Context Collapse - Product Requirements Document

Executive Summary

Product Name: Context Collapse
Tagline: Your second brain that actually connects the dots
Category: Knowledge Management / Personal Productivity
Platform: Web (React), deployable on Vercel
Target Users: Product Managers, Knowledge Workers, Researchers, Creative Professionals

Vision

Create an AI-powered knowledge graph navigator that discovers surprising connections between ideas, generates serendipitous insights, and helps users build a dynamic, interconnected second brain.

Table of Contents

Core Features

1. Knowledge Graph Visualization

1.1 Force-Directed Graph Display

- **Description:** Interactive, physics-based graph visualization where nodes represent concepts and edges represent connections
- **Physics Parameters:**
  - Alpha (simulation strength): 0.3
  - Repulsion force: 5000
  - Attraction multiplier: 0.01
  - Damping factor: 0.8
- **Visual Properties:**
  - Node radius: 10px (default), 12px (hovered)
  - Node colors:
    - Default: Indigo (`#6366f1` light, `#4f46e5` dark)
    - Selected: Purple (`#8b5cf6` light, `#7c3aed` dark)
    - Search match: Blue (`#3b82f6` light, `#2563eb` dark)
  - Edge styles:

        - Regular: Gray solid line (`#4b5563` dark, `#d1d5db` light)
        - Surprising: Yellow dashed line (`#fbbf24` dark, `#f59e0b` light)
    - Edge opacity: Based on connection strength (0.1-1.0)

1.2 Canvas Specifications

- **Container:**
    - Mobile padding: 16px (p-4)
    - Desktop padding: 32px (p-8)
    - Border: 2px rounded (rounded-xl)
    - Border colors: `#374151` (dark), `#e5e7eb` (light)
    - Background: Semi-transparent (`#1f2937`/50 dark, `#ffffff`/50 light)
    - Shadow: Large shadow (shadow-lg)
- **Canvas Sizing:**
    - Dynamically sized to fit container
    - Updates on window resize
    - Maintains aspect ratio
    - Clearly visible borders on all sides

1.3 Graph Interactions

- **Pan:** Click and drag canvas background
- **Zoom:**
    - Requires Cmd (Mac) or Ctrl (Windows/Linux) + Scroll
    - Range: 0.5x to 3.0x
    - Delta per scroll: 0.9 (zoom out) or 1.1 (zoom in)
- **Node Selection:**
    - Click node to select
    - Shows floating popup near node
    - Shows detail panel below graph
    - Selected node highlighted with border
- **Node Dragging:**
    - Click and drag individual nodes
    - Node position updates in real-time
    - Physics disabled for dragged node
    - Popup follows node if open
- **Search Highlighting:**
    - Real-time search as user types
    - Matching nodes highlighted in blue
    - Case-insensitive matching
- **Connection Filtering:**
    - Slider control (0.0 to 1.0)
    - Hides connections below threshold
    - Updates connection count in real-time

1.4 Graph Controls Panel

Location: Top-left of canvas
Style: Floating card with shadow
Contents:

- Zoom level (e.g., "Zoom: 0.5x")
- Node count (e.g., "Nodes: 14")
- Visible connection count (e.g., "Connections: 13")

- Filter slider with funnel icon
- All items have tooltips explaining functionality

1.5 Zoom Hint

Location: Bottom-center of canvas
Style: Pill-shaped, semi-transparent background
Content:

- Desktop: "Hold [⌘/Ctrl] + Scroll to zoom"
- Mobile: "[⌘/Ctrl]+Scroll to zoom"
- Platform-aware keyboard shortcut display

---

2. Content Ingestion System

2.1 Input Methods

File Upload

- **Supported Formats:**
    - Plain text (.txt)
    - Markdown (.md)
    - PDF (.pdf)
- **Process:**
    1. User selects file via file input
    2. File read using FileReader API
    3. Text extraction (native for txt/md, basic text extraction for PDF)
    4. Content sent to AI for concept extraction
- **Limitations:**
    - No OCR for handwritten notes in v1
    - PDF extraction is basic (no complex layout parsing)
    - File size limit: TBD based on API constraints

