

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>FinThera - Feel Better About Your Money</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script src="https://unpkg.com/react@18/umd/react.development.js"></
script>
  <script src="https://unpkg.com/react-dom@18/umd/react-
dom.development.js"></script>
  <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  <style>
    body {
      margin: 0;
      background: linear-gradient(135deg, #000000 0%, #1a0a2e 50%,
#000000 100%);
      min-height: 100vh;
      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
sans-serif;
    }

    .star {
      position: absolute;
      width: 2px;
      height: 2px;
      background: white;
      border-radius: 50%;
      opacity: 0.8;
      animation: twinkle 3s infinite;
    }

    @keyframes twinkle {
      0%, 100% { opacity: 0.3; }
      50% { opacity: 1; }
    }

    .voice-recording {
      animation: pulse 1s infinite;
    }

    @keyframes pulse {
      0% { transform: scale(1); }
```

```

    50% { transform: scale(1.1); }
    100% { transform: scale(1); }
  }

.gradient-text {
  background: linear-gradient(45deg, #14b8a6, #a855f7);
  -webkit-background-clip: text;
  -webkit-text-fill-color: transparent;
  background-clip: text;
}
</style>
</head>
<body>
  <div id="root"></div>

  <script type="text/babel">
    const { useState, useRef, useEffect } = React;

    // Voice Service Implementation
    class VoiceService {
      constructor() {
        this.provider = 'browser';
        this.openaiApiKey = null;
        this.config = {
          openai: {
            model: 'tts-1-hd',
            voice: 'nova',
            speed: 1.15,
            response_format: 'mp3'
          },
          browser: {
            rate: 1.15,
            pitch: 0.95,
            volume: 0.9,
            preferredVoices: [
              'Samantha', 'Alex', 'Victoria', 'Allison', 'Ava', 'Susan',
              'Google US English', 'Microsoft Aria Online', 'Microsoft Jenny
Online'
            ]
          }
        };
      }

      initialize(config = {}) {

```

```

    if (config.openaiApiKey) {
      this.openaiApiKey = config.openaiApiKey;
      this.provider = 'openai';
      console.log('✅ OpenAI TTS initialized - Premium voice active');
    } else {
      this.provider = 'browser';
      console.log('⚠️ Using Browser TTS (fallback) - Add OpenAI key for
premium voice');
    }
  }

  async speak(text, onStart, onEnd, onError) {
    console.log('🎧 Voice Service - Provider: ${this.provider}');

    try {
      switch (this.provider) {
        case 'openai':
          if (!this.openaiApiKey) {
            return await this.speakBrowser(text, onStart, onEnd, onError);
          }
          return await this.speakOpenAI(text, onStart, onEnd, onError);
        default:
          return await this.speakBrowser(text, onStart, onEnd, onError);
      }
    } catch (error) {
      console.error(`❌ ${this.provider} TTS failed, falling back to
browser:`, error);
      return await this.speakBrowser(text, onStart, onEnd, onError);
    }
  }

  async speakOpenAI(text, onStart, onEnd, onError) {
    if (!this.openaiApiKey) {
      throw new Error('OpenAI API key not provided');
    }

    onStart?();

    try {
      const response = await fetch('https://api.openai.com/v1/audio/
speech', {
        method: 'POST',
        headers: {
          'Authorization': `Bearer ${this.openaiApiKey}`,

```

```

        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        model: this.config.openai.model,
        input: text,
        voice: this.config.openai.voice,
        speed: this.config.openai.speed,
        response_format: this.config.openai.response_format
      }),
    });

    if (!response.ok) {
      throw new Error(`OpenAI API error: ${response.status}`);
    }

    const audioBlob = await response.blob();
    const audioUrl = URL.createObjectURL(audioBlob);
    const audio = new Audio(audioUrl);

    audio.onended = () => {
      URL.revokeObjectURL(audioUrl);
      onEnd?.();
    };

    audio.onerror = (error) => {
      URL.revokeObjectURL(audioUrl);
      onError?.(error);
    };

    await audio.play();
    return audio;
  } catch (error) {
    onError?.(error);
    throw error;
  }
}

async speakBrowser(text, onStart, onEnd, onError) {
  if (!window.speechSynthesis) {
    throw new Error('Browser speech synthesis not supported');
  }

  return new Promise((resolve) => {

```

```

const utterance = new SpeechSynthesisUtterance();

utterance.text = this.addNaturalPauses(text);
utterance.rate = this.config.browser.rate;
utterance.pitch = this.config.browser.pitch;
utterance.volume = this.config.browser.volume;

