

Planning

Okay, I have carefully reviewed and processed the information contained within the "Bookworm Assignment Evaluation Checklist" and the "Bookworm Assignment Description" documents. Based *solely* on these documents, here is a detailed description of the Bookworm application:

1. Purpose of Application

- **Rationale:** Derived from the Assignment Description, Section 1.1 "Purpose".
- **Description:** The purpose of the Bookworm application is to serve as an online bookstore website. It aims to provide consumers with a convenient, centralized platform accessible via the internet to search for and find books based on various criteria like author or category. This eliminates the need to visit multiple physical locations, thereby saving users time and effort.

2. Goal to Achieve and Who is it For

- **Rationale:** Derived from Assignment Description Sections 1.1 "Purpose" and 1.2 "Scope". The scope explicitly limits the goal for *this specific assignment*.
- **Description:**

- **Target Audience (Conceptual):** The application conceptually serves two main types of users:
 1. **Front-end Users (Customers):** These are the primary consumers who use the website to query book information, add books to a shopping cart, place orders, and potentially add reviews for books.
 2. **Administrators:** (Conceptually mentioned but out of scope for this assignment) Users responsible for background functions like managing book classifications, information (add/modify/delete), order details, and user reviews.
- **Goal of this Assignment:** The specific goal for this development assignment is to build the **main functions for the front-end user (customer)**. This involves creating the user interface (UI) based on provided mock-ups, ensuring clarity and consistency, and implementing the required customer-facing features using dynamic data from a database. Development of the administrator functions is explicitly excluded from the scope of this assignment. The candidate needs to implement the mandatory features and can optionally implement customer review display and submission for bonus points.

3. Conceptual Model

- **Rationale:** Primarily derived from the Assignment Description Section 6

"Database Diagram" and Section 2 "Site Map". These define the core entities, their relationships, and the main user-facing sections.

- **Description:** The Bookworm application is modeled as an e-commerce platform focused on selling books. The core conceptual entities and their relationships are:

- **user** : Represents registered customers (and potentially administrators, though not implemented in the front-end scope). Contains details like name, email, and hashed password. Users can place **orders** .
- **book** : The central product entity. Each book has a **title** , **summary** , **price** , **cover_photo** , belongs to one **category** , and is written by one **author** . Books can have associated **reviews** and potentially a **discount** .
- **category** : Classifies books (e.g., Fiction, Science). Contains **category_name** and **category_desc** .
- **author** : Represents the book's author. Contains **author_name** and **author_bio** .
- **discount** : Represents promotional pricing for a **book** . Contains **discount_price** , **start_date** , and **end_date** . A discount is linked to a specific **book** .

- **review** : Contains user feedback for a specific **book** . Includes **review_title** , **review_details** , **rating_start** (likely meant to be **rating_star**), and **review_date** . Linked to both a **book** and implicitly to the reviewing **user** (though the direct link isn't shown in the provided diagram, it's logically necessary and implied by authentication requirements).
- **order** : Represents a customer's purchase transaction. Contains **order_date** , **order_amount** , and is linked to the **user** who placed it.
- **order_item** : Represents a specific **book** within an **order** . Contains **quantity** , **price** (at the time of order), and links to both the **order** and the **book** .
- **Site Structure**: Users interact with this model through five main pages: **Home** (showing on-sale and featured books), **Shop** (listing all books with filtering/sorting), **Product** (detailed view of a single book), **Cart** (managing selected books before purchase), and **About** (static company information).

4. Event Storming (Inferred)

- **Rationale**: Inferred from the required features, user interactions described in the checklist, and functional requirements in the assignment description. This

simulates the flow of events and commands within the system from the customer's perspective.