URL Import

- **Supported Sources:**
    - Public Google Docs (auto-converts to export URL)
    - Public websites
    - Blog posts, articles
    - Any publicly accessible web page
- **Process:**
    1. User pastes URL
    2. If Google Docs URL detected, convert to export format
    3. Fetch content via fetch API
    4. Extract text content
    5. Send to AI for processing
- **Error Handling:**
    - Alert user if URL is not publicly accessible
    - Suggest copy-paste alternative for private content

Direct Text Paste

- **Interface:** Multi-line textarea (6 rows)
- **Process:**

1. User pastes or types content
2. Click "Add Note" button
3. Content sent directly to AI
4. Textarea clears on successful addition

2.2 AI Concept Extraction

Prompt Template:

```
Analyze this content and extract 3-7 key concepts or themes.
Return ONLY a JSON array of strings, nothing else.

Content: {first 3000 characters of content}

Return format: ["concept1", "concept2", "concept3"]
```

Response Processing:

- Strip markdown code blocks (`json, `)
- Parse JSON array
- Create node for each concept
- Assign unique ID: `${timestamp}-${index}`
- Set random initial positions within bounds
- Store full original content with each node

Error Handling:

- Catch JSON parse errors
- Alert user with clear error message
- Log error to console for debugging
- Don't add partial data to graph

2.3 Connection Discovery

Two-Phase Connection Process:

Phase 1: External Connections
(Between new concepts and existing graph)

Prompt Template:

```
Given these existing concepts: {comma-separated existing node labels}

And these new concepts: {comma-separated new node labels}

Find surprising and non-obvious connections between them.
Focus on unexpected relationships, not obvious ones.

Return ONLY a JSON array of connections in this exact format:
[
  {
    "existing": "existing concept name",
    "new": "new concept name",
    "strength": 0.1-1.0,
    "reason": "brief explanation",
    "surprising": true/false
  }
]

Return at least 5 connections, prioritizing surprising ones.
Return ONLY valid JSON, no other text.
```

Phase 2: Internal Connections
(Between new concepts from same source)

Prompt Template:

```
Given these concepts from the same content: {comma-separated new concepts}

Find connections between these concepts. They are from the same source,
so look for how they relate to each other.

Return ONLY a JSON array of connections in this exact format:
[
  {
    "concept1": "first concept name",
    "concept2": "second concept name",
    "strength": 0.1-1.0,
    "reason": "brief explanation",
    "surprising": true/false
  }
]

Return at least 3 connections. Return ONLY valid JSON, no other text.
```

Connection Matching:

- Use case-insensitive partial string matching
- Find nodes where label includes the concept name
- Skip if no matching nodes found
- Store connection with source/target IDs

---

3. Serendipity Engine

3.1 Trigger Mechanisms

- **Manual:** Lightning bolt button in header
- **Requirements:** Minimum 3 nodes in graph
- **Frequency:** User-initiated, no auto-suggestions in v1

3.2 Idea Generation

Prompt Template:

```
Given these concepts in a knowledge graph: {comma-separated all node labels}

Generate 5 creative ideas by combining unexpected concepts. Focus on:
- Novel intersections that have not been explored
- Surprising combinations
- Actionable project ideas
- Creative synthesis

Return ONLY a JSON array of strings, each being one creative idea:
["idea 1", "idea 2", "idea 3", "idea 4", "idea 5"]
```

Response Processing:

- Parse JSON array of 5 ideas
- Display in modal dialog
- Allow regeneration for new ideas
- No persistence in v1 (regenerate each time)

3.3 UI Display

Modal Properties:

- Max width: 2xl (672px)
- Max height: 80vh with scroll
- Header: Title with sparkle icon
- Each idea displayed as numbered card
- Hover effect: border color changes to indigo
- "Generate More Ideas" button at bottom

---

4. Node Selection & Detail Views

4.1 Floating Popup (Near Node)

Trigger: Click on any node
Position:

- Appears 20px to the right of click position
- Adjusts to stay within canvas bounds
- Maximum width: 384px (max-w-sm)
- Follows node if dragged

