

I would like to overhaul the Ai Adoption readiness assessment tool. Please redesign it based on the reference documents: the PRD, scoring engine and sample n8n payload. Please confirm you're able to find, read and understand the documents and what im asking you to do

I'll search for and read the reference documents you mentioned. Let me find them first.

```
Glob**/*PRD*
```

```
Glob**/*scoring*
```

```
Glob**/*n8n*
```

```
Glob**/*payload*
```

```
Glob**/tasks/**
```

```
Glob**/docs/**
```

Let me search more broadly for any reference documents, PDFs, or markdown files:

```
Glob**/*.md
```

```
Glob**/*.pdf
```

```
Glob**/*.json
```

```
Glob**/*assessment*
```

```
Glob**/*readiness*
```

Let me also check the main repository and parent directories for these reference documents:

```
Glob**/*
```

```
Bashdir "C:\Users\gwiny\Documents\Vantage Leap Website" /s /b 2>nul |  
findstr /i "PRD scoring n8n payload"
```

```
Exit code 1
```

```
dir: cannot access '/s': No such file or directory
```

```
dir: cannot access '/b': No such file or directory
```

```
dir: cannot access '2': No such file or directory
```

```
FINDSTR: Cannot open PRD scoring n8n payload
```

Let me search the broader Documents folder:

```
BashGet-ChildItem -Path "C:\Users\gwiny\Documents" -Recurse -Include  
"*PRD*", "*scoring*", "*n8n*", "*payload*", "*assessment*" 2>$null |  
Select-Object -ExpandProperty FullName
```

```
Exit code 127
```

```
/usr/bin/bash: line 1: : No such file or directory
```

```
/usr/bin/bash: line 1: Select-Object: command not found
```

```
Bashfind "/c/Users/gwiny/Documents" -type f \( -iname "*PRD*" -o  
-iname "*scoring*" -o -iname "*n8n*" -o -iname "*payload*" \)  
2>/dev/null | head -50
```

```
/c/Users/gwiny/Documents/Vantage Leap Website/vantage-leap/Ref  
Docs/scoring-engine.js
```

```
/c/Users/gwiny/Documents/Vantage Leap
```

```
Website/VL2/node_modules/@angular/build/node_modules/vite/types/hmrPa  
yload.d.ts
```

```
/c/Users/gwiny/Documents/Vantage Leap
```

```
Website/VL2/node_modules/vite/types/hmrPayload.d.ts
```

```
... +4 lines
```

I found the reference documents. Let me read all three of them:

```
ReadC:\Users\gwiny\Documents\Vantage Leap Website\vantage-leap\Ref
Docs\12.4.25 - AI Adoption Readiness Assessment - Combined PRD -
Updated.pdf
```

```
Read 1 line
```

```
ReadC:\Users\gwiny\Documents\Vantage Leap Website\vantage-leap\Ref
Docs\sample-n8n-payload.json
```

```
Read 527 lines
```

```
ReadC:\Users\gwiny\Documents\Vantage Leap Website\vantage-leap\Ref
Docs\scoring-engine.js
```

```
Read 1,042 lines
```

I've successfully found and read all three reference documents. Here's my understanding:

Documents Found

1. PRD (47 pages) - 12.4.25 - AI Adoption Readiness Assessment - Combined PRD - Updated.pdf
2. Scoring Engine - `scoring-engine.js` (1038 lines)
3. Sample N8N Payload - `sample-n8n-payload.json` (522 lines)

What I Understand

The PRD outlines a comprehensive AI Adoption Readiness Assessment Tool v2.0 with:

Structure

- 7 Sections with 33 total questions (across Quick/Standard/Deep Dive tiers)
- 7 Scoring Dimensions: Tech Infrastructure (20%), Process Maturity (15%), Data Readiness (15%), Budget & Resources (15%), AI Knowledge (15%), Org Readiness (10%), Strategic Alignment (10%)
- 5 Readiness Levels: Foundation Building (0-25), Early Stage (26-45), Developing (46-65), Ready (66-80), Advanced (81-100)

Key Features

- Cross-dimensional validation rules (6 flags for edge cases)
- AI analysis for free-text questions
- Lead capture with gated results
- Comprehensive N8N webhook payload structure
- Roadmap generation with phased implementation plans

The Scoring Engine

Implements the scoring logic with question definitions, scoring maps, dimension calculators, validation rules, and payload builder for N8N.

The Payload

Shows the complete data structure sent to N8N, including meta, lead info, scoring results, business profile, pain points, current state, goals, free-text for AI analysis, and raw responses.

Your Request

You want me to redesign the current AI Readiness Assessment in the website (currently in `App.tsx`) to match this new v2.0 specification.

Is that correct? Before I proceed, I'd like to confirm:

1. Should I implement the full "Standard" assessment (18 questions) or the complete "Deep Dive" (33 questions)?
2. Should the assessment be a separate component/file, or remain integrated in `App.tsx`?
3. Do you want the lead capture form before showing full results (as specified in the PRD)?

yes that is correct please redesign the current AI Readiness Assessment in the website (currently in `App.tsx`) to match this new v2.0 specification. 1. implement the complete "Deep Dive" (33 questions) 2. I don't know if the assessment be a separate component/file, or remain integrated in `App.tsx`. what do you recommend? 3. I want the lead capture to be at the beginning of the assessment. The user will enter the following: First Name (required), Last Name(required), Phone Number(required), Email Address(required), Business Name(required), Website URL(optional) and then click "Start Assessment". Only then should the questions appear.

