

Comprehensive Project Analysis: Collaborative Whiteboard Application

1. Feature Analysis & Architecture

Core Features Identified

A. Real-Time Drawing System

- **Purpose:** Enable users to draw collaboratively with immediate visual feedback
- **User Flow:**
 - User selects drawing tool → Interacts with canvas → Stroke data broadcasts → All connected users see updates in real-time
- **Technical Implementation:**
 - Canvas API captures mouse/touch events
 - Socket.IO emits drawing coordinates/actions to server
 - Server broadcasts to all clients in room
 - Canvas re-renders strokes for all users
- **Core Value:** The heart of collaboration - synchronous creative work

B. Session Management (Room System)

- **Purpose:** Isolate collaborative spaces for different groups
- **User Flow:**
 - Create new room → Receive unique ID → Share with collaborators OR Join existing room → Enter room ID → Connect to session
- **Technical Implementation:**
 - Express.js generates unique room IDs
 - Socket.IO namespaces/rooms for user isolation
 - MongoDB stores room metadata
- **Dependencies:** Foundation for all other features

C. Drawing Toolset

- **Purpose:** Provide creative control and editing capabilities
- **Components:**
 - **Pen Tool:** Freehand drawing with variable stroke width
 - **Eraser:** Remove portions of drawing
 - **Color Picker:** Customize stroke colors
 - **Clear Canvas:** Reset entire workspace
- **User Flow:** Tool selection → Parameter adjustment → Apply to canvas
- **Technical Implementation:** State management for active tool, color, size; Canvas API composite operations for eraser

D. Snapshot Persistence

- **Purpose:** Save progress and enable session continuity
- **User Flow:**
 - Trigger snapshot save → Canvas converts to data URL → Stores in MongoDB → Retrieve via endpoint → Load previous state
- **Technical Implementation:**
 - Canvas `toDataURL()` method
 - MongoDB document with room ID, timestamp, image data

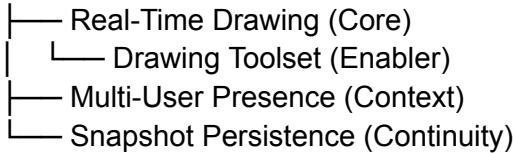
- REST endpoints for save/retrieve
- **Dependencies:** Requires active session

E. Multi-User Presence

- **Purpose:** Show who's actively collaborating
- **User Flow:** User joins → Presence indicator appears → User leaves → Indicator removes
- **Technical Implementation:** Socket.IO connection events, real-time user list broadcast

Feature Dependency Map

Session Management (Foundation)



Core Value Proposition

"Instant visual collaboration without friction" - enabling teams to ideate, sketch, and communicate visually in real-time from anywhere.

2. Design Strategy & Best Practices

Feature: Session Management UI

Layout Patterns:

- **Landing Page:** Hero-centered layout with two clear CTAs (Create/Join)
- **Grid System:** 12-column grid, 1440px max-width container
- **Spacing:** 8px base unit (multiples: 16, 24, 32, 48, 64)
- **Hierarchy:** Large hero text (48-64px) → Subtitle (18-24px) → Input fields (16px) → Helper text (14px)

Interaction Patterns:

- **Micro-interactions:**
 - Room ID copy button with animated checkmark confirmation
 - Smooth fade-in on page load
 - Button hover states with subtle scale (1.02) and shadow elevation
- **Transitions:** 200ms ease-in-out for most UI, 300ms for page transitions
- **States:** Default, hover, focus, active, loading, error

Accessibility:

- WCAG AA contrast ratios (4.5:1 text, 3:1 UI components)
- Focus indicators with 2px offset outline
- Screen reader labels for all interactive elements
- Keyboard navigation: Tab order, Enter to submit

Responsive Approach:

- Mobile-first design
- Breakpoints: 640px (sm), 768px (md), 1024px (lg), 1280px (xl)
- Stack vertically on mobile, side-by-side CTAs on desktop

Component Structure:

```
<LandingPage>
  <Hero>
    <Heading />
    <Subtitle />
  </Hero>
  <ActionPanel>
    <CreateRoomCard />
    <JoinRoomCard />
  </ActionPanel>
</LandingPage>
```

Feature: Drawing Canvas Interface

Layout Patterns:

- **Full-screen canvas:** 100vh minus toolbar height
- **Floating toolbar:** Fixed position, top or left edge
- **Contextual panels:** Slide-out color picker, settings
- **Z-index hierarchy:** Canvas (0) → Toolbar (10) → Modals (100)

