

# CollabCanvas Rubric

Total Points: 100

## Section 1: Core Collaborative Infrastructure (30 points)

### Real-Time Synchronization (12 points)

**Excellent (11-12 points)**

* Sub-100ms object sync
* Sub-50ms cursor sync
* Zero visible lag during rapid multi-user edits

**Good (9-10 points)**

* Consistent sync under 150ms
* Occasional minor delays with heavy load

**Satisfactory (6-8 points)**

* Sync works but noticeable delays (200-300ms)
* Some lag during rapid edits

**Poor (0-5 points)**

* Inconsistent sync
* Frequent delays over 300ms
* Broken under concurrent edits

### Conflict Resolution & State Management (9 points)

**Excellent (8-9 points)**

* Two users edit same object simultaneously → both see consistent final state
* Documented strategy (last-write-wins, CRDT, OT, etc.)
* No "ghost" objects or duplicates
* Rapid edits (10+ changes/sec) don't corrupt state
* Clear visual feedback on who last edited

**Good (6-7 points)**

* Simultaneous edits resolve correctly 90%+ of time
* Strategy documented
* Minor visual artifacts (brief flicker) but state stays consistent
* Occasional ghost objects that self-correct

**Satisfactory (4-5 points)**

* Simultaneous edits sometimes create duplicates
* Strategy unclear or undocumented
* State inconsistencies require refresh
* No indication of edit conflicts

**Poor (0-3 points)**

* Simultaneous edits frequently corrupt state
* Objects duplicate or disappear
* Different users see different canvas states
* Requires manual intervention to fix

**Testing Scenarios for Conflict Resolution:**

1. Simultaneous Move: User A and User B both drag the same rectangle at the same time
2. Rapid Edit Storm: User A resizes object while User B changes its color while User C moves it
3. Delete vs Edit: User A deletes an object while User B is actively editing it
4. Create Collision: Two users create objects at nearly identical timestamps

### Persistence & Reconnection (9 points)

**Excellent (8-9 points)**

* User refreshes mid-edit → returns to exact state
* All users disconnect → canvas persists fully
* Network drop (30s+) → auto-reconnects with complete state
* Operations during disconnect queue and sync on reconnect
* Clear UI indicator for connection status

**Good (6-7 points)**

* Refresh preserves 95%+ of state
* Reconnection works but may lose last 1-2 operations
* Connection status shown
* Minor data loss on network issues

**Satisfactory (4-5 points)**

* Refresh loses recent changes (last 10-30 seconds)
* Reconnection requires manual refresh
* Inconsistent persistence
* No clear connection indicators

**Poor (0-3 points)**

* Refresh loses significant work
* Reconnection fails or requires new session
* Canvas resets when last user leaves
* Frequent data loss

**Testing Scenarios for Persistence:**

1. Mid-Operation Refresh: User drags object, refreshes browser mid-drag → object position preserved
2. Total Disconnect: All users close browsers, wait 2 minutes, return → full canvas state intact
3. Network Simulation: Throttle network to 0 for 30 seconds, restore → canvas syncs without data loss
4. Rapid Disconnect: User makes 5 rapid edits, immediately closes tab → edits persist for other users

## Section 2: Canvas Features & Performance (20 points)

### Canvas Functionality (8 points)

**Excellent (7-8 points)**

* Smooth pan/zoom
* 3+ shape types
* Text with formatting
* Multi-select (shift-click or drag)
* Layer management
* Transform operations (move/resize/rotate)
* Duplicate/delete

**Good (5-6 points)**

* All basic requirements met
* 2+ shapes
* Transforms work well
* Basic text support

**Satisfactory (3-4 points)**

* Basic shapes and movement
* Limited transform capabilities
* Single selection only

**Poor (0-2 points)**

* Missing core features
* Broken transforms
* Poor or no text support

### Performance & Scalability (12 points)

**Excellent (11-12 points)**

* Consistent performance with 500+ objects
* Supports 5+ concurrent users
* No degradation under load
* Smooth interactions at scale