I recommend creating a separate component file for the assessment. Here's why:

Reasons for separate file:

1. The assessment is complex (33 questions, scoring engine, validation, state management)
2. Keeps `App.tsx` manageable and focused on page layout
3. Easier to maintain and test independently
4. The scoring engine alone is substantial - better to co-locate related logic

Let me first read the current assessment implementation to understand what exists:

```
ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.tsx
```

```
Read 1,263 lines
```

```
ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\constants.tsx
```

```
Read 659 lines
```

```
ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\types.ts
```

```
Read 73 lines
```

Now I have a complete picture of the codebase. Let me create a detailed plan in

`tasks/todo.md`:

```
Bashmkdir -p
```

```
"C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\tasks"
(No content)
```

This session is being continued from a previous conversation that ran out of context. The conversation is summarized below: Analysis: Let me chronologically analyze the conversation to capture all important details:

1. Initial User Request: User wants to overhaul the AI Adoption Readiness Assessment tool based on three reference documents: PRD, scoring engine, and sample n8n payload.
2. Document Discovery Phase: I searched for the reference documents using various glob patterns and bash commands. Initially couldn't find them, but eventually located them at:
 - C:\Users\gwiny\Documents\Vantage Leap Website\vantage-leap\Ref Docs\12.4.25 - AI Adoption Readiness Assessment - Combined PRD - Updated.pdf
 - C:\Users\gwiny\Documents\Vantage Leap Website\vantage-leap\Ref Docs\sample-n8n-payload.json
 - C:\Users\gwiny\Documents\Vantage Leap Website\vantage-leap\Ref Docs\scoring-engine.js
3. Document Reading Phase: I read all three reference documents:
 - PRD (47 pages): Comprehensive specification for AI Adoption Readiness Assessment Tool v2.0
 - Scoring Engine (scoring-engine.js): 1038 lines of JavaScript implementing the scoring logic
 - Sample N8N Payload (sample-n8n-payload.json): 522 lines showing the complete data structure
4. User Confirmation: User confirmed I understood correctly and provided specific requirements:
 - Implement complete "Deep Dive" (33 questions)
 - Asked for recommendation on separate component vs integrated in App.tsx
 - Lead capture at BEGINNING of assessment with fields: First Name (required), Last Name (required), Phone Number (required), Email Address (required), Business Name (required), Website URL (optional)
5. My Recommendation: I recommended creating a separate component file because:
 - Assessment is complex (33 questions, scoring engine, validation, state management)
 - Keeps App.tsx manageable
 - Easier to maintain and test independently
 - Scoring engine is substantial - better to co-locate
6. Current State Analysis: I read the existing codebase files:
 - App.tsx: Contains the existing AIReadinessAssessment component (lines 762-919) with basic quiz functionality
 - constants.tsx: Contains QUIZ_QUESTIONS and QUIZ_RESULTS arrays with 16 simple questions

- `types.ts`: Type definitions including `QuizQuestion`, `QuizOption`, `QuizResult`, `LeadInfo`
7. Last Action: I created a tasks directory to organize the work, then the conversation was interrupted for summary.

Key Technical Details from Reference Documents:

PRD Structure:

- 7 Assessment Sections: Business Profile, Technology Infrastructure, Business Processes & Pain Points, Budget & Resources, AI Knowledge & Expectations, Data & Compliance, Goals & Timeline
- 7 Scoring Dimensions with weights: Tech Infrastructure (20%), Process Maturity (15%), Data Readiness (15%), Budget & Resources (15%), AI Knowledge (15%), Org Readiness (10%), Strategic Alignment (10%)
- 5 Readiness Levels: Foundation Building (0-25), Early Stage (26-45), Developing (46-65), Ready (66-80), Advanced (81-100)
- Cross-dimensional validation rules (6 flags)
- AI analysis prompts for free-text questions

Scoring Engine Key Components:

- `QUESTIONS` object with all question definitions
- `SCORING` maps for score lookups
- Dimension calculators: `calculateTechInfrastructure()`, `calculateProcessMaturity()`, `calculateDataReadiness()`, `calculateBudgetResources()`, `calculateAIKnowledge()`, `calculateOrgReadiness()`, `calculateStrategicAlignment()`
- `runValidation()` for cross-dimensional validation
- `getReadinessLevel()` for determining badge/tier
- `calculateAssessmentScore()` main scoring function
- `buildN8NPayload()` for webhook submission

Sample Payload Structure:

- `meta`: assessment version, timestamp, source, session info
- `lead`: email, firstName, lastName, companyName, phone, consultationRequested
- `scoring`: overallScore, readinessLevel, dimensions (with normalized scores), validationFlags
- `businessProfile`, `painPointsAndOpportunities`, `currentState`, `goals`
- `freeTextForAIAnalysis`, `completeQuestionsAndAnswers`, `rawResponses`

Summary:

1. Primary Request and Intent: The user wants to completely redesign the existing AI Adoption Readiness Assessment tool in the Vantage Leap website to match the new v2.0 specification defined in three reference documents. Specific requirements:

