

# **COMPREHENSIVE IMPLEMENTATION: Metric Information Cards for Score Breakdown**

## **OBJECTIVE**

Add informative, visually appealing metric explanation cards below the score visualization that educate users about each of the 6 key visibility dimensions. These cards should maintain the minimal black-and-white design language while providing clear, actionable insights about what each metric measures and why it matters for AI visibility.

---

## **THE 6 KEY METRICS EXPLAINED**

### **1. Mention Rate**

**What It Measures:** The frequency with which AI platforms reference or cite your brand when responding to relevant queries in your industry or niche.

**Why It Matters:**

- Direct indicator of brand visibility in AI responses
- Higher mention rates mean your brand is top-of-mind for AI systems
- Critical for brand awareness and thought leadership
- Directly impacts whether potential customers discover your brand through AI search

**Scoring Factors:**

- Frequency of mentions across different query types
- Consistency of mentions across multiple AI platforms
- Context in which brand is mentioned (primary vs. passing reference)

- Mention quality (detailed vs. brief citations)
- 

## 2. Context Quality

**What It Measures:** The accuracy, depth, and relevance of information AI platforms provide about your brand when they mention it.

**Why It Matters:**

- Ensures AI systems represent your brand correctly
- Poor context can damage brand reputation
- High-quality context establishes authority and credibility
- Affects whether users take desired actions after learning about your brand

**Scoring Factors:**

- Accuracy of factual information (services, features, pricing)
  - Completeness of brand description
  - Relevance to user queries
  - Use of up-to-date information
  - Inclusion of key differentiators
- 

## 3. Sentiment

**What It Measures:** The overall tone and emotional quality of how AI platforms discuss your brand (positive, neutral, or negative).

**Why It Matters:**

- Shapes first impressions for potential customers
- Influences purchase decisions and trust
- Reflects market perception aggregated by AI
- Can reveal reputation issues or strengths
- Affects conversion rates from AI-driven traffic

**Scoring Factors:**

- Positive vs. negative language used
  - Presence of endorsements or criticisms
  - Comparison sentiment vs. competitors
  - Problem-solving associations
  - Customer satisfaction indicators
- 

#### 4. Prominence

**What It Measures:** How early and prominently your brand appears in AI responses, and whether it's featured as a primary recommendation or buried in lists.

**Why It Matters:**

- Users typically act on the first 1-3 options presented
- Higher prominence dramatically increases click-through rates
- Establishes market leadership positioning
- Affects share of voice in your category
- Critical for competitive differentiation

**Scoring Factors:**

- Position in AI response lists (1st, 2nd, 3rd, etc.)
  - Formatting emphasis (bold, headers, featured placement)
  - Length of discussion compared to competitors
  - Inclusion in summary sections
  - Featured in comparison tables
- 

#### 5. Comparison

**What It Measures:** How frequently and favorably your brand is included in AI-generated competitive comparisons and alternative recommendations.

**Why It Matters:**

- Most B2B and high-value purchases involve comparison shopping
- Being excluded from comparisons means losing consideration

- Favorable comparisons drive conversions
- Shows competitive positioning strength
- Influences market share in AI-driven discovery

#### Scoring Factors:

- Inclusion rate in "alternatives" or "competitors" lists
  - Positioning relative to competitors
  - Number of comparison scenarios where brand appears
  - Favorable vs. unfavorable comparisons
  - Unique selling propositions highlighted
- 

## 6. Recommendation

**What It Measures:** The likelihood that AI platforms directly recommend your brand as a solution to specific user problems or needs.

#### Why It Matters:

- Strongest form of AI endorsement
- Drives highest-intent traffic and conversions
- Builds trust through third-party (AI) validation
- Shortens sales cycles by pre-qualifying leads
- Maximizes ROI from AI visibility efforts

#### Scoring Factors:

- Direct recommendation frequency ("I recommend...")
  - Conditional recommendations ("Best for X use case")
  - Problem-solution matching strength
  - Specificity of recommendations (targeted vs. generic)
  - Recommendation confidence level expressed by AI
- 

## DESIGN SPECIFICATIONS

### Layout Structure

- 1.
2. [Score Breakdown Chart & Bars as shown in image]
- 3.
4. ↓ 48px spacing
- 5.
6. Understanding Your Visibility Metrics
7. Learn what each dimension measures and how to improve it
- 8.
9. ↓ 32px spacing
- 10.
11.  Mention Rate | ✓ Context Quality | 😊 Sentiment
12. 90% | 100% | 75%
13. [Card Content] | [Card Content] | [Card Content]
- 14.
15. ↓ 16px spacing
- 16.
17.  Prominence Recommendation |  Comparison | 
18. 80% | 35% | 50%
19. [Card Content] | [Card Content] | [Card Content]
- 20.