Content:

- **Header:**
  - Node label (bold, small text)
  - Close button (X icon)
- **Connections Preview:**
  - "Connections" label (gray, xs text)
  - Top 3 connections only

    - Each showing:
        - Connected node name
        - Sparkle icon if surprising
        - Truncated reason text
- **Scroll:** Max height 128px with overflow

Styling:

- Card background with border
- Shadow-xl for elevation
- Rounded corners (rounded-lg)
- Z-index: 10

4.2 Detail Panel (Below Graph)

Trigger: Same as popup (node selection)
Location: Below canvas, above any other content
Maximum Height: 320px with scroll

Layout:

- Full width panel
- Top border separator
- Padding: 24px (p-6)

Content Sections:

1. **Header:**
    - Node label (text-xl, font-bold)
    - 16px margin below
2. **Metadata Grid (2 columns):**
    - Type: Capitalize (note/url/file)
    - Added: Localized date
3. **Content Preview:**
    - Label: "Content Preview"
    - First 500 characters + "..."
    - Scrollable container (max-h-32)
    - Gray background for contrast
4. **All Connections:**
    - Label: "All Connections"
    - Each connection in bordered card:
        - Connected node name + sparkle if surprising
        - Connection reason (xs, gray text)
        - Strength percentage (xs, lighter gray)
5. **Delete Button:**
    - Full width
    - Red background (`#ef4444`)
    - Trash icon + "Delete Node" text
    - Deletes node and all connected edges
    - Closes both popup and panel
    - No confirmation dialog (uses confirm())

---

5. Theme System

5.1 Theme Options

- **Light Mode:** Bright, high-contrast
- **Dark Mode:** Dark grays, reduced eye strain
- **System Mode:** Follows OS preference via media query

5.2 Theme Toggle

Location: Settings modal
Interface: Three buttons with icons

- Sun icon: Light
- Moon icon: Dark
- Monitor icon: System

Active State:

- Indigo border
- Indigo background tint
- Visual feedback on selection

5.3 Color Specifications

Light Theme:

- Background: `#f9fafb`
- Text: `#111827`
- Card: `#ffffff`
- Border: `#e5e7eb`
- Input: `#ffffff` / `#1f2937` (border)

Dark Theme:

- Background: `#111827`
- Text: `#f3f4f6`
- Card: `#1f2937`
- Border: `#374151`
- Input: `#374151` / `#4b5563` (border)

Theme Persistence:

- Store in localStorage: theme preference
- Apply on mount
- Listen for system preference changes
- Smooth transitions between themes

---

6. API Configuration

6.1 Default Mode

- Uses environment variable API key

- Hidden from user by default
- No configuration needed
- Works out of the box

6.2 Custom API Mode

Toggle: Checkbox in settings ("Use custom API")
Hidden by default to avoid overwhelming UI

When Enabled:

1. **Provider Selection:**
   - Claude (Anthropic)
   - OpenAI
   - OpenAI-compatible (custom)
2. **API Key Input:**
   - Password field
   - Placeholder: "sk-..."
   - Stored in localStorage (warn about security)
3. **Base URL (for custom only):**
   - URL input
   - Placeholder: "https://api.example.com/v1/chat/completions"
   - Optional for Claude/OpenAI

Save Button:

- Persists to localStorage
- Shows success alert
- Config used for all subsequent API calls

6.3 API Call Logic

Provider: Claude

javascript

```
POST https://api.anthropic.com/v1/messages
Headers:
  Content-Type: application/json
  x-api-key: {api_key}
  anthropic-version: 2023-06-01
Body:
  {
    model: "claude-sonnet-4-20250514",
    max_tokens: 4000,
    messages: [{ role: "user", content: {prompt} }]
  }
Response: data.content[0].text
```

Provider: OpenAI / Custom

javascript

```
POST {base_url or https://api.openai.com/v1/chat/completions}
Headers:
  Content-Type: application/json
  Authorization: Bearer {api_key}
Body:
  {
    model: "gpt-4",
    messages: [{ role: "user", content: {prompt} }],
    max_tokens: 4000
  }
Response: data.choices[0].message.content
```