1. Implement the complete "Deep Dive" assessment with all 33 questions
 2. Create lead capture form at the BEGINNING of the assessment (before questions appear)
 3. Lead capture fields: First Name (required), Last Name (required), Phone Number (required), Email Address (required), Business Name (required), Website URL (optional)
 4. User clicks "Start Assessment" button, then questions appear
 5. Follow PRD v2.0 specification for scoring, dimensions, validation rules, and payload structure
2. Key Technical Concepts:
1. 7 Assessment Sections: Business Profile, Technology Infrastructure, Business Processes & Pain Points, Budget & Resources, AI Knowledge & Expectations, Data & Compliance, Goals & Timeline
 2. 7 Scoring Dimensions: Tech Infrastructure (20%), Process Maturity (15%), Data Readiness (15%), Budget & Resources (15%), AI Knowledge (15%), Org Readiness (10%), Strategic Alignment (10%)
 3. 5 Readiness Levels: Foundation Building (0-25), Early Stage (26-45), Developing (46-65), Ready (66-80), Advanced (81-100)
 4. Cross-dimensional validation: 6 flags for edge cases (EXPECTATION_MANAGEMENT, MANAGED_SERVICES_CANDIDATE, TIMELINE_ADJUSTMENT, QUICK_WIN_OPPORTUNITY, COMPLIANCE_FIRST, CHANGE_MANAGEMENT_CRITICAL)
 5. N8N webhook integration for payload submission
 6. Question types: single_select, multi_select, dropdown, scale (1-5), free_text
 7. Conditional questions: DC-05 only appears if DC-02 or DC-03 indicate regulated industry
3. Files and Code Sections:
1. Reference Documents (Read):
 - C:\Users\gwiny\Documents\Vantage Leap Website\vantage-leap\Ref Docs\12.4.25 - AI Adoption Readiness Assessment - Combined PRD - Updated.pdf - 47-page specification document
 - C:\Users\gwiny\Documents\Vantage Leap Website\vantage-leap\Ref Docs\scoring-engine.js - 1038 lines of scoring logic implementation
 - C:\Users\gwiny\Documents\Vantage Leap Website\vantage-leap\Ref Docs\sample-n8n-payload.json - 522 lines showing complete payload structure
 2. App.tsx (Current Implementation to Replace):
 - Contains existing `AIReadinessAssessment` component (lines 762-919)
 - Basic quiz with simple scoring, no lead capture at beginning
 - Current implementation is too simple compared to v2.0 requirements

```

const AIReadinessAssessment = () => {
  const [step, setStep] = useState<'quiz' | 'result'>('quiz');
  const [hasStarted, setHasStarted] = useState(false);
  const [currentQuestion, setCurrentQuestion] = useState(0);
  const [answers, setAnswers] = useState<{questionId: number, points: number,
text?: string}>[]>([]);
  // ... simplified implementation
};

```

3.

4. constants.tsx (Contains Current Quiz Data):

- QUIZ_QUESTIONS: 16 simple questions (needs replacing with 33 Deep Dive questions)
- QUIZ_RESULTS: 4 result tiers (needs replacing with 5 readiness levels)

5. types.ts (Type Definitions):

- Contains QuizQuestion, QuizOption, QuizResult, LeadInfo interfaces
- Will need significant expansion for new assessment types

scoring-engine.js (Reference - Key Functions):

```

const QUESTIONS = { /* 33 question definitions */ };
const SCORING = { /* scoring maps */ };
function calculateAssessmentScore(responses) { /* main scoring */ }
function buildN8NPayload(responses, leadData, metadata) { /* payload builder */ }
function runValidation(dimensions, r) { /* cross-dimensional validation */ }
function getReadinessLevel(score) { /* returns level/badge/color/tier */ }

```

6.

4. Errors and fixes:

1. No errors encountered yet - still in planning/discovery phase

5. Problem Solving:

1. Successfully located reference documents after initial search failures (used `find` command with `bash` instead of Windows `dir`)
2. Analyzed existing codebase structure to understand what needs to be replaced
3. Recommended separate component file approach for maintainability

6. All user messages:

1. "I would like to overhaul the Ai Adoption readiness assessment tool. Please redesign it based on the reference documents: the PRD, scoring engine and sample n8n payload. Please confirm you're able to find, read and understand the documents and whatim asking you to do"
2. "yes that is correct please redesign the current AI Readiness Assessment in the website (currently in App.tsx) to match this new v2.0 specification. 1. implement the complete "Deep Dive" (33 questions) 2. I don't know if the assessment be a separate component/file, or remain integrated in App.tsx. what do you recommend? 3. I want the lead capture to be at the beginning of the assessment. The user will enter the following: First Name (required), Last Name(required),

Phone Number(required), Email Address(required), Business Name(required), Website URL(optional) and then click "Start Assessment". Only then should the questions appear."

7. Pending Tasks:

1. Create new AIReadinessAssessment component file with v2.0 specification
2. Implement all 33 Deep Dive questions with proper question types
3. Implement lead capture form at beginning of assessment
4. Implement 7-dimension scoring engine
5. Implement cross-dimensional validation rules
6. Implement N8N payload builder
7. Create proper TypeScript types for new assessment
8. Update App.tsx to use new component
9. Style according to existing design system (neo-brutalist with pa-black, pa-sage, pa-peach colors)

Current Work: I had just finished reading all reference documents and the existing codebase. I made the recommendation to create a separate component file. I created a tasks directory for organizing work:

```
mkdir -p "C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\tasks"
```

8.

9. The conversation was interrupted before I could begin creating the new assessment component.

10. Optional Next Step: Begin creating the new AI Readiness Assessment implementation by:

1. Creating new TypeScript types for the v2.0 assessment in `types.ts`
2. Creating the scoring engine module (can adapt from `scoring-engine.js` reference)
3. Creating the main `AIReadinessAssessment.tsx` component with lead capture form at the beginning

11. Direct quote from user's most recent request: "I want the lead capture to be at the beginning of the assessment. The user will enter the following: First Name (required), Last Name(required), Phone Number(required), Email Address(required), Business Name(required), Website URL(optional) and then click 'Start Assessment'. Only then should the questions appear.". Please continue the conversation from where we left it off without asking the user any further questions. Continue with the last task that you were asked to work on.

I'll continue with the AI Readiness Assessment redesign. Let me first create the todo plan, then start implementing the new assessment based on the v2.0 specification.

Update Todos

```
Creating todo.md plan file with implementation checklist
Create new TypeScript types for v2.0 assessment
Create scoring engine module (assessmentScoring.ts)
Create assessment questions data file
```



```
Create AIReadinessAssessment.tsx component with lead capture
Update App.tsx to use new component
Test and verify the implementation
```

WriteC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\tasks\todo.md

AI Readiness Assessment v2.0 Redesign

Overview

Redesign the existing AI Adoption Readiness Assessment tool to match the new v2.0 specification from the PRD, implementing all 33 Deep Dive questions with proper scoring and N8N integration.