- **Description:**

- **Commands (User Actions):**

- ViewHomePage
 - ViewShopPage
 - ViewProductPage (triggered by clicking a book card)
 - ViewCartPage
 - ViewAboutPage
 - NavigateToPage (via navigation bar)
 - SignIn (with email/password via popup)
 - SignOut
 - FilterBooks (by category, author, rating)
 - SortBooks (by on-sale, popularity, price low-high, price high-low)
 - ChangeItemsPerPage
 - GoToPageNumber (pagination)
 - SwitchFeaturedTab (Recommended/Popular)
 - ViewAllOnSale (navigates to Shop sorted by sale)
 - AddToCart (from product page, with quantity)
 - UpdateCartItemQuantity

- `RemoveCartItem` (by setting quantity to 0)
- `PlaceOrder`
- `AddReview` (Optional: submit title, details, rating)
- `FilterReviewsByStar` (on product page)
- `SortReviews` (by date newest/oldest)
- **Events (System Reactions / Facts):**
 - `HomePageDisplayed` (with On Sale slider, Featured tabs)
 - `ShopPageDisplayed` (with book list, filters, sorters, pagination)
 - `ProductPageDisplayed` (with details, reviews, forms)
 - `CartPageDisplayed` (with items, totals)
 - `AboutPageDisplayed`
 - `UserAuthenticated` (JWT issued)
 - `UserNameDisplayedInNav`
 - `UserSignedOut`
 - `BooksFiltered`
 - `BooksSorted`
 - `PaginationApplied`
 - `FeaturedTabSwitched`
 - `BookAddedToCart`

- `CartUpdated` (item count in nav bar changed)
- `CartItemQuantityChanged`
- `CartItemRemoved`
- `CartTotalRecalculated`
- `SignInPopupDisplayed`
- `OrderSuccessfullyPlaced` (DB updated, success message shown, redirect occurs)
- `OrderPlacementFailed` (e.g., item unavailable, message shown, item removed)
- `ReviewSuccessfullyAdded` (Optional: list reloaded, success message shown)
- `ReviewsFilteredByStar`
- `ReviewsSorted`
- `InvalidQuantityInputRejected`
- `AuthenticationFailed` (wrong credentials)
- **Read Models / Views:**
 - Navigation Bar (Links, Active Item Indicator, Cart Count, User Name/Sign In)
 - Home Page (On Sale Carousel, Featured Books Tabs/Grid)

- Shop Page (Book Grid/List, Filter Accordions, Sort Dropdown, Pagination Controls, Item Count Selector, Result Info Text)
- Product Page (Book Details Section, Customer Reviews List, Add To Cart Form, Write Review Form)
- Cart Page (Cart Item List, Cart Totals Display)
- Book Card (Summary view: Image, Title, Author, Price/Discount Price)
- **Policies (Key Business Rules):**
 - If a book has an active discount, display both original and discount price; otherwise, display only the book price.
 - If a book lacks a cover image, display a default image.
 - Cart item quantity must be between 1 and 8. Setting to 0 removes the item.
 - User must be signed in to place an order.
 - If an item becomes unavailable during checkout, notify the user and remove it from the cart.
 - Update cart item count in the navigation bar immediately after adding/updating/removing items.
 - Persist user authentication across page refreshes.
 - (Optional) Reload review list after a new review is successfully submitted.

5. Included Features

- **Rationale:** Synthesized from the detailed checklists in the Evaluation document and the functional requirements in the Assignment Description (Sections 4 & 5).
- **Description:**
 - **Common UI & Layout:**
 - Consistent Navigation Bar: Displays on all pages (Home, Shop, About, Cart links). Shows active page indicator. Includes Cart item count.
 - Layout: Clear, consistent design following mock-ups. Uses Bootstrap/Tailwind grid system. Includes Header and Footer. Responsive design is a plus point.
 - Book Card Display (UI): Shows Title, Cover Image (or default), Price (USD/\$), Author Name. Supports long titles while maintaining card height (plus point).
 - Book Card Display (Logic): Displays default image if no cover image exists. Shows discounted price alongside original price if an active discount applies, following mock-up format.
 - Price Display: Uses USD (\$) as the currency. Correctly displays book price and handles available discount prices.
 - Pagination: Allows selecting items per page (5, 15, 20, 25). Displays page number controls (Next/Previous, page numbers). Shows

pagination info text (e.g., "Showing 1-5 of 126 books"). Disables

Previous/Next controls appropriately on first/last pages.

