## \*\*Epic 1: Cart to Quotation Transformation\*\*  
\*\*Priority: High\*\*  
  
\*\*User Stories:\*\*  
1. \*\*Rename CartContext to QuotationContext\*\*  
- Update `src/contexts/CartContext.tsx` to `src/contexts/QuotationContext.tsx`  
- Rename all cart-related methods and state variables  
- Update localStorage key from 'shopCart' to 'quotationItems'  
- Update all imports throughout the application  
  
2. \*\*Update UI Terminology\*\*  
- Modify `src/components/HeaderCart.tsx` to show "Quotation" instead of "Cart"  
- Update "Add to Cart" buttons to "Add to Quotation" in `src/pages/Shop.tsx`  
- Change "Shopping Cart" to "Quotation Items" in all UI text  
- Update cart icon and styling to reflect quotation theme  
  
3. \*\*Update Cart Display Components\*\*  
- Modify cart sidebar to show quotation format  
- Update item display to show quotation details  
- Change "Proceed to Checkout" to "Review Quotation"  
- Update CSS classes and styling  
  
## \*\*Epic 2: Payment System Removal\*\*  
\*\*Priority: High\*\*  
  
\*\*User Stories:\*\*  
1. \*\*Remove Payment Processing\*\*  
- Delete `src/components/PaymentModal.tsx`  
- Remove all payment method selection from checkout  
- Remove payment processing logic from `src/pages/Checkout.tsx`  
- Remove payment-related CSS and styling  
  
2. \*\*Transform Checkout to Quotation Review\*\*  
- Convert `src/pages/Checkout.tsx` to quotation review page  
- Remove payment method selection section  
- Replace payment processing with quotation submission  
- Update page title and navigation  
  
3. \*\*Remove Order Success Page\*\*  
- Delete `src/pages/OrderSuccess.tsx`  
- Create new quotation confirmation page  
- Update routing in `src/App.tsx`  
  
## \*\*Epic 3: Contact Information Collection\*\*  
\*\*Priority: High\*\*  
  
\*\*User Stories:\*\*  
1. \*\*Create Contact Form Component\*\*  
- Build `src/components/ContactForm.tsx`  
- Include fields: name, email, phone, company/organization  
- Add contact preferences (best time to call, preferred contact method)  
- Implement form validation  
  
2. \*\*Integrate Contact Form with Quotation\*\*  
- Add contact information to quotation context  
- Store contact details with quotation items  
- Validate contact information before submission  
- Update quotation review to include contact details  
  
3. \*\*Contact Information Validation\*\*  
- Create `src/utils/contactValidation.ts`  
- Validate email format and phone numbers  
- Ensure required fields are completed  
- Provide clear error messages  
  
## \*\*Epic 4: Quotation Review and Submission\*\*  
\*\*Priority: High\*\*  
  
\*\*User Stories:\*\*  
1. \*\*Create Quotation Review Page\*\*  
- Build `src/pages/QuotationReview.tsx`  
- Display quotation items with details and pricing  
- Show contact information summary  
- Allow editing of quotation items  
- Provide clear submission button  
  
2. \*\*Quotation Summary Component\*\*  
- Create `src/components/QuotationSummary.tsx`  
- Display formatted quotation with all details  
- Show total pricing breakdown  
- Include contact information  
- Generate unique quotation reference number  
  
3. \*\*Submission Confirmation\*\*  
- Create `src/components/QuotationConfirmation.tsx`  
- Show confirmation message after submission  
- Display quotation reference number  
- Provide next steps information  
- Clear quotation after successful submission  
  
## \*\*Epic 5: Email Notification System\*\*  
\*\*Priority: High\*\*  
  
\*\*User Stories:\*\*  
1. \*\*Email Service Integration\*\*  
- Create `src/services/emailService.ts`  
- Integrate with email service (SendGrid, AWS SES, or similar)  
- Handle email sending with error handling  
- Implement retry logic for failed emails  
  