Interaction Patterns:

- **Drawing feedback:** Cursor changes to reflect active tool (crosshair for pen, circle for eraser)
- **Smooth drawing:** Implement Catmull-Rom splines or Bézier curves for smoother lines
- **Touch support:** Prevent default touch behaviors, support multi-touch for future zoom/pan
- **Undo/Redo:** Keyboard shortcuts (Cmd/Ctrl + Z/Y) with visual feedback
- **Toolbar collapse:** Auto-hide on mobile, persistent on desktop

Accessibility:

- Alternative input methods beyond mouse (keyboard drawing with arrow keys)
- High contrast mode support
- Announce tool changes to screen readers
- Keyboard shortcuts overlay (?) key

Responsive Design:

- **Desktop:** Side toolbar with expanded tool options
- **Tablet:** Top toolbar, optimized for touch
- **Mobile:** Minimal toolbar with drawer for advanced options
- Canvas scales to fit viewport maintaining aspect ratio

Component Structure:

```
<WhiteboardWorkspace>
  <Toolbar position="left">
    <ToolSelector />
    <ColorPicker />
    <SizeSlider />
    <ActionButtons />
  </Toolbar>
  <Canvas ref={canvasRef} />
  <UserPresence />
  <SnapshotControls />
```

Feature: Drawing Toolset

Layout Patterns:

- **Vertical icon stack:** 56px width, icons 24x24px
- **Tool grouping:** Visual separators between tool types
- **Active state:** Highlighted background, left border accent

Interaction Patterns:

- **Tool selection:** Single-click activation, deselect previous
- **Parameter adjustment:** Inline sliders/pickers that appear on tool activation
- **Quick access:** Number keys 1-5 for tool shortcuts
- **Visual feedback:** Tool preview on hover (show color/size preview)

Color Picker Patterns:

- **Preset swatches:** 8-12 common colors in grid
- **Custom picker:** Full spectrum picker with recent colors
- **Eyedropper tool:** (Future enhancement) pick colors from canvas

Accessibility:

- Keyboard navigation through tools (arrow keys)
 - Tooltips on hover with keyboard shortcut hints
 - High contrast icons
 - Focus states distinct from hover states
-

Feature: Multi-User Presence

Layout Patterns:

- **Floating avatar row:** Top-right corner, overlapping avatars
- **Compact:** 32px circles, overlap by 8px
- **Expandable:** Hover to see full user list

Interaction Patterns:

- **Join/Leave animations:** Fade + slide in from right
- **Active indicator:** Pulsing dot or ring around avatar
- **Cursor tracking:** Show other users' cursor positions with name label
- **Color coding:** Each user assigned unique color for their cursor/strokes

Accessibility:

- Screen reader announces user joins/leaves
 - User count indicator for quick reference
 - Keyboard accessible user list panel
-

Feature: Snapshot System

Layout Patterns:

- **Floating action button:** Bottom-right corner
- **Modal gallery:** Grid of thumbnail previews
- **Timeline view:** Chronological list with timestamps

Interaction Patterns:

- **Save feedback:** Success toast notification with undo option
- **Auto-save:** Periodic background saves with visual indicator
- **Preview on hover:** Zoom thumbnail in gallery
- **Load confirmation:** Prevent accidental overwrites

Accessibility:

- Keyboard shortcut for quick save (Cmd/Ctrl + S)
 - Clear labels for save/load actions
 - Confirmation dialogs with keyboard navigation
-

3. UI/UX Inspiration & References

Session Management & Onboarding

1. **Zoom's Meeting Join Flow** - Simple room code entry with instant validation
2. **Figma's File Creation** - Clean, minimal landing with strong CTAs
3. **Miro's Board Creation** - Beautiful illustrations + clear value proposition
4. **Linear's Onboarding** - Smooth transitions, progressive disclosure
5. **Vercel's Dashboard** - Modern card-based layout for actions

Design References:

- Dribbble: Search "collaborative workspace onboarding"
 - Awwwards: <https://www.awwwards.com/websites/collaboration/>
-

Whiteboard Canvas Interface

1. **Excalidraw** (<https://excalidraw.com>) - Clean, distraction-free canvas with minimal toolbar
2. **Miro** - Comprehensive toolset, excellent touch optimization
3. **FigJam** - Playful interactions, smooth drawing experience
4. **tldraw** (<https://tldraw.com>) - Open-source, beautiful minimal design
5. **Google Jamboard** - Simple, accessible interface