- **Authentication:**

- Sign In: Popup triggered by Sign In button. Uses email and password fields. Handles successful login (displays full name in nav bar, persists session on refresh) and unsuccessful login (error handling for wrong credentials). Uses OAuth2 (Password flow) and JWT.
- Sign Out: Functionality to log the user out, removing their name from the nav bar.
- Authenticated State: Persists across page refreshes. Displays user's full name (first + last) in the navigation bar when logged in.

- **Home Page:**

- On Sale Section: Displays top 10 most discounted books (calculated as `book_price - discount_price`) in a slider/carousel (like Bootstrap's) with multiple items visible. Includes previous/next controls. Clicking a book opens its Product page. Includes a "View All" button linking to the Shop page, pre-sorted by "on sale".
- Featured Books Section: Displays in 2 tabs ("Recommended", "Popular") following the mock-up. Allows switching between tabs, with an indicator for the active tab.

- Recommended: Top 8 books by highest average rating star
(considering lowest final price as a tie-breaker).
- Popular: Top 8 books by most reviews (considering lowest final price as a tie-breaker).
- **Shop Page:**
 - Layout: Contains "Filter By" and "Book List" sections, following the mock-up.
 - Listing: Displays books using the Book Card component, incorporating Pagination.
 - Sorting: Provides 4 sort options (on sale, popularity, price: low to high, price: high to low). Default sort is "on sale". Sort logic matches definitions from Home Page features (On Sale, Popular) and standard price sorting.
 - Filtering (UI): Displays 3 filter sections (Category, Author, Rating Review) using Bootstrap accordion behavior. Categories (5) ordered by name. Authors (10) ordered by name. Ratings (1-5 stars).
 - Filtering (Logic): Allows filtering by clicking items in filter sections. Filters correctly by category, author, or minimum rating (equal or greater than selected star). Displays the applied filter criteria after the page title (e.g., "Filtered by Category #1").

- **Filter & Sort Interaction:** Filtering and sorting can be applied simultaneously. URLs should correctly reflect applied filters/sorts/pagination for potential deep linking/refresh.
- **Product Page:**
 - **Layout:** Contains 4 sections: Book Details, Customer Reviews list, Add to Cart form, Write a Review form (optional), following mock-ups.
 - **Book Details:** Displays full book information: cover photo, title, description, author name.
 - **Add to Cart Form:** Displays price (handling discounts correctly). Provides quantity input (validation: 1-8). Includes "Add to cart" button. Shows a success message upon adding to cart. Updates cart item count in the navigation bar.
 - **Customer Reviews List (Optional Feature):**
 - Displays average star rating (calculated as $(1*a+2*b+3*c+4*d+5*e) / (a+b+c+d+e)$).
 - Displays total review count per star rating (e.g., "5 star (200)").
 - Allows clicking on star counts to filter reviews by that rating (displays e.g., "Filtered by 5 star").
 - Includes pagination for reviews.

- Displays individual reviews with title, details, rating star, and created date (Month Date, Year format).
- Allows sorting reviews by date (newest to oldest - default, oldest to newest).
- Write a Review Form (Optional Feature):
 - Allows users to submit a review.
 - Title input (mandatory, max 120 length).
 - Details input (optional).
 - Rating star dropdown (mandatory, 1-5 options).
 - Shows a success message after submission.
 - Reloads the customer review list with the new review upon successful submission.
- **Cart Page:**
 - Layout: Contains 2 sections: Product list and Cart totals, following mock-up. Displays correct items added to the cart. Handles long book titles without breaking layout (plus point).
 - Cart Item List: Shows information for each item (image, title, author, price, quantity). Allows updating quantity per item (recalculates total). Removes item if quantity is set to 0. Enforces maximum quantity of 8 per item. Each item links to its Product page (opens in new tab).