const voices = speechSynthesis.getVoices();
const selectedVoice = this.selectBestVoice(voices);
if (selectedVoice) {
    utterance.voice = selectedVoice;
}

utterance.onstart = () => onStart?.();
utterance.onend = () => {
    onEnd?.();
    resolve(utterance);
};
utterance.onerror = (error) => {
    onError?.(error);
    resolve(utterance);
};

speechSynthesis.speak(utterance);
});
}

selectBestVoice(voices) {
    for (const voiceName of this.config.browser.preferredVoices) {
        const voice = voices.find(v =>
            v.name.includes(voiceName) && v.lang.startsWith('en')
        );
        if (voice) return voice;
    }
    return voices.find(v => v.lang.startsWith('en')) || voices[0];
}

addNaturalPauses(text) {
    return text
        .replace(/\\.\\.\\.\\/g, '... ')
        .replace(/!/g, '! ')
        .replace(/\\/g, '? ')
        .replace(/:/g, ': ')
        .replace(/,/g, ', ');
}

```

```

    }

    stop() {
      if (this.provider === 'browser' && window.speechSynthesis) {
        speechSynthesis.cancel();
      }
    }
  }

  switchProvider(provider, config = {}) {
    this.provider = provider;
    if (config.openaiApiKey) this.openaiApiKey = config.openaiApiKey;
  }

  updateConfig(provider, newConfig) {
    this.config[provider] = { ...this.config[provider], ...newConfig };
  }
}

const voiceService = new VoiceService();

// Main FinThera Component
const FinancialTherapyPlatform = () => {
  const [currentPage, setCurrentPage] = useState('landing');
  const [userJourney, setUserJourney] = useState({
    hasCompletedConversation: false,
    hasSetupProfile: false,
    hasCompletedPlanning: false,
    currentStep: 'landing'
  });

  const [chatMessages, setChatMessages] = useState([]);
  const [chatInput, setChatInput] = useState('');
  const [currentConversationStep, setCurrentConversationStep] =
    useState('intro');
  const [conversationResponses, setConversationResponses] =
    useState({});
  const [showConversationResults, setShowConversationResults] =
    useState(false);
  const [personalizedInsights, setPersonalizedInsights] = useState([]);

  const [isRecording, setIsRecording] = useState(false);
  const [isAISpeaking, setIsAISpeaking] = useState(false);
  const [voiceEnabled, setVoiceEnabled] = useState(true);
  const [speechRecognition, setSpeechRecognition] = useState(null);

```

```

const [voiceProvider, setVoiceProvider] = useState('browser');
const [apiKey, setApiKey] = useState('');
const [selectedVoice, setSelectedVoice] = useState('nova');
const recognitionRef = useRef(null);
const currentAudioRef = useRef(null);

const [wellnessRings, setWellnessRings] = useState({
  safety: { progress: 45, goal: 100, color: 'from-teal-400 to-teal-600' },
  freedom: { progress: 25, goal: 100, color: 'from-white to-gray-300' },
  fulfillment: { progress: 70, goal: 100, color: 'from-purple-400 to-
purple-600' }
});
const [showReflectionPrompt, setShowReflectionPrompt] =
useState(false);
const [dailyReflection, setDailyReflection] = useState('');

const [mobileMenuOpen, setMobileMenuOpen] = useState(false);

const [financialProfile, setFinancialProfile] = useState({
  age: 25,
  currentIncome: 60000,
  monthlyExpenses: 4500,
  emergencyFund: 5000,
  investments: 15000,
  retirement401k: 8000,
  debt: {
    student: 25000,
    credit: 3000,
    car: 12000,
    mortgage: 0,
    other: 0
  },
  riskTolerance: 'medium',
  financialGoals: {
    buyHome: true,
    travel: true,
    earlyRetirement: false,
    startBusiness: false,
    payOffDebt: true
  },
  lifeValues: {
    coreMotivations: [],
    personalityType: '',
    financialArchetype: '',

```

```

        hiddenFears: [],
        behavioralPatterns: []
    }
});

// Initialize voice conversation
useEffect(() => {
    if (currentPage === 'conversation' && chatMessages.length === 0) {
        setChatMessages([
            {
                type: 'therapist',
                message: "Hey! I'm here to chat about your money stuff - no
judgment, just real talk.",
                timestamp: new Date()
            },
            {
                type: 'therapist',
                message: "Let's start easy - what's something you bought
recently that made you smile?",
                timestamp: new Date()
            }
        ]);
    }
}, [currentPage]);