Error Handling:

- Check response.ok
- Throw descriptive errors
- Alert user with actionable message
- Log full error to console

---

7. Data Persistence

7.1 LocalStorage Schema

Graph Data:

javascript

```javascript
Key: "context-collapse-graph"
Value: JSON.stringify({
  nodes: [
    {
      id: string,
      label: string,
      content: string,
      type: "note" | "url" | "file",
      timestamp: number,
      x: number,
      y: number,
      vx: number,
      vy: number
    }
  ],
  connections: [
    {
      source: string (node id),
      target: string (node id),
      strength: number (0.1-1.0),
      reason: string,
      isSurprising: boolean
    }
  ]
})
```

API Config:

javascript

```javascript
Key: "context-collapse-api"
Value: JSON.stringify({
  provider: "claude" | "openai" | "custom",
  apiKey: string,
  baseUrl?: string
})
```

Welcome State:

javascript

```
Key: "context-collapse-welcome"
Value: "true" (string, set after first visit)
```

7.2 Save/Load Logic

- **Auto-save:** Graph updates trigger localStorage write
- **Auto-load:** On component mount, read from localStorage
- **Error handling:** Try-catch with console errors, don't break app

7.3 Export/Import

Export:

- Button in settings
- Generates JSON file with pretty printing
- Filename: `context-collapse-{timestamp}.json`
- Downloads via Blob URL

Import:

- File input (hidden, triggered by button)
- Accepts only .json files
- Parses and validates JSON
- Replaces current graph entirely
- Alert on parse errors

Clear All:

- Red button in settings
- Browser confirm() dialog
- Removes graph from localStorage
- Resets graph state to empty
- Cannot be undone

---

8. Welcome Experience

8.1 Trigger

- Shows on first visit (no "context-collapse-welcome" in localStorage)
- Can be reopened via eye icon in header
- Overlay modal blocks interaction

8.2 Modal Content

Header:

- Sparkle icon + "Welcome to Context Collapse"
- Subtitle: "Your second brain that actually connects the dots"

Feature Explanations (4 sections):

1. **Add Your Knowledge** (Plus icon, Indigo background)
   - Upload files, paste URLs, or add notes
   - AI extracts key concepts automatically
2. **Discover Surprising Connections** (Sparkle icon, Yellow background)
   - AI finds non-obvious relationships
   - Shown as dashed yellow lines
3. **Generate Serendipitous Ideas** (Zap icon, Purple background)
   - Click lightning bolt for creative ideas
   - Unexpected concept combinations
4. **Explore Your Graph** (Eye icon, Green background)
   - Click nodes to see details
   - Drag to rearrange
   - Hold Cmd/Ctrl + Scroll to zoom
   - Use filter to hide weak connections

Tip Box:

- Blue background with border
- "Tip: Add diverse content from different domains..."

Action Buttons:

1. **Get Started** (Primary, Indigo)
   - Closes welcome
   - Sets localStorage flag
   - Opens upload modal
2. **Skip Tutorial** (Secondary, Gray text)
   - Closes welcome
   - Sets localStorage flag

---

9. Header Navigation

9.1 Layout

Container: Sticky top, z-index 50, border-bottom
Max width: 7xl (1280px), centered
Flex layout: Space-between on desktop, wrap on mobile

9.2 Left Section

- **Logo:** Sparkle icon (indigo)
- **Title:** "Context Collapse" (hidden on mobile <640px)

9.3 Center Section

Search Bar:

- Flex-1, max-width 448px (max-w-md)
- Search icon (left-aligned inside input)

- Placeholder: "Search concepts..."
- Rounded, border, focus ring
- Real-time filtering
- Tooltip: "Search for concepts in your knowledge graph..."