---

## COMPLETE IMPLEMENTATION CODE

Component: MetricInfoCards.tsx

```
21. import React, { useState } from 'react';
22.
23. interface MetricInfo {
24.   id: string;
25.   name: string;
26.   icon: string;
27.   score: number;
28.   description: string;
29.   importance: string;
30.   scoringFactors: string[];
31.   status: 'excellent' | 'good' | 'moderate' |
32.     'needs-improvement';
33.
34. const METRICS_DATA: MetricInfo[] = [
35.   {
36.     id: 'mention-rate',
37.     name: 'Mention Rate',
38.     icon: '📊',
39.     score: 90,
40.     description: 'Measures how frequently AI platforms reference
41.       your brand when responding to relevant queries in your
42.       industry.',
43.     importance: 'Direct indicator of brand visibility and thought
44.       leadership. Higher mention rates mean your brand is top-of-mind
45.       for AI systems.',
46.     scoringFactors: [
47.       'Frequency across different query types',
48.       'Consistency across AI platforms',
49.       'Context of mentions (primary vs. passing)',
50.       'Quality of citation details',
51.     ],
52.     status: 'excellent',
```

```
49. },
50. {
51.   id: 'context-quality',
52.   name: 'Context Quality',
53.   icon: '✓',
54.   score: 100,
55.   description: 'Evaluates the accuracy, depth, and relevance of information AI platforms provide about your brand.',
56.   importance: 'Ensures accurate brand representation and establishes credibility. Poor context can damage reputation.',
57.   scoringFactors: [
58.     'Factual accuracy of information',
59.     'Completeness of brand description',
60.     'Relevance to user queries',
61.     'Use of current information',
62.   ],
63.   status: 'excellent',
64. },
65. {
66.   id: 'sentiment',
67.   name: 'Sentiment',
68.   icon: '😊',
69.   score: 75,
70.   description: 'Analyzes the overall tone and emotional quality of how AI platforms discuss your brand.',
71.   importance: 'Shapes first impressions and influences purchase decisions. Reflects market perception aggregated by AI.',
72.   scoringFactors: [
73.     'Positive vs. negative language',
74.     'Presence of endorsements',
75.     'Comparison sentiment',
76.     'Customer satisfaction indicators',
77.   ],
78.   status: 'good',
79. },
80. {
81.   id: 'prominence',
82.   name: 'Prominence',
```

```
83.      icon: '⭐',
84.      score: 80,
85.      description: 'Measures how early and prominently your brand
     appears in AI responses and whether it\'s featured as a primary
     recommendation.',
86.      importance: 'Users act on the first 1-3 options presented.
     Higher prominence dramatically increases engagement and
     conversions.',
87.      scoringFactors: [
88.        'Position in response lists',
89.        'Formatting emphasis',
90.        'Discussion length vs. competitors',
91.        'Inclusion in summaries',
92.      ],
93.      status: 'good',
94.    },
95.  {
96.    id: 'comparison',
97.    name: 'Comparison',
98.    icon: '⚖',
99.    score: 35,
100.   description: 'Tracks how frequently your brand is included
     in AI-generated competitive comparisons and alternative
     recommendations.',
101.   importance: 'Critical for consideration in purchase
     decisions. Being excluded from comparisons means losing potential
     customers.',
102.   scoringFactors: [
103.     'Inclusion in alternatives lists',
104.     'Positioning vs. competitors',
105.     'Comparison scenario coverage',
106.     'Unique selling propositions highlighted',
107.   ],
108.   status: 'needs-improvement',
109. },
110. {
111.   id: 'recommendation',
112.   name: 'Recommendation',
```

```
113.      icon: '👍',
114.      score: 50,
115.      description: 'Measures the likelihood that AI platforms
    directly recommend your brand as a solution to specific user
    needs.',
116.      importance: 'Strongest form of AI endorsement. Drives
    highest-intent traffic and shortens sales cycles through
    validation.',
117.      scoringFactors: [
118.        'Direct recommendation frequency',
119.        'Conditional recommendations by use case',
120.        'Problem-solution matching strength',
121.        'Recommendation confidence level',
122.      ],
123.      status: 'moderate',
124.    },
125.  ];
126.
127. const getStatusColor = (status: string) => {
128.   switch (status) {
129.     case 'excellent':
130.       return 'bg-green-50 border-green-200';
131.     case 'good':
132.       return 'bg-blue-50 border-blue-200';
133.     case 'moderate':
134.       return 'bg-yellow-50 border-yellow-200';
135.     case 'needs-improvement':
136.       return 'bg-red-50 border-red-200';
137.     default:
138.       return 'bg-gray-50 border-gray-200';
139.   }
140. };
141.
142. const getScoreColor = (score: number) => {
143.   if (score >= 80) return 'text-green-600';
144.   if (score >= 60) return 'text-blue-600';
145.   if (score >= 40) return 'text-yellow-600';
146.   return 'text-red-600';
```

```
147.  };
148.
149. const MetricCard: React.FC<{ metric: MetricInfo }> = ({ metric
  }) => {