// Initialize voice functionality
useEffect(() => {
    const initializeVoice = async () => {
        voiceService.initialize({ openaiApiKey: apiKey || null });

        const SpeechRecognition = window.SpeechRecognition ||
window.webkitSpeechRecognition;

        if (SpeechRecognition) {
            try {
                await navigator.mediaDevices.getUserMedia({ audio: true });

                const recognition = new SpeechRecognition();
                recognition.continuous = false;
                recognition.interimResults = true;
                recognition.lang = 'en-US';

                recognition.onstart = () => setIsRecording(true);
                recognition.onresult = (event) => {

```



```

        let finalTranscript = '';
        for (let i = event.resultIndex; i < event.results.length; i++) {
            const transcript = event.results[i][0].transcript;
            if (event.results[i].isFinal) {
                finalTranscript += transcript;
            }
        }
        if (finalTranscript) {
            setChatInput(finalTranscript.trim());
            setIsRecording(false);
        }
    };
    recognition.onerror = () => setIsRecording(false);
    recognition.onend = () => setIsRecording(false);

    recognitionRef.current = recognition;
    setSpeechRecognition(recognition);
  } catch (error) {
    setSpeechRecognition(null);
  }
}

};

initializeVoice();
}, [apiKey]);

// Auto-speak new therapist messages
useEffect(() => {
  if (voiceEnabled && chatMessages.length > 0) {
    const lastMessage = chatMessages[chatMessages.length - 1];
    if (lastMessage.type === 'therapist') {
      speakMessage(lastMessage.message);
    }
  }
}, [chatMessages, voiceEnabled]);

const startRecording = async () => {
  if (!recognitionRef.current) {
    alert('Speech recognition not available. Please try typing your
response.');
```

```

    return;
  }

  try {
```

```

        await navigator.mediaDevices.getUserMedia({ audio: true });
        if (!isRecording) {
            setChatInput('');
            recognitionRef.current.start();
        }
    } catch (error) {
        alert('Unable to access microphone. Please check your browser
settings.');
```

```

        setIsRecording(false);
    }
};
```

```

const stopRecording = () => {
    if (recognitionRef.current && isRecording) {
        recognitionRef.current.stop();
    }
};
```

```

const speakMessage = async (text) => {
    if (!voiceEnabled) return;

    if (currentAudioRef.current) {
        currentAudioRef.current.pause();
        currentAudioRef.current = null;
    }
    voiceService.stop();
```

```

    try {
        const audio = await voiceService.speak(
            text,
            () => setIsASpeaking(true),
            () => setIsASpeaking(false),
            () => setIsASpeaking(false)
        );

        if (audio && audio.pause) {
            currentAudioRef.current = audio;
        }
    } catch (error) {
        setIsASpeaking(false);
    }
};
```

```

const toggleVoice = () => {
```

```

    setVoiceEnabled(!voiceEnabled);
    if (isAISpeaking) {
      if (currentAudioRef.current) {
        currentAudioRef.current.pause();
        currentAudioRef.current = null;
      }
      voiceService.stop();
      setIsAISpeaking(false);
    }
  };

const configureOpenAI = () => {
  const key = prompt('Enter your OpenAI API key:');
  if (key && key.startsWith('sk-')) {
    setApiKey(key);
    setVoiceProvider('openai');
    localStorage.setItem('openai_api_key', key);
    voiceService.switchProvider('openai', { openaiApiKey: key });
  } else {
    alert('Invalid OpenAI API key. It should start with "sk-");
  }
};

const switchVoiceProvider = (provider) => {
  setVoiceProvider(provider);
  if (provider === 'openai' && !apiKey) {
    configureOpenAI();
  } else if (provider === 'browser') {
    voiceService.switchProvider('browser');
  }
};

const changeVoice = (newVoice) => {
  setSelectedVoice(newVoice);
  if (voiceProvider === 'openai') {
    voiceService.updateConfig('openai', { voice: newVoice });
  }
};

// Therapy conversation flow
const therapyFlow = [
  {
    id: 'intro',
    question: "What's something you bought recently that made you

```

```

smile?",
    followUp: "That's cool! Now here's a different question - what's
something you spent money on and immediately regretted it?"
  },
  {
    id: 'guilt_spending',
    question: "What's something you spent money on and immediately
regretted it?",
    followUp: "Ugh, been there! Let's talk about fun stuff - if you had
$200 to spend on yourself this weekend, what would you
do?"
  },
  {
    id: 'ideal_weekend',
    question: "If you had $200 to spend on yourself this weekend, what
would you do?",
    followUp: "Sounds fun! Here's a weird question - do you know
roughly how much you spend in a week? Just ballpark it."
  },
  {
    id: 'weekly_spending',
    question: "Do you know roughly how much you spend in a week?",
    followUp: "No judgment on the number! Now think about this - what's
the biggest money worry on your mind lately?"
  },
  {
    id: 'money_worry',
    question: "What's the biggest money worry on your mind lately?",
    followUp: "I get that. Last question - is there something you really
want but it feels too expensive right now?"
  },
  {
    id: 'big_want',
    question: "Is there something you really want but feels too expensive
right now?",
    followUp: "Alright, here's the interesting part - I think your spending
patterns might actually help you get that. Let me
show you something..."
  }
];