**Good (9-10 points)**

* Consistent performance with 300+ objects
* Handles 4-5 users
* Minor slowdown under heavy load

**Satisfactory (6-8 points)**

* Consistent performance with 100+ objects
* 2-3 users supported
* Noticeable lag with complexity

**Poor (0-5 points)**

* Fails performance targets
* Drops below 60 FPS easily
* Can't handle multiple users

## Section 3: Advanced Figma-Inspired Features (15 points)

### Overall Scoring

**Excellent (13-15 points):** 3 Tier 1 + 2 Tier 2 + 1 Tier 3 features, all working excellently

**Good (10-12 points):** 2-3 Tier 1 + 1-2 Tier 2 features, all working well

**Satisfactory (6-9 points):** 2-3 Tier 1 features OR 1 Tier 2 feature working adequately

**Poor (0-5 points):** Only 0-1 additional features or features poorly implemented

### Feature Tiers

**Tier 1 Features (2 points each, max 3 features = 6 points)**

* Color picker with recent colors/saved palettes
* Undo/redo with keyboard shortcuts (Cmd+Z/Cmd+Shift+Z)
* Keyboard shortcuts for common operations (Delete, Duplicate, Arrow keys to move)
* Export canvas or objects as PNG/SVG
* Snap-to-grid or smart guides when moving objects
* Object grouping/ungrouping
* Copy/paste functionality

**Tier 2 Features (3 points each, max 2 features = 6 points)**

* Component system (create reusable components/symbols)
* Layers panel with drag-to-reorder and hierarchy
* Alignment tools (align left/right/center, distribute evenly)
* Z-index management (bring to front, send to back)
* Selection tools (lasso select, select all of type)
* Styles/design tokens (save and reuse colors, text styles)
* Canvas frames/artboards for organizing work

**Tier 3 Features (3 points each, max 1 feature = 3 points)**

* Auto-layout (flexbox-like automatic spacing and sizing)
* Collaborative comments/annotations on objects
* Version history with restore capability
* Plugins or extensions system
* Vector path editing (pen tool with bezier curves)
* Advanced blend modes and opacity
* Prototyping/interaction modes (clickable links between frames)

## Section 4: AI Canvas Agent (25 points)

### Command Breadth & Capability (10 points)

**Excellent (9-10 points)**

* 8+ distinct command types
* Covers all categories: creation, manipulation, layout, complex
* Commands are diverse and meaningful

**Good (7-8 points)**

* 6-7 command types
* Covers most categories
* Good variety

**Satisfactory (5-6 points)**

* Exactly 6 command types
* Limited variety
* Minimal category coverage

**Poor (0-4 points)**

* Fewer than 6 commands
* Commands don't work reliably
* Very limited scope

### AI Command Categories (must demonstrate variety):

**Creation Commands (at least 2 required)**

* "Create a red circle at position 100, 200"
* "Add a text layer that says 'Hello World'"
* "Make a 200x300 rectangle"

**Manipulation Commands (at least 2 required)**

* "Move the blue rectangle to the center"
* "Resize the circle to be twice as big"
* "Rotate the text 45 degrees"

**Layout Commands (at least 1 required)**

* "Arrange these shapes in a horizontal row"
* "Create a grid of 3x3 squares"
* "Space these elements evenly"

**Complex Commands (at least 1 required)**

* "Create a login form with username and password fields"
* "Build a navigation bar with 4 menu items"
* "Make a card layout with title, image, and description"

### Complex Command Execution (8 points)

**Excellent (7-8 points)**

* "Create login form" produces 3+ properly arranged elements
* Complex layouts execute multi-step plans correctly
* Smart positioning and styling
* Handles ambiguity well

**Good (5-6 points)**

* Complex commands work but simpler implementations
* Basic layouts created
* 2-3 elements arranged

**Satisfactory (3-4 points)**

* Basic interpretation of complex commands
* Poor layout quality
* Elements created but not arranged

**Poor (0-2 points)**

* Complex commands fail
* Nonsensical results
* Cannot handle multi-step operations