Requirements

- Lead capture at BEGINNING of assessment (before questions)
- Lead fields: First Name (req), Last Name (req), Phone (req), Email (req), Business Name (req), Website URL (opt)
- 33 Deep Dive questions across 7 sections
- 7 scoring dimensions with weighted scoring
- 5 readiness levels
- Cross-dimensional validation rules
- N8N webhook payload submission

Implementation Checklist

1. Types & Interfaces

- [] Create assessment-specific types in `assessmentTypes.ts`
 - LeadData interface
 - Question types (single_select, multi_select, dropdown, scale, free_text)

Show full diff (48 more lines)

Update Todos

~~Create todo.md plan file with implementation checklist~~

Creating new TypeScript types for v2.0 assessment

Create scoring engine module (assessmentScoring.ts)

Create assessment questions data file

Create AIReadinessAssessment.tsx component with lead capture

Update App.tsx to use new component

Test and verify the implementation

Now let me read the scoring engine reference to get the exact question and scoring details:

ReadC:\Users\gwiny\Documents\Vantage Leap Website\vantage-leap\Ref Docs\scoring-engine.js

Read 1,042 lines

Now I have all the details. Let me create the TypeScript types first:

WriteC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\assessmentTypes.ts

```
/**
 * AI Adoption Readiness Assessment - Types
 * Version 2.0
 */

//
=====
// LEAD CAPTURE TYPES
//
=====

export interface LeadData {
  firstName: string;
  lastName: string;
  phone: string;
  email: string;
  businessName: string;
  websiteUrl?: string;
  consultationRequested?: boolean;
}

//
=====
```

Show full diff (203 more lines)

Update Todos

- ~~Create todo.md plan file with implementation checklist~~
- ~~Create new TypeScript types for v2.0 assessment~~
- Creating scoring engine module
- Create assessment questions data file
- Create AIReadinessAssessment.tsx component with lead capture
- Update App.tsx to use new component
- Test and verify the implementation

Now let me create the scoring engine module in TypeScript:

WriteC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\assessmentScoring.ts

```
/**
 * AI Adoption Readiness Assessment - Scoring Engine
 * Version 2.0
 */
```

```

import {
  AssessmentResponses,
  DimensionScore,
  DimensionScores,
  ReadinessLevel,
  ValidationFlag,
  ScoringResult,
  LeadData,
  N8NPayload
} from './assessmentTypes';
import { ASSESSMENT_QUESTIONS } from './assessmentQuestions';

```

```
//
```

```
=====
=====
```

```
// SCORING MAPS
```

```
//
```

```
=====
=====
```

Show full diff (539 more lines)

Update Todos

~~Create todo.md plan file with implementation checklist~~

~~Create new TypeScript types for v2.0 assessment~~

~~Create scoring engine module (assessmentScoring.ts)~~

Creating assessment questions data file

Create AIReadinessAssessment.tsx component with lead capture

Update App.tsx to use new component

Test and verify the implementation

Now let me create the assessment questions data file:

Write C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\assessmentQuestions.ts

```

/**
 * AI Adoption Readiness Assessment - Question Definitions
 * Version 2.0 - Deep Dive (33 Questions)
 */

import { AssessmentQuestion, AssessmentResponses } from
'./assessmentTypes';

export const ASSESSMENT_SECTIONS = [
  { id: 'BP', title: 'Business Profile', description: 'Tell us about
your business' },
  { id: 'TI', title: 'Technology Infrastructure', description: 'Your
current tech stack' },

```

```

    { id: 'PP', title: 'Business Processes', description: 'How your
business operates' },
    { id: 'BR', title: 'Budget & Resources', description: 'Your
resources for AI adoption' },
    { id: 'AK', title: 'AI Knowledge', description: 'Your familiarity
with AI tools' },
    { id: 'DC', title: 'Data & Compliance', description: 'Data
management and regulations' },
    { id: 'GT', title: 'Goals & Timeline', description: 'Your AI
adoption goals' }
  ];

export const ASSESSMENT_QUESTIONS: AssessmentQuestion[] = [
  //
  =====
  // Section 1: Business Profile & Foundation (BP01-BP06)

```

Show full diff (334 more lines)

Update Todos

- ~~Create todo.md plan file with implementation checklist~~
- ~~Create new TypeScript types for v2.0 assessment~~
- ~~Create scoring engine module (assessmentScoring.ts)~~
- ~~Create assessment questions data file~~
- Creating AIReadinessAssessment.tsx component
- Update App.tsx to use new component
- Test and verify the implementation

Now let me read the current App.tsx to understand the styling and component structure:

ReadC:\Users\gwin\claude-worktrees\vantage-leap\elated-hamilton\App.tsx

Read 1,263 lines

Now I have all the context I need. Let me create the new AIReadinessAssessment component.

I'll create it as a separate file and then update App.tsx to use it:

WriteC:\Users\gwin\claude-worktrees\vantage-leap\elated-hamilton\AIReadinessAssessment.tsx

```

/**
 * AI Adoption Readiness Assessment Component
 * Version 2.0 - Deep Dive Assessment
 */

import React, { useState, useEffect } from 'react';
import { ArrowRight, ArrowLeft, CheckCircle2, Mail, AlertTriangle }
from 'lucide-react';
import {
  LeadData,

```

```

    AssessmentResponses,
    AssessmentStep,
    ScoringResult
  } from './assessmentTypes';
import {
  ASSESSMENT_QUESTIONS,
  ASSESSMENT_SECTIONS,
  getVisibleQuestions,
  shouldShowQuestion
} from './assessmentQuestions';
import { calculateAssessmentScore, buildN8NPayload, getReadinessLevel
} from './assessmentScoring';