9.4 Right Section (Icon Buttons)

All buttons:

- Rounded (rounded-lg)
- Hover background change
- Padding: 8px (p-2)
- Transitions

Buttons (left to right):

1. **Plus:** Add content (opens upload modal)
2. **Zap:** Generate serendipity (runs AI generation)
3. **Eye:** Help (opens welcome modal)
4. **Settings:** Settings (opens settings modal)

All have descriptive tooltips

---

10. Upload Modal

10.1 Modal Structure

- Fixed overlay (bg-black/50)
- Centered card (max-w-md)
- Z-index: 50

Header:

- "Add Content" title
- Close button (X)

10.2 Three Input Sections

Section 1: File Upload

- Label: "Upload File"
- Input type="file", accept=".txt,.md,.pdf"
- Disabled during processing
- Help text: "Supports .txt, .md, .pdf"
- Tooltip on input

Section 2: URL Import

- Label: "Or paste URL"
- URL input field
- Enter key triggers import
- Disabled during processing
- Help text: "Public Google Docs, websites, articles"
- Tooltip on input

Section 3: Text Paste

- Label: "Or paste text"
- Textarea (6 rows, resize-none)
- "Add Note" button below
- Disabled during processing
- Tooltip on textarea and button

10.3 Processing State

- All inputs disabled
- Modal remains open
- Processing overlay on canvas shows spinner

---

11. Settings Modal

11.1 Modal Structure

- Scrollable overlay (fixed, overflow-y-auto)
- Large card (max-w-2xl)
- Vertical margin for scroll (my-8)

11.2 Sections

1. Theme (covered in Theme System)

2. API Configuration (covered in API Configuration)

3. Data Management Three buttons:

- Export Graph (Download icon)
- Import Graph (Upload icon, file input hidden)
- Clear All Data (Trash icon, red button)

4. Graph Legend Visual guide:

- Yellow dashed line: Surprising connection
- Gray solid line: Regular connection
- Sparkle icon: Unexpected relationship

Technical Architecture

Framework & Libraries

json

```
{
  "framework": "React 18+",
  "language": "TypeScript",
  "styling": "Tailwind CSS (utility classes only)",
  "icons": "lucide-react",
  "deployment": "Vercel",
  "runtime": "Browser (no Node.js dependencies)"
}
```

State Management

React useState for:

- Graph data (nodes, connections)
- Selected node
- UI state (modals, theme, processing)
- Filters and search
- Canvas interaction state

React useRef for:

- Canvas element
- Container element (for sizing)
- Animation frame ID

React useEffect for:

- Canvas size updates
- Theme system preferences
- LocalStorage persistence
- Graph physics simulation

No External State Libraries

- Pure React hooks
- No Redux, Zustand, Jotai, etc.
- Keep it simple and portable

Data Flow Architecture

Content Addition Flow

```
User Input → Content Extraction → AI API Call →
Concept Parsing → Node Creation →
Connection Discovery (2 phases) → Graph Update →
LocalStorage Save → Canvas Re-render
```

Selection Flow

```
Canvas Click → Coordinate Transform →
Node Hit Detection → State Update →
Popup Position Calculation → Panel Render
```

Graph Rendering Flow

```
requestAnimationFrame → Physics Simulation →
Position Updates → Canvas Clear →
Draw Connections → Draw Nodes →
Draw Labels → Loop
```

Performance Requirements

Canvas Rendering

- **Target:** 60 FPS during interaction
- **Optimization:** RequestAnimationFrame for smooth animation
- **Damping:** Prevent infinite bouncing

Node Limits

- **Recommended:** Up to 100 nodes for smooth experience
- **Maximum tested:** 200 nodes (performance degrades)
- **Future:** Add progressive loading warning

API Calls

- **Timeout:** 30 seconds max
- **Rate limiting:** User-initiated only, no auto-requests
- **Caching:** None in v1 (every request is fresh)

Memory Management

- **LocalStorage limits:** ~5-10MB typical browser limit
- **Graph size estimate:** ~1KB per node with content
- **No memory leaks:** Clean up animation frames on unmount

Browser Compatibility

Minimum Requirements

- **Chrome/Edge:** 90+
- **Firefox:** 88+
- **Safari:** 14+
- **Mobile:** iOS Safari 14+, Chrome Android 90+

Required APIs

- Canvas 2D Context
- Fetch API
- FileReader API
- LocalStorage
- RequestAnimationFrame
- CSS Variables