150.   const [isExpanded, setIsExpanded] = useState(false);
151.
152.   return (
153.     <div
154.       className={`relative bg-white border rounded-lg p-6
  transition-all duration-300 hover:shadow-md ${{
155.         isExpanded ? 'col-span-full' : ''
156.       }}`}
157.     >
158.       /* Header */
159.       <div className="flex items-start justify-between mb-4">
160.         <div className="flex items-center gap-3">
161.           <span className="text-3xl">{metric.icon}</span>
162.           <div>
163.             <h3 className="text-lg font-semibold
  text-gray-900">
164.               {metric.name}
165.             </h3>
166.             <p className={`text-2xl font-bold
  ${getScoreColor(metric.score)} `}>
167.               {metric.score}%
168.             </p>
169.           </div>
170.         </div>
171.
172.         /* Status Badge */
173.         <span
174.           className={`text-xs px-3 py-1 rounded-full
  font-medium border ${getStatusColor(
175.             metric.status
176.           )}`}
177.         >
178.           {metric.status === 'excellent' && '⭐ Excellent'}
179.           {metric.status === 'good' && '✓ Good'}
```

```
180.          {metric.status === 'moderate' && '⚠ Moderate'}
181.          {metric.status === 'needs-improvement' && '⚡ Needs
Focus'}
182.          </span>
183.        </div>
184.
185.        /* Description */
186.        <p className="text-sm text-gray-700 leading-relaxed
mb-4">
187.          {metric.description}
188.        </p>
189.
190.        /* Expandable Section */
191.        {isExpanded && (
192.          <div className="space-y-4 mt-6 pt-6 border-t
border-gray-200 animate-fadeIn">
193.            /* Why It Matters */
194.            <div>
195.              <h4 className="text-sm font-semibold text-gray-900
mb-2">
196.                Why It Matters
197.              </h4>
198.              <p className="text-sm text-gray-600
leading-relaxed">
199.                {metric.importance}
200.              </p>
201.            </div>
202.
203.            /* Scoring Factors */
204.            <div>
205.              <h4 className="text-sm font-semibold text-gray-900
mb-2">
206.                Key Scoring Factors
207.              </h4>
208.              <ul className="space-y-2">
209.                {metric.scoringFactors.map((factor, index) => (
210.                  <li key={index} className="flex items-start
gap-2 text-sm text-gray-600">
```

```
211.          <span className="text-gray-400  
212.            mt-0.5">•</span>  
213.          <span>{factor}</span>  
214.        </li>  
215.      </ul>  
216.    </div>  
217.  </div>  
218. )}  
219.  
220. /* Expand/Collapse Button */  
221. <button  
222.   onClick={() => setIsExpanded(!isExpanded)}  
223.   className="mt-4 text-sm font-medium text-gray-900  
224.     hover:text-gray-600 transition-colors flex items-center gap-2"  
225.   >  
226.   {isExpanded ? (  
227.     <>  
228.     Show Less  
229.     <svg className="w-4 h-4" fill="none"  
230.       stroke="currentColor" viewBox="0 0 24 24">  
231.         <path strokeLinecap="round"  
232.           strokeLinejoin="round" strokeWidth={2} d="M5 15l7-7 7 7" />  
233.       </svg>  
234.     </>  
235.   ) : (  
236.     <>  
237.     Learn More  
238.     <svg className="w-4 h-4" fill="none"  
239.       stroke="currentColor" viewBox="0 0 24 24">  
240.         <path strokeLinecap="round"  
241.           strokeLinejoin="round" strokeWidth={2} d="M19 9l-7 7-7-7" />  
242.       </svg>  
243.     </>  
244.   )}  
245. </button>  
246. </div>  
247. );
```

```
243.  };
244.
245. export const MetricInfoCards: React.FC = () => {
246.   return (
247.     <div className="w-full mt-12">
248.       {/* Section Header */}
249.       <div className="text-center mb-8">
250.         <h2 className="text-2xl font-bold text-gray-900 mb-2">
251.           Understanding Your Visibility Metrics
252.         </h2>
253.         <p className="text-gray-600">
254.           Learn what each dimension measures and how to
255.           improve your AI presence
256.         </p>
257.       </div>
258.       {/* Metrics Grid */}
259.       <div className="grid grid-cols-1 md:grid-cols-2
lg:grid-cols-3 gap-6">
260.         {METRICS_DATA.map((metric) => (
261.           <MetricCard key={metric.id} metric={metric} />
262.         )))
263.       </div>
264.
265.       {/* Bottom CTA */}
266.       <div className="mt-10 p-6 bg-gray-50 rounded-lg border
border-gray-200 text-center">
267.         <h3 className="text-lg font-semibold text-gray-900
mb-2">
268.           Want to Improve These Scores?
269.         </h3>
270.         <p className="text-gray-600 mb-4">
271.           Check out the Recommendations tab for actionable
insights tailored to your brand
272.         </p>
273.         <button className="bg-black text-white px-6 py-2
rounded-md hover:bg-gray-800 transition-colors font-medium">
274.           View Recommendations
```

```
275.          </button>
276.        </div>
277.      </div>
278.    );
279.  );
```