Specific Inspirations:

- **Toolbar Design:** Excalidraw's floating, semi-transparent toolbar
- **Color Picker:** Figma's organized swatch + custom picker combo
- **Canvas Feel:** tldraw's infinite canvas with subtle grid

Design References:

- Dribbble: "whiteboard app UI"
 - Behance: "collaborative canvas interface"
-

Drawing Tools & Controls

1. **Procreate** - Excellent brush/tool selection UX (iPad)
 2. **Adobe Fresco** - Intuitive color selection
 3. **Concepts App** - Infinite canvas controls
 4. **Paper by WeTransfer** - Minimal, gesture-driven tools
 5. **Notability** - Clean eraser implementation
-

User Presence & Cursors

1. **Figma's Multiplayer Cursors** - Industry standard for real-time collaboration
 2. **Google Docs** - Named cursor labels with user colors
 3. **Notion** - Subtle presence indicators
 4. **Linear** - Elegant avatar stacks
 5. **GitHub** - Real-time collaboration indicators
-

Overall Design System References

1. **Radix UI Design System** - Excellent component patterns
 2. **Chakra UI** - Accessibility-first approach
 3. **shadcn/ui** - Modern, customizable components
 4. **Material Design 3** - Comprehensive guidelines
 5. **Apple Human Interface Guidelines** - Touch interaction patterns
-

4. Visual Design System

Color Palette Option 1: "Modern Minimalist"

Rationale: Clean, professional, high contrast - ideal for focused work environments

Light Mode:

- **Primary:** #2563EB (Blue 600) - Trust, professionalism
- **Secondary:** #8B5CF6 (Purple 500) - Creativity, innovation
- **Accent:** #F59E0B (Amber 500) - Attention, energy
- **Background:** #FFFFFF
- **Surface:** #F9FAFB (Gray 50)
- **Border:** #E5E7EB (Gray 200)
- **Text Primary:** #111827 (Gray 900)
- **Text Secondary:** #6B7280 (Gray 500)

Semantic:

- **Success:** #10B981 (Green 500)
- **Warning:** #F59E0B (Amber 500)
- **Error:** #EF4444 (Red 500)
- **Info:** #3B82F6 (Blue 500)

Dark Mode:

- **Background:** #0F172A (Slate 900)

- **Surface:** #1E293B (Slate 800)
 - **Border:** #334155 (Slate 700)
 - **Text Primary:** #F1F5F9 (Slate 100)
 - **Text Secondary:** #94A3B8 (Slate 400)
-

Color Palette Option 2: "Vibrant Creative"

Rationale: Energetic, playful, inspiring - appeals to creative teams and brainstorming sessions

Light Mode:

- **Primary:** #EC4899 (Pink 500) - Energy, creativity
- **Secondary:** #8B5CF6 (Purple 500) - Imagination
- **Accent:** #14B8A6 (Teal 500) - Balance, growth
- **Background:** #FFFFFF
- **Surface:** #FDF4FF (Fuchsia 50)
- **Border:** #F5D0FE (Fuchsia 200)
- **Text Primary:** #1F2937 (Gray 800)
- **Text Secondary:** #6B7280 (Gray 500)

Semantic:

- **Success:** #22C55E (Green 500)
- **Warning:** #F97316 (Orange 500)
- **Error:** #F43F5E (Rose 500)
- **Info:** #06B6D4 (Cyan 500)

Dark Mode:

- **Background:** #18181B (Zinc 900)
 - **Surface:** #27272A (Zinc 800)
 - **Border:** #3F3F46 (Zinc 700)
-

Color Palette Option 3: "Soft Professional"

Rationale: Warm, approachable, reduces eye strain - perfect for extended collaboration sessions

Light Mode:

- **Primary:** #0EA5E9 (Sky 500) - Openness, clarity
- **Secondary:** #64748B (Slate 500) - Neutrality, balance
- **Accent:** #FB923C (Orange 400) - Warmth, creativity
- **Background:** #FEFCE8 (Yellow 50) - Warm, comfortable
- **Surface:** #FFFFFF
- **Border:** #E2E8F0 (Slate 200)
- **Text Primary:** #334155 (Slate 700)
- **Text Secondary:** #64748B (Slate 500)

Semantic:

- **Success:** #84CC16 (Lime 500)
- **Warning:** #FACC15 (Yellow 400)
- **Error:** #F87171 (Red 400)
- **Info:** #38BDF8 (Sky 400)