Not Required

- **No localStorage/sessionStorage in artifacts** (critical constraint for Claude.ai)
- No WebGL
- No Service Workers
- No Web Workers (single-threaded for simplicity)

Security & Privacy

Data Storage

- **Local-first:** All data in browser localStorage
- **No server storage:** Graph never sent to external servers
- **API calls only:** Send content to LLM for processing only

API Keys

- **Warning:** LocalStorage is not encrypted
- **Best practice:** Use environment variables for default key
- **User responsibility:** Custom keys stored at own risk

Content Privacy

- **User control:** All content uploaded by user
- **No tracking:** No analytics, no telemetry
- **No cookies:** Pure localStorage

CORS Considerations

- **URL fetching:** Subject to CORS restrictions
- **Public URLs only:** Cannot access authenticated content
- **Error handling:** Clear messaging when fetch fails

Accessibility

Keyboard Navigation

- Tab through all interactive elements
- Enter to activate buttons
- Escape to close modals
- Focus visible on all controls

Screen Reader Support

- Semantic HTML where possible
- ARIA labels on icon buttons
- Title attributes as tooltips
- Proper heading hierarchy

Color Contrast

- WCAG AA compliance minimum
- High contrast in both themes
- Not relying solely on color for information

Touch Targets

- Minimum 44x44px touch targets
- Adequate spacing between controls
- Mobile-optimized interactions

Future Enhancements

Phase 2 Features

1. **Auto-serendipity:** Periodic idea generation
2. **Export formats:** PDF, Markdown, PNG image
3. **Collaboration:** Share graphs via URL
4. **Tags & categories:** Manual organization
5. **Time-based filtering:** View graph evolution
6. **Advanced search:** Fuzzy matching, regex
7. **Keyboard shortcuts:** Power user features

Phase 3 Features

1. **Real Google Drive integration:** OAuth + API
2. **Multi-graph support:** Separate workspaces

3. **Graph templates:** Pre-built structures
4. **Custom layouts:** Tree, radial, hierarchical
5. **AI chat interface:** Query your knowledge
6. **Mobile app:** Native iOS/Android
7. **Backend sync:** Optional cloud backup

Research Ideas

1. **Automatic clustering:** Detect topic groups
2. **Graph analytics:** Centrality, communities
3. **Smart suggestions:** "You might be interested in..."
4. **Version history:** Time travel through graph states
5. **Citation tracking:** Academic paper support

Implementation Checklist

✅ Completed Features

Core Functionality

- ☑ Force-directed graph visualization with physics
- ☑ Canvas with proper containment and borders
- ☑ Node creation from AI-extracted concepts
- ☑ Connection discovery (external + internal)
- ☑ Node selection with accurate hit detection
- ☑ Node dragging with position updates
- ☑ Pan and zoom controls (with modifier key requirement)
- ☑ Search functionality with highlighting
- ☑ Connection strength filtering

Content Ingestion

- ☑ File upload (.txt, .md, .pdf)
- ☑ URL import with Google Docs support
- ☑ Direct text paste
- ☑ AI concept extraction
- ☑ Error handling for failed imports

UI Components

- ☑ Header with navigation and search
- ☑ Upload modal with three input methods
- ☑ Settings modal with all configuration options
- ☑ Welcome modal with tutorial
- ☑ Serendipity modal with idea generation
- ☑ Floating popup near selected node
- ☑ Detail panel below graph
- ☑ Graph controls panel (zoom, nodes, connections, filter)
- ☑ Zoom hint at canvas bottom

Visual Design

- ☑ Light/dark/system theme support
- ☑ Theme toggle in settings
- ☑ Responsive mobile-first design
- ☑ Desktop-optimized layouts
- ☑ Proper color scheme for both themes
- ☑ Visual distinction for surprising connections
- ☑ Hover and selected states for nodes
- ☑ Smooth transitions and animations

Data Management

- ☑ LocalStorage persistence
- ☑ Auto-save on graph changes
- ☑ Auto-load on app mount
- ☑ Export graph as JSON
- ☑ Import graph from JSON
- ☑ Clear all data functionality