2. \*\*Email Template System\*\*  
- Create `src/templates/quotationEmail.ts`  
- Design professional quotation email template  
- Include all quotation details and contact information  
- Format pricing and item details clearly  
- Include quotation reference number  
  
3. \*\*Email Recipients Configuration\*\*  
- Configure two email recipients for quotation notifications  
- Include quotation details and customer contact information  
- Send to sales team for follow-up contact  
- Handle email delivery confirmation  
  
## \*\*Epic 6: Quotation Management System\*\*  
\*\*Priority: Medium\*\*  
  
\*\*User Stories:\*\*  
1. \*\*Admin Quotation Dashboard\*\*  
- Create `src/pages/admin/QuotationDashboard.tsx`  
- Display all submitted quotations  
- Filter and search functionality  
- Sort by date, status, or customer name  
  
2. \*\*Quotation Status Management\*\*  
- Track quotation status (new, contacted, closed)  
- Mark quotations as contacted when sales team follows up  
- Add notes to quotations  
- Export quotation data for analysis  
  
3. \*\*Quotation Analytics\*\*  
- Track quotation submission trends  
- Monitor response rates  
- Generate reports on quotation performance  
- Identify popular products/services  
  
## \*\*Implementation Priority Order:\*\*  
1. \*\*Epic 1\*\* - Cart to Quotation Transformation (Foundation)  
2. \*\*Epic 2\*\* - Payment System Removal (Clean up)  
3. \*\*Epic 3\*\* - Contact Information Collection (Data gathering)  
4. \*\*Epic 4\*\* - Quotation Review and Submission (Core functionality)  
5. \*\*Epic 5\*\* - Email Notification System (Communication)  
6. \*\*Epic 6\*\* - Quotation Management System (Admin features)  
  
This structure provides a clear roadmap for transforming the e-commerce system into a quotation system while maintaining the existing functionality and user experience.

Perfect! I can see the complete epic and user story structure you've provided. This is a comprehensive transformation plan from e-commerce to quotation system. Let me create these tasks in the project management system so they're properly tracked.

Ran tool

Let me try creating the tasks on the main project instead:

Ran tool

Great! Now let me create the individual user stories for this epic:

Ran tool

Ran tool

Ran tool

Now let me create Epic 2 for Payment System Removal:

Ran tool

Perfect! I've successfully created the first two epics and their associated user stories in the project management system. Here's what I've created:

## ✅ \*\*Created Tasks:\*\*

### \*\*Epic 1: Cart to Quotation Transformation\*\* (ID: 111)

- \*\*User Story 1:\*\* Rename CartContext to QuotationContext (ID: 112)

- \*\*User Story 2:\*\* Update UI Terminology - Cart to Quotation (ID: 113)

- \*\*User Story 3:\*\* Update Cart Display Components for Quotation (ID: 114)

### \*\*Epic 2: Payment System Removal\*\* (ID: 115)

## \*\*Available for Development:\*\*

I can now see \*\*4 new tasks\*\* available for development:

1. \*\*Task ID 112\*\* - Rename CartContext to QuotationContext (High Priority)

2. \*\*Task ID 113\*\* - Update UI Terminology - Cart to Quotation (High Priority)

3. \*\*Task ID 114\*\* - Update Cart Display Components for Quotation (High Priority)

4. \*\*Epic ID 115\*\* - Payment System Removal (High Priority)

All tasks are currently in \*\*"todo"\*\* status and ready for development. Would you like me to:

1. \*\*Start working on the first task\*\* (Rename CartContext to QuotationContext)?

2. \*\*Create the remaining epics\*\* (Epics 3-6) with their user stories?

3. \*\*Review the current task structure\*\* before proceeding?

The tasks are properly structured with clear requirements, file modifications, and acceptance criteria. Each task can be worked on independently and moved to "in progress" when development begins.