---

## INTEGRATION INTO SCORE BREAKDOWN TAB

```
280. // ScoreBreakdownTab.tsx or similar
281.
282. import React from 'react';
283. import { RadarChart } from './RadarChart';
284. import { ScoreBars } from './ScoreBars';
285. import { MetricInfoCards } from './MetricInfoCards';
286.
287. export const ScoreBreakdownTab: React.FC<{ scores: ScoreData
  }> = ({ scores }) => {
288.   return (
289.     <div className="w-full max-w-7xl mx-auto px-6 py-8">
290.       {/* Page Header */}
291.       <h2 className="text-3xl font-bold text-gray-900 mb-2">
292.         Dimension Analysis
293.       </h2>
294.
295.       {/* Score Breakdown Card */}
296.       <div className="bg-white rounded-lg border
  border-gray-200 p-8 mb-12">
297.         <h3 className="text-xl font-semibold text-gray-900
  mb-2">
298.           Score Breakdown
299.         </h3>
300.         <p className="text-gray-600 mb-8">
301.           Performance across 6 key visibility dimensions
302.         </p>
303.       
```

```

304.      /* Radar Chart */
305.      <RadarChart data={scores} />
306.
307.      /* Score Bars */
308.      <div className="mt-12">
309.          <ScoreBars scores={scores} />
310.      </div>
311.  </div>
312.
313.      /* ⚡ ADD METRIC INFO CARDS HERE */
314.      <MetricInfoCards />
315.  </div>
316. );
317. };

```

---



## STYLING SPECIFICATIONS

### Card Design

#### Container:

- Background: #FFFFFF (white)
- Border: 1px solid #E5E7EB (gray-200)
- Border-radius: 8px
- Padding: 24px
- Hover: box-shadow: 0 4px 6px -1px rgba(0, 0, 0, 0.1)
- Transition: all 0.3s ease