API Integration

- ☑ Default API mode (environment variable)
- ☑ Custom API mode (user-provided key)
- ☑ Multi-provider support (Claude, OpenAI, custom)
- ☑ API call error handling
- ☑ Processing state indicators

UX Enhancements

- ☑ Tooltips on all interactive elements
- ☑ Welcome experience for first-time users
- ☑ Help button to reopen tutorial
- ☑ Graph legend in settings
- ☑ Disabled states during processing
- ☑ Confirmation dialogs for destructive actions

❌ Known Limitations / Not Implemented

Current Constraints

- □ OCR for handwritten notes (not in v1)
- □ Advanced PDF parsing (basic text extraction only)
- □ Private Google Drive integration (OAuth required)
- □ Authenticated URL fetching (public URLs only)
- □ Server-side storage (local-only by design)
- □ Real-time collaboration (single-user only)
- □ Undo/redo functionality
- □ Graph history/versioning

Performance Optimizations

- □ Progressive loading for large graphs (100+ nodes)
- □ Virtual scrolling for connection lists
- □ Web Workers for heavy computations
- □ Canvas rendering optimizations (culling, LOD)
- □ Debouncing for search/filter inputs

Advanced Features

- □ Auto-save to cloud backup
- □ Multiple graph workspaces
- □ Custom graph layouts (tree, radial, etc.)
- □ Tags and manual categorization
- □ Time-based filtering
- □ Advanced search (fuzzy, regex)
- □ Keyboard shortcuts
- □ Export to PDF/PNG/Markdown
- □ AI chat interface to query graph
- □ Graph analytics (centrality, clustering)
- □ Automatic topic detection
- □ Smart content suggestions

Mobile Enhancements

- □ Touch gesture optimization
- □ Mobile-specific controls
- □ Native app versions (iOS/Android)
- □ Offline-first PWA capabilities
- □ Share sheet integration

Accessibility Improvements

- □ Full keyboard navigation for graph
- □ Screen reader announcements for state changes
- □ High contrast mode
- □ Focus management in modals
- □ ARIA live regions for dynamic updates

Developer Experience

- □ Comprehensive test suite (unit, integration, e2e)
- □ Storybook for component development
- □ API documentation
- □ Contributing guidelines
- □ CI/CD pipeline
- □ Error tracking (Sentry, etc.)
- □ Analytics (privacy-preserving)

🚀 Deployment Checklist

Pre-Deploy

- ☑ Environment variable setup for default API key
- ☑ Build configuration for Vercel
- ☑ Error boundaries for graceful failures
- □ Performance testing with large datasets
- □ Cross-browser testing
- □ Mobile device testing
- □ Accessibility audit

Post-Deploy

- □ User onboarding documentation
- □ Video tutorial/demo
- □ FAQ/troubleshooting guide
- □ Feedback collection mechanism
- □ Bug reporting process
- □ Feature request tracking
- □ Usage analytics (opt-in, privacy-focused)

📋 Testing Checklist

Functional Testing

- ☑ File upload works for all supported formats
- ☑ URL import handles Google Docs correctly
- ☑ Text paste creates nodes and connections
- ☑ Node selection shows popup and panel
- ☑ Node dragging updates position
- ☑ Pan and zoom work with modifier keys
- ☑ Search highlights matching nodes
- ☑ Filter hides weak connections
- ☑ Serendipity generates ideas
- ☑ Theme switching updates colors
- ☑ API configuration saves and applies
- ☑ Export/import preserves graph state
- ☑ Clear all removes all data

Edge Cases

- □ Empty graph state
- □ Single node (no connections)
- □ Very large graph (200+ nodes)
- □ Very long node labels
- □ Very long connection reasons
- □ Malformed API responses
- □ Network failures during API calls

- □ Invalid JSON in import file
- □ Duplicate concept names
- □ Special characters in content

Browser Compatibility

- □ Chrome (latest)
- □ Firefox (latest)
- □ Safari (latest)
- □ Edge (latest)
- □ Mobile Safari (iOS 14+)
- □ Chrome Android (