Typography System

Recommended Font Pairing 1: "Modern & Readable"

- **Headings:** Inter (Sans-serif) - Clean, highly legible
- **Body:** Inter - Consistency across interface
- **Monospace:** JetBrains Mono - For room codes, technical info

Scale (Tailwind-inspired):

- **xs:** 12px / 16px line-height
- **sm:** 14px / 20px
- **base:** 16px / 24px
- **lg:** 18px / 28px
- **xl:** 20px / 28px
- **2xl:** 24px / 32px
- **3xl:** 30px / 36px
- **4xl:** 36px / 40px
- **5xl:** 48px / 1 (tight)

Font Pairing 2: "Elegant & Professional"

- **Headings:** Sora - Geometric, modern
- **Body:** DM Sans - Warm, readable
- **Monospace:** Source Code Pro

Font Pairing 3: "Playful & Friendly"

- **Headings:** Outfit - Rounded, approachable
- **Body:** Plus Jakarta Sans - Friendly, legible
- **Monospace:** Fira Code

Spacing System (8px base unit)

- **0:** 0px
- **1:** 4px
- **2:** 8px
- **3:** 12px
- **4:** 16px
- **5:** 20px
- **6:** 24px
- **8:** 32px
- **10:** 40px
- **12:** 48px
- **16:** 64px
- **20:** 80px
- **24:** 96px

Sizing System:

- **xs:** 20px
- **sm:** 24px
- **md:** 28px
- **lg:** 32px

- **xl:** 36px
 - **2xl:** 42px
 - **3xl:** 48px
-

5. Creative Ideation

Innovative Feature Enhancements

A. Smart Gesture Recognition

- **Concept:** AI recognizes hand-drawn shapes (circles, squares, arrows) and auto-corrects them
- **User Delight:** Draw sloppy diagram → Instantly perfected
- **Tech:** TensorFlow.js for client-side shape recognition
- **Inspiration:** Microsoft PowerPoint's ink-to-shape feature

B. Voice-to-Sticky-Note

- **Concept:** Speak ideas, automatically creates sticky notes on canvas
- **Use Case:** Rapid brainstorming without typing
- **Tech:** Web Speech API
- **UX Pattern:** Floating microphone button, visual waveform during recording

C. Template Library

- **Concept:** Pre-designed templates (brainstorm grids, flowcharts, wireframes)
- **User Value:** Faster setup, professional structure
- **Implementation:** MongoDB stores template data, one-click apply
- **Monetization Opportunity:** Free basic, premium advanced templates

D. Time-Travel Playback

- **Concept:** Replay entire drawing session as timelapse
- **Use Cases:** Presentations, documentation, teaching
- **Tech:** Store timestamped stroke arrays, playback with speed control
- **Delight Factor:** "Watch your ideas come to life"

E. Collision Detection & Awareness

- **Concept:** Visual indicators when multiple users draw in same area
- **Prevention:** Reduce accidental overlap conflicts
- **Implementation:** Real-time coordinate tracking, proximity alerts
- **UX:** Subtle pulse/glow around other users' active zones

F. Infinite Canvas with Minimap

- **Concept:** Pan/zoom infinite workspace with navigation minimap
- **User Value:** Organize large, complex ideas
- **Tech:** Canvas translation/scaling, minimap in corner
- **Inspiration:** Miro, Figma's canvas navigation

G. Laser Pointer Mode

- **Concept:** Temporary cursor trail for presentations (doesn't persist)
- **Use Case:** Guide team attention during real-time discussions
- **Implementation:** Temporary drawing layer that auto-fades
- **UX:** Toggle mode with distinct cursor style

H. Emoji Reactions & Annotations

- **Concept:** Drop emoji reactions on canvas areas
- **Social Layer:** Quick feedback without interrupting flow
- **Implementation:** Draggable emoji with fade-out animation
- **Inspiration:** Slack's emoji reactions, Zoom's nonverbal feedback

I. Export Options

- **Formats:** PNG, SVG, PDF, animated GIF (for time-travel)
- **Branding:** Optional watermark for free tier
- **Quality:** High-res option for presentations

J. Collaborative Cursors with Labels

- **Enhancement:** Show cursor position + user name + current tool
 - **Tech:** Socket.IO broadcasts cursor coordinates, React renders
 - **Customization:** Users choose cursor color from palette
-