```

Show full diff (702 more lines)

Limit reached resets 11pm (America/New_York) ·

continue

Update Todos

- ~~Create todo.md plan file with implementation checklist~~
- ~~Create new TypeScript types for v2.0 assessment~~
- ~~Create scoring engine module (assessmentScoring.ts)~~
- ~~Create assessment questions data file~~
- ~~Create AIReadinessAssessment.tsx component with lead capture~~
- Updating App.tsx to use new component
- Test and verify the implementation

Now let me update App.tsx to use the new component. I need to:

1. Import the new AIReadinessAssessment component
2. Remove the old AIReadinessAssessment component
3. Remove the old quiz imports from constants

EditC:\Users\gwinyl\claude-worktrees\vantage-leap\elated-hamilton\App.tsx

```

import React, { useState, useEffect } from 'react';
import { Menu, X, ArrowRight, ArrowUpRight, Mail, MapPin, Phone,
RefreshCw, Bot, Sparkles, Send, ChevronRight, ChevronDown,
CheckCircle2, ShieldCheck, PlayCircle, ArrowLeft, Cpu, Database,
Settings } from 'lucide-react';
import { NAV_ITEMS, HERO_SLIDES, SERVICES, APPROACH_PHASES,
IMPLEMENTATION_PHASES, TRAINING_PHASES, USE_CASES, WHY_US_VALUES,
BARRIERS, TESTIMONIALS, QUIZ_QUESTIONS, QUIZ_RESULTS } from
'./constants';
import { LeadInfo } from './types';
import { NAV_ITEMS, HERO_SLIDES, SERVICES, APPROACH_PHASES,
IMPLEMENTATION_PHASES, TRAINING_PHASES, USE_CASES, WHY_US_VALUES,
BARRIERS, TESTIMONIALS } from './constants';
import { LeadInfo } from './types';
import AIReadinessAssessment from './AIReadinessAssessment';

```

Now let me remove the old AIReadinessAssessment component from App.tsx (lines 762-919):
EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.tsx

```
const AIReadinessAssessment = () => {
  const [step, setStep] = useState<'quiz' | 'result'>('quiz');
  const [hasStarted, setHasStarted] = useState(false);

  const [currentQuestion, setCurrentQuestion] = useState(0);
  const [answers, setAnswers] = useState<{questionId: number, points:
number, text?: string}[]>([]);
  const [qualitativeAnswers, setQualitativeAnswers] = useState<{[key:
number]: string}>({});

  const handleOptionSelect = (points: number) => {
    const newAnswers = [...answers, { questionId:
QUIZ_QUESTIONS[currentQuestion].id, points }];
    setAnswers(newAnswers);
    moveToNext();
  };

  const handleTextSubmit = (e: React.FormEvent) => {
    e.preventDefault();
    // For qualitative questions, points are 0 or handled differently,
    here just stored
    const text =
qualitativeAnswers[QUIZ_QUESTIONS[currentQuestion].id];
    const newAnswers = [...answers, { questionId:
QUIZ_QUESTIONS[currentQuestion].id, points: 0, text }];
    setAnswers(newAnswers);
  };
};
```

Show full diff (140 more lines)

Now let me verify the App.tsx still renders the AIReadinessAssessment component correctly in the main export:

ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.tsx

Read 50 lines

The App.tsx looks correct - it imports and uses the new AIReadinessAssessment component.

Now let me run the build to verify everything compiles correctly:

Update Todos

- ~~Create todo.md plan file with implementation checklist~~
- ~~Create new TypeScript types for v2.0 assessment~~
- ~~Create scoring engine module (assessmentScoring.ts)~~
- ~~Create assessment questions data file~~
- ~~Create AIReadinessAssessment.tsx component with lead capture~~
- ~~Update App.tsx to use new component~~

Testing and verifying the implementation

Bashcd

```
"C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton" &&  
npm run build 2>&1
```

Exit code 1

```
> vantage-leap@0.0.0 build
```

```
> vite build
```

'vite' is not recognized as an internal or external command,
operable program or batch file.

Let me try with npx or install dependencies first:

Bashcd

```
"C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton" &&  
npm install 2>&1
```

```
run `npm fund` for details
```

found 0 vulnerabilities

... +3 lines

Bashcd

```
"C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton" &&  
npx tsc --noEmit 2>&1
```

Exit code 2

```
AIReadinessAssessment.tsx(608,46): error TS2339: Property  
'normalized' does not exist on type 'object'.
```

```
AIReadinessAssessment.tsx(612,26): error TS2339: Property  
'normalized' does not exist on type 'object'.
```

```
assessmentScoring.ts(377,9): error TS2367: This comparison appears to  
be unintentional because the types '"techInfrastructure" |  
"processMaturity" | "dataReadiness" | "budgetResources" |  
"aiKnowledge" | "orgReadiness" | "strategicAlignment"' and  
'"overall"' have no overlap.
```

I need to fix these TypeScript errors. Let me fix the scoring module first:

```
ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\ass  
essmentScoring.ts
```

Read 30 lines

```
EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\asse  
ssmentScoring.ts
```

```
let totalScore = 0;  
for (const [key, dim] of Object.entries(dimensions) as [keyof typeof  
weights, DimensionScore][]) {  
  if (key === 'overall') continue;  
  for (const key of Object.keys(weights) as (keyof typeof weights)[])  
{
```

```

    const dim = dimensions[key];
    dim.weightedContribution = dim.normalized * weights[key];
    totalScore += dim.weightedContribution;
  }

```

Now let me fix the AIReadinessAssessment.tsx error around line 608:

ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\AIR
eadinessAssessment.tsx

Read 40 lines

I need to fix the type issue in the dimension rendering. The problem is that TypeScript doesn't know the type of `dim` when using `Object.entries` on dimensions. Let me fix this:

ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\AIR
eadinessAssessment.tsx

Read 50 lines

EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\AIRe
adinessAssessment.tsx

```

    Your Readiness by Dimension
  </h4>
  <div className="space-y-3">
    {Object.entries(dimensions).map(([key, dim]) => {
      if (key === 'overall' || typeof dim !== 'object') return
null;

      const dimensionNames: Record<string, string> = {
        techInfrastructure: 'Technology Infrastructure',
        processMaturity: 'Process Maturity',
        dataReadiness: 'Data Readiness',
        budgetResources: 'Budget & Resources',
        aiKnowledge: 'AI Knowledge',
        orgReadiness: 'Organizational Readiness',
        strategicAlignment: 'Strategic Alignment'
      };
      return (
        <div key={key} className="flex items-center gap-4">
          <span className="font-medium text-sm w-48
text-pa-black">
            {dimensionNames[key] || key}
            {(['techInfrastructure', 'processMaturity',
'dataReadiness', 'budgetResources', 'aiKnowledge', 'orgReadiness',
'strategicAlignment'] as const).map((key) => {
              const dim = dimensions[key];

```

Show full diff (16 more lines)

Now let me run the TypeScript check again:

Bashcd

```

"C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton" &&
npx tsc --noEmit 2>&1

```


(No content)

TypeScript compiles successfully. Now let me run the development server to verify everything works:

Bash

```
"C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton" &&
npx vite build 2>&1
[2mdist/[22m[32mindex.html          [39m[1m[2m 2.32
kB[22m[1m[22m[2m | gzip: 0.93 kB[22m
[2mdist/[22m[36massets/index-VsLSclrH.js [39m[1m[2m285.32
kB[22m[1m[22m[2m | gzip: 85.12 kB[22m
[32m✓ built in 6.78s[39m
... +7 lines
```

The build is successful. Let me update the todo list and the tasks/todo.md file:

Update Todos

```
Create todo.md plan file with implementation checklist
Create new TypeScript types for v2.0 assessment
Create scoring engine module (assessmentScoring.ts)
Create assessment questions data file
Create AIReadinessAssessment.tsx component with lead capture
Update App.tsx to use new component
Test and verify the implementation
```

ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\tasks\todo.md

Read 72 lines

WriteC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\tasks\todo.md

AI Readiness Assessment v2.0 Redesign

Overview

Redesign the existing AI Adoption Readiness Assessment tool to match the new v2.0 specification from the PRD, implementing all 33 Deep Dive questions with proper scoring and N8N integration.

Requirements

- Lead capture at BEGINNING of assessment (before questions)
- Lead fields: First Name (req), Last Name (req), Phone (req), Email (req), Business Name (req), Website URL (opt)
- 33 Deep Dive questions across 7 sections
- 7 scoring dimensions with weighted scoring
- 5 readiness levels
- Cross-dimensional validation rules
- N8N webhook payload submission

Implementation Checklist

1. Types & Interfaces

- [x] Create assessment-specific types in `assessmentTypes.ts`
 - LeadData interface
 - Question types (single_select, multi_select, dropdown, scale, free_text)

Show full diff (97 more lines)

The AI Readiness Assessment v2.0 redesign is complete. Here's a summary of what was implemented:

Summary

I've completely redesigned the AI Adoption Readiness Assessment tool based on the PRD v2.0 specification. Here's what was created:

New Files Created:

1. `assessmentTypes.ts` - TypeScript type definitions for the entire system
2. `assessmentScoring.ts` - Complete scoring engine with 7 dimension calculators, validation rules, and N8N payload builder
3. `assessmentQuestions.ts` - All 33 Deep Dive questions across 7 sections
4. `AIReadinessAssessment.tsx` - Main component with 3-step flow

Key Features:

- Lead capture at the BEGINNING - Users must enter First Name, Last Name, Phone, Email, Business Name (all required), and Website URL (optional) before starting
- 33 Deep Dive questions across 7 sections (Business Profile, Technology Infrastructure, Business Processes, Budget & Resources, AI Knowledge, Data & Compliance, Goals & Timeline)
- 5 question types: single_select, multi_select (with max selection limits), dropdown, scale (1-5), and free_text
- 7 scoring dimensions with weighted scoring (Tech Infrastructure 20%, Process Maturity 15%, Data Readiness 15%, Budget & Resources 15%, AI Knowledge 15%, Org Readiness 10%, Strategic Alignment 10%)
- 5 readiness levels: Foundation Building, Early Stage, Developing, Ready, Advanced
- 6 cross-dimensional validation flags for edge cases (EXPECTATION_MANAGEMENT, MANAGED_SERVICES_CANDIDATE, etc.)
- Results page with dimension breakdown visualization and key insights
- N8N webhook integration for payload submission

Environment Variable:

You'll need to set `N8N_ASSESSMENT_WEBHOOK` in your environment to enable webhook submissions.

The build compiles successfully with no TypeScript errors.

help me preview website on port 3001

Bash

```
cd
"C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton" &&
npx vite --port 3001 2>&1
(No content)
```

No response requested.

why is this taking so long?

Let me check the dev server status:

Bash

```
cd
"C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton" &&
npx vite --port 3001 2>&1 &
Command running in background with ID: 9303ee
```

BashOutput

BashOutput

The server is running. Ports 3001 and 3002 were already in use, so it started on port 3003.