### AI Performance & Reliability (7 points)

**Excellent (6-7 points)**

* Sub-2 second responses
* 90%+ accuracy
* Natural UX with feedback
* Shared state works flawlessly
* Multiple users can use AI simultaneously

**Good (4-5 points)**

* 2-3 second responses
* 80%+ accuracy
* Good UX
* Shared state mostly works
* Minor conflicts with multi-user AI

**Satisfactory (2-3 points)**

* 3-5 second responses
* 60%+ accuracy
* Basic UX
* Shared state has issues

**Poor (0-1 points)**

* Slow responses (5s+)
* Unreliable execution
* Broken shared state
* Poor or no UX feedback

## Section 5: Technical Implementation (10 points)

### Architecture Quality (5 points)

**Excellent (5 points)**

* Clean, well-organized code
* Clear separation of concerns
* Scalable architecture
* Proper error handling
* Modular components

**Good (4 points)**

* Solid structure
* Minor organizational issues
* Generally maintainable

**Satisfactory (3 points)**

* Functional but messy
* Some architectural concerns
* Limited modularity

**Poor (0-2 points)**

* Poor code organization
* Architectural problems
* Difficult to maintain

### Authentication & Security (5 points)

**Excellent (5 points)**

* Robust auth system
* Secure user management
* Proper session handling
* Protected routes
* No exposed credentials

**Good (4 points)**

* Functional auth
* Minor security considerations
* Generally secure

**Satisfactory (3 points)**

* Basic auth works
* Some security gaps
* Needs improvement

**Poor (0-2 points)**

* Broken authentication
* Insecure implementation
* Major vulnerabilities

## Section 6: Documentation & Submission Quality (5 points)

### Repository & Setup (3 points)

**Excellent (3 points)**

* Clear README
* Detailed setup guide
* Architecture documentation
* Easy to run locally
* Dependencies listed

**Good (2 points)**

* Adequate documentation
* Setup mostly clear
* Can be run with some effort

**Satisfactory (1 point)**

* Minimal documentation
* Setup unclear
* Missing key info

**Poor (0 points)**

* Missing or inadequate documentation
* Cannot be set up

### Deployment (2 points)

**Excellent (2 points)**

* Stable deployment
* Publicly accessible
* Supports 5+ users
* Fast load times

**Good (1 point)**

* Deployed
* Minor stability issues
* Generally accessible

**Poor (0 points)**

* Broken deployment
* Not accessible
* Major issues

## Section 7: AI Development Log (Required - Pass/Fail)

PASS Requirements: Must include **ANY 3 out of 5 sections** with meaningful reflection:

1. **Tools & Workflow used** - What AI tools you used and how you integrated them
2. **3-5 effective prompting strategies** - Specific prompts that worked well
3. **Code analysis** - Rough percentage of AI-generated vs hand-written code
4. **Strengths & limitations** - Where AI excelled and where it struggled
5. **Key learnings** - Insights about working with AI coding agents

## 

## Section 8: Demo Video (Required - Pass/Fail)

**PASS Requirements**: 3-5 minute video demonstrating:

* Real-time collaboration with 2+ users (show both screens)
* Multiple AI commands executing
* Advanced features walkthrough
* Architecture explanation
* Clear audio and video quality

**FAIL Penalty**: Missing requirements OR poor quality OR not submitted = -10 points

## Bonus Points (Maximum +5)

**Innovation (+2 points)**

* Novel features beyond requirements
* Examples: AI-powered design suggestions, smart component detection, generative design tools

**Polish (+2 points)**

* Exceptional UX/UI
* Smooth animations
* Professional design system
* Delightful interactions

**Scale (+1 point)**

* Demonstrated performance beyond targets
* 1000+ objects at 60 FPS
* 10+ concurrent users

## Grade Scale

**A (90-100 points)**: Exceptional implementation, exceeds all targets, production-ready quality

**B (80-89 points)**: Strong implementation, meets all core requirements, good quality

**C (70-79 points)**: Functional implementation, meets most requirements, acceptable quality

**D (60-69 points)**: Basic implementation, significant gaps, needs improvement

**F (<60 points)**: Does not meet minimum requirements, major issues