Plaquita.com Design Editor

Improvements & Missing Features Report

Document Version: 1.0
Date: 2024
Project: Custom T-Shirt Design Editor

Executive Summary

This document outlines the key improvements needed and missing features for the Plaquita.com Custom T-Shirt Design Editor. The recommendations are organized by priority and impact to help guide development efforts and resource allocation.

Table of Contents

Critical Improvements

Security Issues

- **API Keys Exposed in Source Code**
  - WooCommerce API keys are currently hardcoded in the source code
  - This is a critical security vulnerability
  - Keys should be moved to environment variables immediately
  - Add environment variable validation
  - Create secure configuration management
- **Input Validation Missing**
  - No validation for file uploads (type, size)
  - User input not sanitized properly
  - API responses not validated
  - Need rate limiting for API calls
- **XSS Prevention**
  - SVG content not sanitized before rendering
  - Need Content Security Policy headers
  - Validate all external content sources

Architecture Issues

- **Monolithic Component Structure**
  - Main DesignEditor component is approximately 4,700 lines
  - Contains UI rendering, canvas logic, business logic, and state management
  - Violates Single Responsibility Principle
  - Makes code difficult to test, maintain, and extend
  - Needs to be broken down into smaller, focused components
- **Excessive Prop Drilling**
  - SideMenu component receives 30+ props
  - State management scattered across multiple useState hooks
  - Makes components tightly coupled and fragile
  - Difficult to add new features without breaking existing code

---

High Priority Improvements

State Management

- **Implement Context API**
  - Replace prop drilling with React Context
  - Centralize design-related state management
  - Improve component reusability
  - Reduce coupling between components
- **Use Reducer Pattern**
  - Implement useReducer for complex state logic

    - Better state management for undo/redo functionality
    - More predictable state updates
    - Easier to debug state changes

Performance Issues

- **Canvas Rendering Performance**
    - Heavy re-renders on every state change
    - Canvas redraws on every keystroke
    - No debouncing for text input updates
    - Multiple canvas instances initialized unnecessarily
    - Need to optimize rendering cycles
- **Memory Leaks**
    - Event listeners not properly cleaned up
    - Multiple canvas instances may cause memory issues
    - Large images kept in memory unnecessarily
    - Need proper cleanup on component unmount

Error Handling

- **Missing Error Boundaries**
    - No error boundaries to catch React errors
    - Application crashes affect entire user experience
    - Need graceful error handling and recovery
    - User-friendly error messages required
- **API Error Handling**
    - No comprehensive error handling for API calls
    - Network errors not handled gracefully
    - Server errors not communicated to users
    - Need retry mechanisms for failed requests

Medium Priority Improvements

Code Organization

- **File Structure**
    - Flat component structure
    - No clear separation of concerns
    - Utilities mixed with components
    - Need organized folder structure with clear responsibilities
- **Code Duplication**
    - Similar functions with different names (handleObjectScaling vs handleObjectScaling1)
    - Repeated canvas setup code for different views
    - Similar event handlers across multiple components
    - Need to extract common logic into reusable utilities

Styling Consistency

- **Mixed Styling Approaches**
    - Combination of inline styles, styled-components, and Tailwind CSS
    - Inconsistent spacing and colors throughout application
    - Need to standardize on one primary styling approach
    - Create design system with consistent tokens

Component Structure

- **Extract Custom Hooks**
    - Canvas initialization and lifecycle management
    - Object manipulation logic
    - Text editing functionality
    - Image processing operations
    - Clip art management
- **Split UI Components**
    - Separate canvas wrapper from logic
    - Extract toolbar components
    - Create dedicated property panels
    - Organize dialog components

Missing Features

Core Functionality

- **Undo/Redo System**
    - Currently not implemented

    - Essential for user experience
    - Users expect this in any design tool
    - Need command pattern implementation
    - History management required

- **Keyboard Shortcuts**
    - No keyboard shortcuts available
    - Standard shortcuts expected (Ctrl+Z, Delete, Copy, Paste)
    - Improve workflow efficiency
    - Need comprehensive shortcut system

Design Features

- **Design Templates Library**
    - No pre-designed templates available
    - Users must start from scratch
    - Templates would improve user onboarding
    - Need template categories (sports, business, casual)
    - Allow saving custom templates

- **Advanced Text Features**
    - Limited text effects available
    - No text shadows or glows
    - Missing 3D text effects
    - No text alignment options beyond basic
    - Limited text shapes
    - Need text along path feature

- **Image Editing Tools**
    - No crop functionality
    - No resize tools
    - Missing filters and effects
    - No image adjustments (brightness, contrast, saturation)
    - Need comprehensive image editing suite

Export & Save Features

- **Export Options**
    - Limited export formats
    - No high-resolution export option
    - Missing print-ready formats
    - No batch export capability
    - Need multiple format support (PNG, SVG, PDF, JPG)

- **Design History & Saving**
    - No local save functionality
    - No cloud save option
    - Missing design library/gallery
    - No recent designs feature
    - Need persistent storage solution

Collaboration Features

- **Sharing Capabilities**
    - Cannot share designs
    - No collaboration features
    - Missing comments and annotations
    - No version history
    - Need sharing via link functionality

Advanced Tools

- **Advanced Color Tools**
    - Basic color picker only
    - No eyedropper tool
    - Missing color palette generator
    - No color harmony tools
    - Limited custom color swatches

- **Alignment & Distribution Tools**
    - No snap-to-grid option
    - Missing alignment guides
    - No object distribution tools
    - Need smart guides when dragging

User Experience Enhancements

Loading & Feedback

- **Loading States**

    - No loading indicators during image upload
    - Missing progress bars for bulk operations
    - No feedback during async operations
    - Need skeleton loaders for content
    - Disable buttons during processing

- **User Feedback**
    - No success notifications
    - Missing error messages
    - No toast notifications
    - Need clear action confirmations

Mobile Experience

- **Responsive Design**
    - Mobile experience needs significant improvement
    - Sidebar behavior problematic on mobile
    - Canvas scaling issues on small screens
    - Need touch gesture support (pinch-to-zoom, pan)
    - Require mobile-optimized toolbar

- **Touch Interactions**
    - Limited touch support
    - No gesture recognition
    - Need touch-friendly controls
    - Require adaptive UI for mobile devices

Drag & Drop

- **Visual Feedback**
    - No visual feedback during drag operations
    - Missing snap indicators
    - Need alignment guides
    - Require duplicate on drag option

Performance Improvements

Optimization Needs

- **Canvas Performance**
    - Debounce text input updates
    - Optimize canvas rendering cycles
    - Use requestAnimationFrame for smooth updates
    - Implement object caching for static elements
    - Lazy load canvas instances

- **Component Optimization**
    - Memoize expensive components
    - Optimize re-renders
    - Use React.memo for pure components
    - Memoize callbacks and computed values

- **Asset Optimization**
    - Compress SVG cliparts
    - Optimize PNG assets
    - Implement lazy loading for cliparts
    - Code splitting for routes
    - Optimize font loading

Memory Management

- **Cleanup Required**
    - Properly dispose canvas on unmount
    - Clean up event listeners
    - Release image memory
    - Manage multiple canvas instances efficiently

Security Improvements

Immediate Actions Required

- **API Key Security**
    - Move all API keys to environment variables
    - Never commit keys to version control
    - Implement key rotation policy
    - Add key validation

- **Input Sanitization**

    - Validate all file uploads
    - Sanitize user input
    - Validate API responses
    - Implement rate limiting
- **Content Security**
    - Sanitize SVG content
    - Implement CSP headers
    - Validate external content
    - Secure image processing