- **Cart Totals:** Automatically calculates and displays the total amount for items in the cart.
- **Place Order:**
 - Displays Sign In popup if the user is not logged in.
 - If authenticated, clicking "Place order" inserts order and order_item data into the DB.
 - Shows a success message within 10 seconds, then redirects to the home page.
 - Handles cases where an item is no longer available: shows an unsuccessful message identifying the item(s) and automatically removes them from the cart.
- **About Page:**
 - Displays static text content following the mock-up (Welcome, Our Story, Our Vision).

6. User Stories

- **Rationale:** Translating the features and interactions described above into the standard user story format (As a [user type], I want to [action], so that [benefit]), focusing on the customer perspective.
- **Description:**

- As a customer, I want to see a consistent navigation bar on every page, so I can easily access Home, Shop, About, and my Cart.
- As a customer, I want the navigation bar to highlight the current page I'm on, so I know my location within the site.
- As a customer, I want to see the number of items in my cart displayed in the navigation bar, so I can track my selections easily.
- As a customer, I want the website layout to be clear, consistent, and responsive, so I can have a good experience on any device.
- As a customer browsing books (on Home or Shop), I want to see a clear card for each book showing its cover, title, author, and price, so I can quickly evaluate it.
- As a customer, I want to see a default image if a book cover is missing, so the layout remains consistent.
- As a customer, I want to clearly see if a book has a discount price and what the original price was, so I can recognize deals.
- As a customer, I want to sign in using my email and password via a popup, so I can access my account features securely.
- As a customer, I want to remain logged in even if I refresh the page, so I don't have to repeatedly authenticate.

- As a customer, I want to see my full name in the navigation bar when I'm logged in, so I know my session is active.
- As a customer, I want to be able to sign out, so I can end my authenticated session.
- As a customer on the Home page, I want to see a selection of books currently on sale presented in an easy-to-browse slider, so I can quickly find discounted items.
- As a customer on the Home page, I want to see featured books categorized as "Recommended" (high rating) and "Popular" (most reviews), so I can discover noteworthy books.
- As a customer, I want to click on any book card (Home, Shop) to view its detailed product page.
- As a customer on the Shop page, I want to browse through all available books using pagination, so I don't get overwhelmed by a single long list.
- As a customer on the Shop page, I want to choose how many books are displayed per page (5, 15, 20, or 25), so I can control the browsing density.
- As a customer on the Shop page, I want to sort the book list by relevance (on sale, popularity) or price, so I can find books according to my priorities.
- As a customer on the Shop page, I want to filter the book list by category, author, or minimum star rating, so I can narrow down the results to my

interests.

- As a customer on the Shop page, I want to see which filters are currently applied, so I understand the displayed book list.
- As a customer on the Product page, I want to see detailed information about the book including its description, author, and price.
- As a customer on the Product page, I want to add the book to my cart with a specific quantity (between 1 and 8), so I can purchase it.
- As a customer, I want to receive confirmation (a success message) after adding an item to the cart.
- (Optional) As a customer on the Product page, I want to read reviews left by other users, including their ratings and comments, so I can make a more informed purchase decision.
- (Optional) As a customer on the Product page, I want to see the average star rating for the book and how many reviews exist for each star level.
- (Optional) As a customer on the Product page, I want to filter reviews by star rating, so I can focus on specific feedback levels.
- (Optional) As a customer on the Product page, I want to sort reviews by date, so I can see the most recent or oldest feedback first.
- (Optional) As a customer, I want to write my own review (title, details, star rating) for a book on its product page, so I can share my opinion with

others.