const sendMessage = () => {
  if (!chatInput.trim()) return;

```

```

const newUserMessage = {
  type: 'user',
  message: chatInput,
  timestamp: new Date()
};

setChatMessages(prev => [...prev, newUserMessage]);

const currentStepData = therapyFlow.find(step => step.id ===
currentConversationStep);
if (currentStepData) {
  setConversationResponses(prev => ({
    ...prev,
    [currentConversationStep]: chatInput
  }));
}

setTimeout(() => {
  const stepIndex = therapyFlow.findIndex(step => step.id ===
currentConversationStep);

  if (stepIndex !== -1 && stepIndex < therapyFlow.length - 1) {
    const currentStepData = therapyFlow[stepIndex];
    const therapistResponse = {
      type: 'therapist',
      message: currentStepData.followUp,
      timestamp: new Date()
    };

    setChatMessages(prev => [...prev, therapistResponse]);
    setCurrentConversationStep(therapyFlow[stepIndex + 1].id);
  } else {
    const finalResponse = {
      type: 'therapist',
      message: "Alright, this is getting interesting! Let me crunch
these numbers and show you what I found...",
      timestamp: new Date()
    };

    setChatMessages(prev => [...prev, finalResponse]);

    setTimeout(() => {
      generateLifestyleAnalysis();
    });
  }
}

```

```

        setShowConversationResults(true);
        setUserJourney(prev => ({
            ...prev,
            hasCompletedConversation: true,
            currentStep: 'profile'
        }));
    }, 2000);
}
}, 1500);

setChatInput('');
};

// Analysis functions
const getFinancialArchetype = (conversationText) => {
    const text = conversationText.toLowerCase();

    if ((text.includes('save') || text.includes('independence')) &&
        (text.includes('freedom') || text.includes('security'))) {
        return {
            category: "🎯 Your Financial Archetype",
            title: "💎 The Wealth Builder",
            description: "You're not just saving money—you're architecting financial freedom. You see money as a tool for ultimate independence.",
            deepInsight: "Your drive for financial control could be masking a fear of being dependent on others.",
            actionItems: [
                "Set up automatic investing so wealth building happens without constant decisions",
                "Create a 'joy budget' - allocate money for present happiness without guilt",
                "Track your 'enough number' - what amount would make you feel truly secure?"
            ]
        };
    }

    if (text.includes('experience') || text.includes('travel') ||
        text.includes('adventure') || text.includes('memories')) {
        return {
            category: "🎯 Your Financial Archetype",
            title: "✨ The Experience Collector",
            description: "Money, to you, is energy that creates memories and

```

```

connections. You understand that life is happening now,
  not someday.",
    deepInsight: "Your 'spending' on experiences is actually
sophisticated emotional investing.",
    actionItems: [
      "Create an 'Adventure Fund' that makes saving feel exciting",
      "Use travel rewards credit cards to turn spending into future
experiences",
      "Set up automatic investing in index funds - 'set it and forget it'
while you live"
    ]
  };
}

return {
  category: "🎯 Your Financial Archetype",
  title: "🌟 The Balanced Seeker",
  description: "You're still discovering your relationship with money,
and that's actually a strength. You have the flexibility
  to create a financial life that truly fits you.",
  deepInsight: "Your uncertainty about money might actually be
wisdom—you're sensing that generic advice doesn't fit
  everyone.",
  actionItems: [
    "Start with basic automation - emergency fund and retirement
savings",
    "Experiment with small financial decisions to learn your
preferences",
    "Set one specific financial goal to build confidence"
  ]
};

const getHiddenMotivations = (conversationText) => {
  const text = conversationText.toLowerCase();
  const fears = [];
  const desires = [];

  if (text.includes('worry') || text.includes('stress') ||
text.includes('behind')) {
    fears.push("Financial vulnerability and dependence");
    desires.push("Complete control over life circumstances");
  }
}

```