Open: <http://localhost:3003/>

1. Add "Meta-Prompting" beneath "Context Engineering" in the "AI Training Workshops" service card
2. Move "Technology training for staff" from "AI Consulting & Advisory Services" to "AI Training Workshops".
3. Remove the border around the "Our Approach" content (title and phase cards"
4. Increase the font size in the sliders that is 14px to 16px

I'll make these changes to the constants and App.tsx files.

```
ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\constants.tsx
```

Read 659 lines

Now I'll make the changes to constants.tsx:

```
EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\constants.tsx
```

```
details: [
  'AI adoption suitability assessments',
  'AI adoption roadmap design & implementation',
  'Efficiency, opportunity & systems audits',
  'Technology training for staff'
  'Efficiency, opportunity & systems audits'
icon: <ScanSearch className="w-8 h-8" />,
color: 'bg-pa-peach'
},
```

```
EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\constants.tsx
```

```
description: 'Upskill your teams with hands-on
training/workshops.',
```

```

details: [
  'Prompt engineering',
  'Context engineering',
  'Context engineering',
  'Meta-Prompting',
  'Technology training for staff'
]
icon: <GraduationCap className="w-8 h-8" />,
color: 'bg-pa-sage'
}

```

Now let me fix the "Our Approach" border and the slider font size in App.tsx:

ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.tsx

Read 50 lines

EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.tsx

```

        /* Approach Section - appears OUTSIDE grid, below all
service cards */
        {expandedService && (
          <div className="mt-8 animate-fade-in">
            <div className="bg-white p-6 md:p-8 border-2
border-pa-black rounded-xl shadow-[8px_8px_0px_0px_#1a1a1a]">
              <div className="bg-white p-6 md:p-8">
                </div>
              </div>
            </div>
          )}

```

Now let me fix the slider font size (14px to 16px). Let me find the slider text:

ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.tsx

Read 30 lines

I see the slider text at line 108 already has `style={{ fontSize: '16px' }}`. That looks correct. But you mentioned "that is 14px" - let me search for any 14px references:

Grep14px

Those 14px are in the WhoWeAre section, not the slider. Let me check the index.html ones:

ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\index.html

Read 20 lines

I see - the global CSS is setting 14px as the base font size. But you specifically mentioned "the sliders" text that is 14px. The hero slider subheadline already has `fontSize: '16px'` inline style overriding the global 14px.

Looking at the slider again, the subheadline is at 16px via inline style. Are you referring to a different text in the slider? Let me check if the Tailwind classes are overriding to 14px somewhere. The `text-sm` class in Tailwind is 14px, `text-base` is 16px. The slider subheadline uses `text-lg md:text-xl` with an inline `fontSize: '16px'` override.

Could you clarify which specific text in the slider you want changed to 16px? The current setup seems to have:

- **Headline:** text-3xl md:text-4xl
- **Subheadline:** text-lg md:text-xl with style={{ fontSize: '16px' }}

If you want me to update the global base font from 14px to 16px in index.html, I can do that. Or if there's specific slider text that's still showing at 14px, please let me know which one.

i made a mistake, make the subtext font size in the hero slider 18px, weight 600, line height 24

```
EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.  
tsx
```

```
<p className="font-sans text-lg md:text-xl text-pa-black max-w-lg  
mb-10 leading-relaxed font-medium" style={{ fontSize: '16px' }}>  
<p className="font-sans text-lg md:text-xl text-pa-black max-w-lg  
mb-10" style={{ fontSize: '18px', fontWeight: 600, lineHeight: '24px'  
}}>
```

Done. The hero slider subtext is now 18px font size, 600 weight, and 24px line height.

im not seeing changes on the website

The dev server should hot reload automatically. Let me check if it's still running:

BashOutput

```
<tool_use_error>No shell found with ID: 9303ee</tool_use_error>
```

The dev server stopped. Let me restart it:

Bashcd

```
"C:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton" &&  
npx vite --port 3001 2>&1
```

Command running in background with ID: c9daec

BashOutput

BashOutput

Server is running on <http://localhost:3004/> - try refreshing that page. The changes should be visible now.

do the same for the text in the what we do section change from 14px to 18px font size, 600 weight, and 24px line height.

```
ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App  
.tsx
```

Read 30 lines

```
EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.  
tsx
```

```
<p className="text-sm text-pa-black font-medium font-sans mb-6"  
style={{ lineHeight: '20px', fontSize: '14px', fontWeight: 500,  
color: 'rgb(26, 26, 26)' }}>
```

Every business has hidden and costly inefficiencies - manual processes that drain time, disconnected systems that create errors, and repetitive tasks that

burn out your best people. We identify those inefficiencies and turn them into revenue optimization opportunities.

</p>

```
<p className="text-sm text-pa-black font-medium font-sans mb-8" style={{ lineHeight: '20px', fontSize: '14px', fontWeight: 500, color: 'rgb(26, 26, 26)' }}>
```

```
<p className="text-pa-black font-sans mb-6" style={{ lineHeight: '24px', fontSize: '18px', fontWeight: 600 }}>
```

Every business has hidden and costly inefficiencies - manual processes that drain time, disconnected systems that create errors, and repetitive tasks that burn out your best people. We identify those inefficiencies and turn them into revenue optimization opportunities.

</p>

```
<p className="text-pa-black font-sans mb-8" style={{ lineHeight: '24px', fontSize: '18px', fontWeight: 600 }}>
```

Ask our virtual advisor about how adopting AI into your business can potentially help you optimize and grow.

</p>

Done. The "What We Do" section text is now 18px font size, 600 weight, and 24px line height.

Refresh <http://localhost:3004/> to see the changes.

change the weight to 500

EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.tsx

```
<p className="text-pa-black font-sans mb-6" style={{ lineHeight: '24px', fontSize: '18px', fontWeight: 600 }}>
```

Every business has hidden and costly inefficiencies - manual processes that drain time, disconnected systems that create errors, and repetitive tasks that burn out your best people. We identify those inefficiencies and turn them into revenue optimization opportunities.

