to continue shopping or print a receipt after order completion.
- **Navigation & User Experience**:
  - As a user, I want clear navigation to move easily between pages.
  - As a customer, I want a consistent, professional design to trust the site.
  - As a shopper, I want a responsive design to shop on any device.
  - As a user, I want clear feedback on actions to understand system responses.

- **Data Persistence**:
  - As a customer, I want my account information stored securely.
  - As a customer, I want my order history saved for future reference.

6. Milestones and Deliverables

6.1 Milestones

- **Milestone 1: Authentication & Navigation** (Completed)
  - Implement enhanced registration/login with 5-character validation, real-time feedback, and JWT authentication.
  - Develop Home.html with navigation links to About, Login, and Register pages.
  - Ensure secure session management with redirects for unauthenticated access.
- **Milestone 2: Product Catalog & Cart Management** (Completed)
  - Build shop.html with dynamic product grid loaded from MongoDB via GET /products API.
  - Implement shopping cart with add, update, and remove functionality using /cart/* APIs.
  - Ensure cart persistence and real-time updates with MongoDB integration.
- **Milestone 3: Checkout & Order Processing** (Completed)
  - Develop checkout.html with city dropdown (20 US cities) and phone number validation ((XXX) XXX-XXXX).
  - Implement order processing via POST /orders API with confirmation page (confirmation.html).
  - Streamline navigation from cart to checkout with full validation.
- **Milestone 4: Final Enhancements & Testing** (Completed)
  - Enhance error handling with toast notifications and user-friendly messages.
  - Align all API calls to http://localhost:5000.
  - Simplify checkout by removing payment processing and focusing on shipping info.
  - Test all components for responsiveness, security, and performance.

6.2 Deliverables

- **Source Code**: Complete frontend (HTML, CSS, JavaScript) and backend (Node.js, Express.js, MongoDB) codebase.
- **Documentation**: Updated project proposal, requirements specification, and architecture plan.
- **Testing Reports**: API test results using Thunder Client and usability testing for responsive design.
- **Deployment Instructions**: Guide for running the application locally with Live Server and MongoDB.

7. Timeline

Milestone	Tasks	Duration	Status
Authentication & Navigation	Implement registration, login, session management, navigation	2-3 w eeks	Completed
Product Catalog & Cart	Build product grid, cart management, backend APIs	3-4 weeks	Completed
Checkout & Order Processing	Develop checkout form, order APIs, confirmation page	2-3 weeks	Completed
Final Enhancements & Testing	Enhance validation, error handling, testing, and documentation	1 week	Completed

Total Duration: 4+ months started coding toward the latter weeks of Senior Project 1. Continued Development Part Time in-between semesters, then fully picked up development towards second week of Senior Project 2. (Fully Completed as of August 9th, 2025)

8. Resources

- **Team**: Single developer (student) responsible for frontend, backend, and database development.
- **Tools**:
  - **Development**: Visual Studio Code, Live Server, Thunder Client.
  - **Version Control**: Git (optional, not specified in status report).
  - **Database**: MongoDB (local or cloud-based with .env configuration).
- **Dependencies** (via Package.json):
  - Node.js, Express.js, MongoDB driver.
  - JWT for authentication, bcrypt for password hashing.
  - Environment variables (.env) for secure configuration.

9. Risks and Mitigation

- **Risk**: Inconsistent API port usage causing errors.
  - **Mitigation**: Standardized all API calls to http://localhost:5000.
- **Risk**: Complex checkout process overwhelming users.

    - **Mitigation**: Simplified checkout by removing payment processing and adding clear validation feedback.
- **Risk**: Security vulnerabilities in authentication or data storage.
    - **Mitigation**: Implemented JWT middleware, bcrypt hashing, and environment variables for sensitive data.

10. Success Criteria

- **Functional Success**:
    - 100% completion of core features (FR-001 to FR-009, FR-011, FR-012).
    - Seamless end-to-end user journey from registration to order confirmation.
    - Fully integrated frontend-backend system with MongoDB persistence.
- **Non-Functional Success**:
    - Responsive design across devices (desktop, tablet, mobile).
    - Secure data handling with no vulnerabilities in authentication or storage.
    - User-friendly interface with real-time feedback and error handling.
- **User Satisfaction**:
    - Intuitive navigation and consistent design.
    - Clear feedback for all user actions (success, failure, validation).
    - Streamlined checkout with minimal user friction.

11. Conclusion

The E-Commerce Web Application project delivers a fully functional online store with enhanced user authentication, dynamic product browsing, persistent cart management, and a streamlined checkout process. By leveraging modern web technologies (HTML, CSS, JavaScript, Node.js, Express.js, MongoDB) and adhering to secure, user-friendly design principles, the project meets all specified functional and non-functional requirements. The application is complete as of August 10, 2025, with potential for future enhancements such as order history and advanced product filtering and possibly tighter login/registration security.