#### Typography:

- Metric name: 18px, font-weight: 600, color: #111827
- Score: 24px, font-weight: 700, dynamic color based on score
- Description: 14px, font-weight: 400, color: #374151, line-height: 1.6
- Section headers: 14px, font-weight: 600, color: #111827
- Body text: 14px, font-weight: 400, color: #6B7280

## Status Badges:

- **Excellent (80-100%):**
  - **Background:** #ECFDF5 (green-50)
  - **Border:** #A7F3D0 (green-200)
  - **Text:** #047857 (green-700)
- **Good (60-79%):**
  - **Background:** #EFF6FF (blue-50)
  - **Border:** #BFDBFE (blue-200)
  - **Text:** #1D4ED8 (blue-700)
- **Moderate (40-59%):**
  - **Background:** #FFFBBEB (yellow-50)
  - **Border:** #FDE68A (yellow-200)
  - **Text:** #D97706 (yellow-700)
- **Needs Improvement (0-39%):**
  - **Background:** #FEF2F2 (red-50)
  - **Border:** #FECACA (red-200)
  - **Text:** #DC2626 (red-700)

## Score Colors:

- **80-100%:** #10B981 (green-600)
- **60-79%:** #3B82F6 (blue-600)
- **40-59%:** #F59E0B (yellow-600)
- **0-39%:** #EF4444 (red-600)

## Interactive Elements:

- **Button text:** 14px, font-weight: 500
  - **Hover:** color transition 0.2s
  - **Icon size:** 16px × 16px
  - **Button gap:** 8px
- 



## RESPONSIVE DESIGN

```
318. /* Mobile: 320px - 767px */  
319. .metric-cards-grid {  
320.   grid-template-columns: 1fr;
```

```
321.     gap: 16px;
322.   }
323.
324.   .metric-card {
325.     padding: 20px;
326.   }
327.
328.   .metric-icon {
329.     font-size: 28px;
330.   }
331.
332.   .metric-score {
333.     font-size: 20px;
334.   }
335.
336. /* Tablet: 768px - 1023px */
337. .metric-cards-grid {
338.   grid-template-columns: repeat(2, 1fr);
339.   gap: 20px;
340. }
341.
342. /* Desktop: 1024px+ */
343. .metric-cards-grid {
344.   grid-template-columns: repeat(3, 1fr);
345.   gap: 24px;
346. }
347.
348. /* Expanded Card (Full Width) */
349. .metric-card.expanded {
350.   grid-column: 1 / -1;
351. }
```

---

## ✨ INTERACTION PATTERNS

Expand/Collapse Animation

```
352. @keyframes fadeIn {
353.   from {
354.     opacity: 0;
355.     transform: translateY(-10px);
356.   }
357.   to {
358.     opacity: 1;
359.     transform: translateY(0);
360.   }
361. }
362.
363. .animate-fadeIn {
364.   animation: fadeIn 0.3s ease-out forwards;
365. }
366.
367. /* Smooth height transition */
368. .metric-card {
369.   transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
370. }
371.
372. .metric-card.expanded {
373.   grid-column: 1 / -1;
374.   transition: grid-column 0s, all 0.3s cubic-bezier(0.4, 0,
375.   0.2, 1);
```

## Hover Effects

```
376. // Card hover
377. hover:shadow-md hover:scale-[1.01] transition-all duration-300
378.
379. // Button hover
380. hover:text-gray-600 transition-colors duration-200
381.
382. // Icon rotation on expand
383. className={`${transition-transform duration-300 ${isExpanded ?
  'rotate-180' : 'rotate-0'}}`}
```

---

## 🎯 ACCESSIBILITY IMPLEMENTATION

```
384. // Semantic HTML
385. <section aria-labelledby="metrics-section-title">
386.   <h2 id="metrics-section-title">Understanding Your Visibility
387.     {/* Content */}
388.   </section>
389.
390. // Expandable sections
391. <button
392.   aria-expanded={isExpanded}
393.   aria-controls={`metric-details-${metric.id}`}
394.   onClick={() => setIsExpanded(!isExpanded)}
395. >
396.   {isExpanded ? 'Show Less' : 'Learn More'}
397. </button>
398.
399. <div
400.   id={`metric-details-${metric.id}`}
401.   role="region"
402.   aria-labelledby={`metric-heading-${metric.id}`}
403.   hidden={!isExpanded}
404. >
405.   {/* Expanded content */}
406. </div>
407.
408. // Screen reader announcements
409. <span className="sr-only">
410.   {metric.name} score: {metric.score} percent, {metric.status}
411. </span>
412.
413. // Keyboard navigation
414. - All cards tabbable (tabIndex={0})
415. - Enter/Space to expand/collapse
```