Unique UX Patterns

Haptic Feedback (Mobile)

- **Concept:** Subtle vibration on tool select, canvas clear
- **Platform:** Native mobile wrapper or PWA with vibration API
- **Enhancement:** Increases tactile engagement

Contextual Radial Menu

- **Trigger:** Right-click or long-press on canvas
- **Content:** Quick access to recently used tools/colors
- **Inspiration:** Blender's radial menus
- **Benefit:** Reduces toolbar trips

"Focus Mode"

- **Concept:** Dim other users' strokes to reduce visual noise
- **Toggle:** Keyboard shortcut or toolbar button
- **Use Case:** Individual work within collaborative session

Progressive Disclosure Toolbar

- **Concept:** Starts minimal (pen + color), expands on demand
 - **Benefit:** Reduces cognitive load for new users
 - **Implementation:** Expandable sections, "Show more tools"
-

Emerging Design Trends to Consider

Glassmorphism for Toolbars

- **Style:** Frosted glass effect, subtle backdrop blur
- **CSS:** `backdrop-filter: blur(10px)`, semi-transparent backgrounds
- **Advantage:** Modern, doesn't obscure canvas completely
- **Browser Support:** Check caniuse.com for backdrop-filter

Neumorphism for Tool Buttons

- **Style:** Soft shadows creating "extruded" button effect
- **Application:** Active tool state
- **Caution:** Can reduce accessibility if contrast is poor

Micro-animations with Framer Motion

- **Examples:**
 - Spring physics for modal entry
 - Gesture-based swipe to dismiss
 - Stagger animations for user list
- **Library:** Framer Motion for React
- **Performance:** Use `transform` and `opacity` for 60fps

3D Elements (Subtle)

- **Concept:** Slight perspective tilt on hover for cards
- **Implementation:** CSS 3D transforms
- **Example:** Landing page hero cards with depth

Dark Mode with Auto-Detection

- **Implementation:** Respect `prefers-color-scheme` media query
 - **Toggle:** User override option
 - **Persistence:** LocalStorage preference
-

6. Implementation Roadmap

Phase 1: Foundation

Core Infrastructure:

- Set up React app with Vite (faster than CRA)
- Express server with Socket.IO integration
- MongoDB connection with Mongoose schemas
- Basic routing (React Router)

Components to Build:

```
/components
/layout
- Header.jsx
- Footer.jsx
/pages
- LandingPage.jsx
- WhiteboardPage.jsx
/ui
- Button.jsx
- Input.jsx
- Card.jsx
```

Design System Setup:

- Implement chosen color palette as CSS variables

- Configure Tailwind with custom theme
- Set up typography scale
- Create base component variants

Technical Considerations:

- Decide on state management (Context API vs Zustand vs Redux)
 - Set up error boundaries
 - Configure CORS for Socket.IO
 - Environment variables for MongoDB URI
-

Phase 2: Core Canvas & Drawing

Canvas Implementation:

- Canvas context setup (2D)
- Mouse/touch event listeners
- Basic stroke rendering
- Undo/redo state management (consider Immer for immutability)

Components:

```
/components
  /whiteboard
    - Canvas.jsx (main canvas component)
    - Toolbar.jsx
    - ToolButton.jsx
    - ColorPicker.jsx
    - SizeSlider.jsx
```

Drawing Logic:

- Implement pen tool with variable stroke width
- Eraser with composite operation
- Clear canvas functionality
- Smooth line rendering (consider Bézier curves)

Technical Considerations:

- Canvas performance optimization (requestAnimationFrame)
- Debounce/throttle for socket emissions
- Handle high-frequency mouse events efficiently
- Mobile touch event handling (prevent default scroll)

Recommended Libraries:

- **react-canvas-draw** (baseline, may need customization)
 - **perfect-freehand** (library for smooth strokes)
 - **react-colorful** (accessible color picker)
-

Phase 3: Real-Time Collaboration

Socket.IO Integration:

- Room join/leave logic
- Drawing data broadcast
- User presence events
- Connection error handling

Components:

```
/components
/collaboration
- UserPresence.jsx
- CursorDisplay.jsx
- ConnectionState.jsx
```

Real-Time Features:

- Broadcast stroke data
- Receive and render remote strokes
- User list with avatars
- Cursor position tracking

Technical Considerations:

- Socket event optimization (bundle multiple events)
 - Handle network latency gracefully
 - Reconnection logic with state recovery
 - Rate limiting on server to prevent abuse
-