- As a customer, I want to view my shopping cart to see all the items I've added, their quantities, and prices.
- As a customer in the Cart, I want to update the quantity of any item, so I can adjust my order before purchasing.
- As a customer in the Cart, I want to remove an item completely (by setting quantity to 0), so I can curate my final purchase list.
- As a customer in the Cart, I want to see the total cost of all items, so I know the amount before proceeding to checkout.
- As a customer in the Cart, I want to click a button to place my order.
- As a customer, I want to be prompted to sign in if I try to place an order while logged out.
- As a customer, I want to receive a success message after placing an order and be redirected to the home page.
- As a customer, I want to be notified if an item in my cart is unavailable when I try to place the order, and have it automatically removed, so I don't pay for something I won't receive.
- As a customer, I want to view an "About" page to learn more about the Bookworm company.

7. Project Requirements

- **Rationale:** Collated from Assignment Description Sections 1.2 (Scope), 1.3 (Technical Requirements), 4 (Common Functional Requirements), 5 (Page Functional Requirements), and the Evaluation Checklist details.
- **Description:**
 - **Functional Requirements:**
 - Implement all features described in section 5 ("Included Features") above, covering Home, Shop, Product, Cart, and About pages, plus common elements like Navigation, Authentication, Book Display, Price Display, and Pagination.
 - Adhere strictly to the logic defined for:
 - Discount calculation and display.
 - "On Sale" book selection (top 10 most discounted).
 - "Recommended" book selection (top 8 highest avg rating, lowest price tie-breaker).
 - "Popular" book selection (top 8 most reviews, lowest price tie-breaker).
 - Shop page sorting options (on sale, popularity, price asc/desc).
 - Shop page filtering (category, author, rating \geq selected).
 - Pagination display and logic.
 - Cart quantity validation (1-8, 0 for removal).

- Order placement logic (DB insertion, success/failure handling for unavailable items).
- Authentication flow (popup, success/failure, persistence, name display).
- (Optional) Customer review display (average calculation, filtering, sorting, pagination).
- (Optional) Review submission (validation, success message, list refresh).
- **Non-Functional Requirements:**
 - **UI/UX:** UI must follow the provided mock-ups for layout and appearance. The design should be clear, consistent, and user-friendly. Must provide feedback to the user (e.g., success messages, validation errors, loading states - implied).
 - **Responsiveness:** The website must be responsive and display correctly on various screen sizes. Achieving this using Bootstrap or TailwindCSS is required. Well-supported responsiveness is a plus point.
 - **Data:** Must use dynamic data fetched from the PostgreSQL database via API endpoints. Currency must be USD (\$). Data formatting for

dates (Month Date, Year for reviews) and prices must be consistent.

Default image must be used for books without covers.

- **Security:** Implement authentication using OAuth2 with Password (and hashing) and Bearer JWT tokens.
- **Performance:** UI interactions (slider, pagination, tab switching) should feel smooth. Success messages should appear promptly (e.g., within 10 seconds for order placement).
- **Technical Requirements:**
 - **Backend:** Use FastAPI framework to build the backend system providing API endpoints.
 - **Frontend:** Use ReactJs library to build the front-end system as a Single Page Application (SPA).
 - **Database:** Use PostgreSQL for the database, adhering to the provided schema.
 - **Styling/Responsiveness:** Use either TailwindCSS or Bootstrap.
 - **Authentication Implementation:** Use OAuth2 (Password flow with hashing) and JWT Bearer tokens for securing API communication and managing user sessions.
- **Scope:**

- **In Scope:** All front-end user features as detailed above (mandatory and optional). Manual database manipulation (insert/update) is expected for testing data display as the admin interface is not built.
- **Out of Scope:** Administrator back-end functions and UI.
- **Optional (Bonus):** Implementation of the Customer Reviews list display features and the Submit Review form functionality on the Product page.

I am approaching the token limit. Please let me know if you'd like me to continue with any further elaboration or refinement in the next message.