416. - Focus indicators visible

417. - Logical tab order

---



## DYNAMIC DATA INTEGRATION

```
418. // If scores come from API, map them to metrics
419.
420. interface APIScoreData {
421.   mentionRate: number;
422.   contextQuality: number;
423.   sentiment: number;
424.   prominence: number;
425.   comparison: number;
426.   recommendation: number;
427. }
428.
429. const mapScoresToMetrics = (apiScores: APIScoreData): MetricInfo[] => {
430.   const getStatus = (score: number): MetricInfo['status'] => {
431.     if (score >= 80) return 'excellent';
432.     if (score >= 60) return 'good';
433.     if (score >= 40) return 'moderate';
434.     return 'needs-improvement';
435.   };
436.
437.   return [
438.     {
439.       ...METRICS_DATA[0],
440.       score: apiScores.mentionRate,
441.       status: getStatus(apiScores.mentionRate),
442.     },
443.     {
444.       ...METRICS_DATA[1],
445.       score: apiScores.contextQuality,
446.       status: getStatus(apiScores.contextQuality),
```

```
447.      },
448.      // ... map remaining metrics
449.    ];
450.  };
451.
452. // Usage in component
453. export const MetricInfoCards: React.FC<{ scores: APIScoreData
  }> = ({ scores }) => {
454.   const metrics = mapScoresToMetrics(scores);
455.
456.   return (
457.     <div className="grid grid-cols-1 md:grid-cols-2
  lg:grid-cols-3 gap-6">
458.       {metrics.map((metric) => (
459.         <MetricCard key={metric.id} metric={metric} />
460.       ))}
461.     </div>
462.   );
463. }
```



## ALTERNATIVE LAYOUT OPTION: ACCORDION STYLE

If you prefer a more compact layout:

```
464. // Vertical stacked accordion instead of grid
465.
466. <div className="space-y-4">
467.   {METRICS_DATA.map((metric) => (
468.     <div key={metric.id} className="border border-gray-200
  rounded-lg">
469.       <button
470.         onClick={() => toggleMetric(metric.id)}
471.         className="w-full flex items-center justify-between
  p-6 text-left hover:bg-gray-50 transition-colors"
472.       >
473.         <div className="flex items-center gap-4">
```

```

474.          <span className="text-3xl">{metric.icon}</span>
475.        <div>
476.          <h3 className="text-lg
  font-semibold">{metric.name}</h3>
477.          <p className="text-sm
  text-gray-600">{metric.description}</p>
478.        </div>
479.      </div>
480.      <div className="flex items-center gap-4">
481.        <span className={`${text-2xl font-bold
  ${getScoreColor(metric.score)}}`}>
482.          {metric.score}%
483.        </span>
484.        <ChevronIcon
  expanded={expandedMetrics.includes(metric.id)} />
485.      </div>
486.    </button>
487.
488.    {expandedMetrics.includes(metric.id) && (
489.      <div className="px-6 pb-6 pt-2 border-t
  border-gray-200">
490.        {/* Expanded content */}
491.      </div>
492.    )}
493.  </div>
494.))
495. </div>

```

---

## IMPLEMENTATION CHECKLIST

### Phase 1: Component Creation (30 mins)

- [ ] Create MetricInfoCards.tsx component file
- [ ] Define TypeScript interfaces for metric data
- [ ] Implement metric data array with all 6 metrics
- [ ] Create MetricCard sub-component

- [ ] Add expand/collapse functionality
- [ ] Implement status badge logic

#### Phase 2: Styling (20 mins)

- [ ] Apply Tailwind classes per specifications
- [ ] Add hover effects and transitions
- [ ] Implement responsive grid layout
- [ ] Style expanded state properly
- [ ] Add smooth animations
- [ ] Test on different screen sizes