Phase 4: Session Management

Room System:

- Generate unique room IDs (nanoid library)
- Room creation endpoint
- Join room validation
- Share link generation

Components:

```
/components
/session
- CreateRoomForm.jsx
- JoinRoomForm.jsx
- RoomCodeDisplay.jsx
- ShareModal.jsx
```

User Flow:

- Landing page with create/join options
- Room code copy-to-clipboard
- Direct URL joining (whiteboard.app/room/abc123)
- Room expiration logic (optional)

Technical Considerations:

- URL routing with room IDs

- Clipboard API for share functionality
 - QR code generation for easy mobile join (optional)
 - Room capacity limits
-

Phase 5: Persistence

Snapshot System:

- Canvas to base64 conversion
- Save endpoint (POST /api/snapshots)
- Retrieve endpoint (GET /api/snapshots/:roomId)
- Load snapshot to canvas

MongoDB Schema:

```
{  
  roomId: String,  
  imageData: String, // base64  
  timestamp: Date,  
  metadata: {  
    userCount: Number,  
    strokeCount: Number  
  }  
}
```

Components:

```
/components  
  /snapshots  
    - SaveButton.jsx  
    - SnapshotGallery.jsx  
    - SnapshotThumbnail.jsx
```

Technical Considerations:

- Image compression before storage
 - Thumbnail generation for gallery
 - Pagination for snapshot history
 - Auto-save intervals (debounced)
-

Phase 6: Polish & Enhancement

Micro-interactions:

- Button hover/active states
- Tool selection feedback
- Save success animations
- Loading skeletons

Accessibility Audit:

- ARIA labels
- Keyboard navigation testing

- Screen reader testing
- Color contrast validation

Performance Optimization:

- Code splitting (React.lazy)
- Canvas rendering optimization
- Socket event batching
- Image lazy loading

Testing:

- Unit tests for utility functions
 - Integration tests for drawing logic
 - E2E tests with Playwright/Cypress
 - Multi-user testing scenarios
-

Component Library Recommendations

Option 1: shadcn/ui (Recommended)

- **Why:** Tailwind-based, copy-paste components, full control
- **Components:** Button, Input, Dialog, Slider, Tooltip
- **Customization:** Easily matches your design system
- **Accessibility:** Built-in ARIA support

Option 2: Radix UI Primitives

- **Why:** Unstyled, accessible primitives
- **Use Case:** Maximum design flexibility
- **Learning Curve:** Steeper, requires more styling work

Option 3: Chakra UI

- **Why:** Comprehensive, accessible, theme-based
- **Components:** Full suite out-of-the-box
- **Trade-off:** Less design freedom, heavier bundle

Option 4: Headless UI + Tailwind

- **Why:** Minimal, pairs perfectly with Tailwind
 - **Best For:** If using Tailwind CSS already
-

Technical Stack Recommendations

Frontend:

- **Build Tool:** Vite (fast, modern)
- **State Management:** Zustand (lightweight) or Jotai (atomic)
- **Forms:** React Hook Form (performance)
- **Animations:** Framer Motion
- **Icons:** Lucide React (clean, consistent)

Backend:

- **Server:** Express.js
- **Real-time:** Socket.IO
- **Database ODM:** Mongoose
- **Validation:** Zod or Joi
- **Environment:** dotenv

DevOps:

- **Hosting:** Vercel (frontend) + Railway/Render (backend)
 - **Database:** MongoDB Atlas
 - **CI/CD:** GitHub Actions
 - **Monitoring:** Sentry (error tracking)
-

Summary & Next Steps

This collaborative whiteboard has potential to be a standout project with:

1. **Core Strength:** Real-time collaboration with smooth drawing UX
2. **Differentiation:** Consider adding gesture recognition, time-travel playback, or templates
3. **Design Direction:** Choose between:
 - **Modern Minimalist** (professional, high contrast)
 - **Vibrant Creative** (energetic, playful)
 - **Soft Professional** (warm, extended-use friendly)
4. **Recommended First Steps:**
 - Choose color palette (I'd lean toward Modern Minimalist for broad appeal)
 - Set up design system with shadcn/ui + Tailwind
 - Build Phase 1-2 foundation with focus on drawing feel
 - Iterate on real-time performance before adding features
5. **Key Success Metrics:**
 - Drawing latency <50ms
 - Smooth 60fps canvas performance
 - Support 10+ concurrent users per room
 - Snapshot save/load <2 seconds