</p>

```
<p className="text-pa-black font-sans mb-8" style={{ lineHeight: '24px', fontSize: '18px', fontWeight: 600 }}>
```

```
<p className="text-pa-black font-sans mb-6" style={{ lineHeight: '24px', fontSize: '18px', fontWeight: 500 }}>
```

Every business has hidden and costly inefficiencies - manual processes that drain time, disconnected systems that create errors, and repetitive tasks that burn out your best people. We identify those inefficiencies and turn them into revenue optimization opportunities.

```
        </p>
        <p className="text-pa-black font-sans
mb-8" style={{ lineHeight: '24px', fontSize: '18px', fontWeight: 500
}}>
```

Ask our virtual advisor about how adopting AI into your business can potentially help you optimize and grow.

```
</p>
```

Done. Weight changed to 500.

make it 400

```
EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.
tsx
```

```
<p className="text-pa-black font-sans mb-6" style={{ lineHeight:
'24px', fontSize: '18px', fontWeight: 500 }}>
```

Every business has hidden and costly inefficiencies - manual processes that drain time, disconnected systems that create errors, and repetitive tasks that burn out your best people. We identify those inefficiencies and turn them into revenue optimization opportunities.

```
</p>
```

```
<p className="text-pa-black font-sans
mb-8" style={{ lineHeight: '24px', fontSize: '18px', fontWeight: 500
}}>
```

```
<p className="text-pa-black font-sans mb-6" style={{ lineHeight:
'24px', fontSize: '18px', fontWeight: 400 }}>
```

Every business has hidden and costly inefficiencies - manual processes that drain time, disconnected systems that create errors, and repetitive tasks that burn out your best people. We identify those inefficiencies and turn them into revenue optimization opportunities.

```
</p>
```

```
<p className="text-pa-black font-sans
mb-8" style={{ lineHeight: '24px', fontSize: '18px', fontWeight: 400
}}>
```

Ask our virtual advisor about how adopting AI into your business can potentially help you optimize and grow.

```
</p>
```

Done. Weight changed to 400.

make the font size for this "Becoming an AI powered business is fast becoming a default requisite for competitive positioning. We help you find the opportunities that directly impact your revenue trajectory." 16px. Do the same for the other text on the other sliders too

```
EditC:\Users\gwinyl\.claude-worktrees\vantage-leap\elated-hamilton\App.tsx
```

```
<p className="font-sans text-lg md:text-xl text-pa-black max-w-lg mb-10" style={{ fontSize: '18px', fontWeight: 600, lineHeight: '24px' }}>
```

```
<p className="font-sans text-lg md:text-xl text-pa-black max-w-lg mb-10" style={{ fontSize: '16px', fontWeight: 600, lineHeight: '24px' }}>
```

No response requested.

For ONLY the mobile orientation/version in the "Our Services" section, when any of the "How we work" buttons are pressed, the other two services are right below the active/green tab. Please make the inactive/white service tabs disappear on mobile so that the "our approach" sections is below. A "close" button should also appear only on mobile below the last subtab of the service. Click the "close" button should the "our approach section"

```
ReadC:\Users\gwinyl\.claude-worktrees\vantage-leap\elated-hamilton\App.tsx
```

Read 110 lines

```
EditC:\Users\gwinyl\.claude-worktrees\vantage-leap\elated-hamilton\App.tsx
```

```
    {SERVICES.map((service) => (
      <div
        key={service.id}
        className={` ${expandedService === service.id ?
'bg-pa-sage' : 'bg-white'} p-6 md:p-8 border-2 border-pa-black
rounded-xl flex flex-col shadow-[12px_12px_0px_0px_#1a1a1a]
transition-colors`}
      >
        <h3 className="font-display text-2xl font-bold mb-2">{service.title}</h3>
        <p className="font-serif italic text-pa-black mb-4 font-medium">
          {service.description}
        </p>
        <ul className="space-y-2 mb-6">
          {service.details.map((detail, idx) => (
            <li key={idx} className="text-sm font-sans border-l-2 border-pa-black pl-3 py-1 text-pa-black font-medium">
              {detail}
            </li>
          ))}
        </ul>
        <button
          onClick={() => toggleService(service.id)}

```



```
aria-expanded={expandedService === service.id}
className="mt-auto flex items-center
justify-center gap-2 bg-pa-black text-white px-6 py-3 rounded-lg
font-display font-bold text-sm border-2 border-pa-black
hover:bg-pa-peach hover:text-pa-black transition-all"
Show full diff (134 more lines)
```

Done. On mobile:

1. When a "How We Work" button is pressed, the other two service cards are hidden
2. Only the active (green) service card remains visible
3. The "Our Approach" section appears directly below it
4. A "Close" button appears at the bottom of the approach section (mobile only)
5. Clicking "Close" hides the approach section and shows all service cards again

On desktop, everything works as before with all three cards visible.

Match services card border style the testimonial card border style

ReadC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.
.tsx

Read 30 lines

The testimonial cards use `shadow-[4px_4px_0px_0px_rgba(0,0,0,1)]` and the service cards use `shadow-[12px_12px_0px_0px_#1a1a1a]`. I'll update the service cards to match the testimonial style:

EditC:\Users\gwiny\.claude-worktrees\vantage-leap\elated-hamilton\App.
tsx

```
className={` ${expandedService === service.id ? 'bg-pa-sage' :
'bg-white'} ${expandedService && expandedService !== service.id ?
'hidden md:flex' : 'flex'} p-6 md:p-8 border-2 border-pa-black
rounded-xl flex-col shadow-[12px_12px_0px_0px_#1a1a1a]
transition-colors`}
```

```
className={` ${expandedService === service.id ? 'bg-pa-sage' :
'bg-white'} ${expandedService && expandedService !== service.id ?
'hidden md:flex' : 'flex'} p-6 md:p-8 border-2 border-pa-black
rounded-xl flex-col shadow-[4px_4px_0px_0px_rgba(0,0,0,1)]
transition-colors`}
```