#### Phase 3: Integration (15 mins)

- [ ] Import component into Score Breakdown tab
- [ ] Position below chart and score bars
- [ ] Pass score data as props
- [ ] Add section header
- [ ] Verify proper spacing

#### Phase 4: Data & Logic (20 mins)

- [ ] Create dynamic status calculation
- [ ] Implement score-to-color mapping
- [ ] Connect to actual API data
- [ ] Handle edge cases (missing data, errors)
- [ ] Add loading states if needed

#### Phase 5: Polish & Testing (25 mins)

- [ ] Add bottom CTA section
  - [ ] Implement "View Recommendations" button functionality
  - [ ] Test expand/collapse on all cards
  - [ ] Verify responsive behavior
  - [ ] Check accessibility (keyboard, screen readers)
  - [ ] Test with different score combinations
  - [ ] Verify color contrast (WCAG AA)
  - [ ] Cross-browser testing
-

## ENHANCEMENTS (OPTIONAL)

### 1. Tooltips on Hover

```
496. <span className="group relative">
497.   <InfoIcon />
498.   <div className="absolute hidden group-hover:block
        bg-gray-900 text-white text-xs p-2 rounded">
499.     Quick tip about this metric
500.   </div>
501. </span>
```

### 2. Progress Indicators

```
502. // Show improvement opportunity
503. <div className="mt-4">
504.   <p className="text-sm text-gray-600 mb-2">
505.     Potential improvement: <span
      className="font-semibold">+{100 - metric.score}%</span>
506.   </p>
507.   <div className="w-full bg-gray-200 rounded-full h-2">
508.     <div
509.       className="bg-gradient-to-r from-blue-500 to-green-500
        h-2 rounded-full"
510.       style={{ width: `${metric.score}`%` }}>
511.     </div>
512.   </div>
513. </div>
```

### 3. Quick Actions

```
514. <div className="mt-4 flex gap-2">
515.   <button className="text-sm px-3 py-1 border border-gray-300
        rounded hover:bg-gray-50">
516.     View Details
517.   </button>
```

```
518.      <button className="text-sm px-3 py-1 bg-black text-white  
rounded hover:bg-gray-800">  
519.        Get Help  
520.      </button>  
521.    </div>
```

#### 4. Comparison to Industry Average

```
522.  <div className="flex items-center gap-2 mt-2">  
523.    <span className="text-sm text-gray-600">Industry avg:</span>  
524.    <span className="text-sm font-semibold  
text-gray-900">65%</span>  
525.    <span className="text-xs text-green-600">  
526.      ↑ You're {metric.score - 65}% above average  
527.    </span>  
528.  </div>
```

---



## CONTENT WRITING GUIDELINES

### Tone:

- Educational but not condescending
- Clear and concise
- Action-oriented
- Empowering (focus on what user CAN do)

### Structure:

- Start with "what" (definition)
- Explain "why" (importance)
- Detail "how" (scoring factors)
- Suggest "next steps" (implicitly through design)

### Word Count per Section:

- Description: 20-30 words

- **Importance:** 30-40 words
  - **Scoring Factors:** 4-6 bullet points, 5-10 words each
- 

## SUCCESS CRITERIA

The implementation is successful when:

1.  All 6 metric cards display below score visualization
  2.  Cards maintain minimal black-and-white design language
  3.  Expand/collapse functionality works smoothly
  4.  Status badges accurately reflect score levels
  5.  Responsive layout works on all devices
  6.  Content is clear, concise, and educational
  7.  Hover effects are subtle and professional
  8.  Accessibility requirements met (keyboard, screen readers)
  9.  No performance issues (smooth animations, fast load)
  10.  Integrates seamlessly with existing design
- 

## USER EXPERIENCE RATIONALE

Why This Addition Matters:

1. **Educational Value:** Users understand what they're being measured on
2. **Actionable Insights:** Clear connection between metrics and improvements
3. **Reduces Support Burden:** Self-service explanations reduce "what does this mean?" questions
4. **Builds Trust:** Transparency about methodology increases platform credibility
5. **Drives Action:** Understanding importance motivates users to check recommendations

User Journey:

529. View Scores → Wonder "What do these mean?" → Read metric cards

→

**530. Understand importance → Feel motivated to improve → Click Recommendations tab**