```

return {
  category: "🔍 Hidden Psychology",
  title: "🧠 What Really Drives Your Money Decisions",
  description: "Your financial behavior isn't just about money—it's
about deeper emotional needs.",
  deepInsight: `You're likely motivated by: ${desires.join(', ')} ||
'personal growth and security'}. But underneath, you might
be avoiding: ${fears.join(', ')} || 'loss of control and independence'.`,
  actionItems: [
    "Before major financial decisions, ask: 'What am I really trying to
achieve?'",
    "Notice when you feel anxious about money and explore the
deeper fear",
    "Create goals that align with your core values, not external
expectations"
  ]
};
};

```

```

const getBehavioralPatterns = (conversationText) => {
  const text = conversationText.toLowerCase();
  let pattern = "Adaptive Planner";
  let description = "You take a thoughtful, balanced approach to money
decisions.";

  if (text.includes('plan') || text.includes('strategy') || text.includes('goal'))
{
    pattern = "Strategic Optimizer";
    description = "You approach money like a chess game—always
thinking several moves ahead.";
  } else if (text.includes('experience') || text.includes('friends') ||
text.includes('social')) {
    pattern = "Social Investor";
    description = "You spend money to create connections and
memories. Your ROI is measured in joy and relationships.";
  }

  return {
    category: "⚡ Behavioral Patterns",
    title: `🧠 You're a ${pattern}`,
    description: description,
    deepInsight: `This pattern serves you well but might create blind
spots in other financial areas.`,
  }
}

```



```

        actionItems: [
            "Identify one financial area where your pattern might not serve
you",
            "Seek advice from people with complementary financial patterns",
            "Set up automatic systems for areas where you're not naturally
strong"
        ]
    };
};

```

```

const getPersonalizedInsights = (conversationText) => {
    const insights = [];
    insights.push(getFinancialArchetype(conversationText));
    insights.push(getHiddenMotivations(conversationText));
    insights.push(getBehavioralPatterns(conversationText));
    return insights;
};

```

```

const generateLifestyleAnalysis = () => {
    const conversationValues =
Object.values(conversationResponses).filter(Boolean).join(' ').toLowerCase();
    const lifeValuesInsights = getPersonalizedInsights(conversationValues);
    setPersonalizedInsights(lifeValuesInsights);

```

```

    setFinancialProfile(prev => ({
        ...prev,
        lifeValues: {
            coreMotivations: ['Personal Growth'],
            personalityType: 'Balanced Explorer',
            financialArchetype: lifeValuesInsights.find(i =>
i.category.includes('Archetype'))?.title || 'The Balanced Seeker',
            hiddenFears: ['Uncertainty about the future'],
            behavioralPatterns: ['Thoughtful decision-making']
        }
    }));

```

```

    setWellnessRings(prev => ({
        safety: { ...prev.safety, progress: 60 },
        freedom: { ...prev.freedom, progress: 40 },
        fulfillment: { ...prev.fulfillment, progress: 75 }
    }));
};

```

```

const calculateFinancialImpact = () => {

```

```

const responses = conversationResponses;
const monthlyWaste = parseInt(responses.waste_spending) || 100;
const yearlyWaste = monthlyWaste * 12;
const tenYearInvested = yearlyWaste * 10 * 1.07;

return {
  monthlyWaste,
  yearlyWaste,
  tenYearInvested: Math.round(tenYearInvested)
};
};

// Wellness Rings component
const renderWellnessRings = () => {
  const rings = [
    {
      id: 'safety',
      title: 'Safety',
      subtitle: 'Can I survive if things go wrong?',
      icon: '🚑',
      progress: wellnessRings.safety.progress
    },
    {
      id: 'freedom',
      title: 'Freedom',
      subtitle: 'Am I building the life I want?',
      icon: '🚀',
      progress: wellnessRings.freedom.progress
    },
    {
      id: 'fulfillment',
      title: 'Fulfillment',
      subtitle: 'Does my money align with my values?',
      icon: '💖',
      progress: wellnessRings.fulfillment.progress
    }
  ];

  return (
    <div className="bg-gray-900/50 backdrop-blur-sm rounded-3xl
p-6 border border-gray-700 mb-6">
      <div className="flex justify-between items-center mb-6">
        <div>
          <h2 className="text-2xl font-bold text-white">Your Financial

```

Wellness</h2>

<p className="text-gray-400">Daily rings to track your financial health</p>

</div>

<button

onClick={() => setShowReflectionPrompt(true)}

className="bg-gradient-to-r from-teal-500 to-purple-500

text-white px-4 py-2 rounded-xl font-medium

hover:shadow-lg transition-all"

>

Daily Check-in 🧠

</button>

</div>

<div className="grid grid-cols-3 gap-8">

{rings.map((ring) => (

<div key={ring.id} className="text-center">

<div className="relative w-32 h-32 mx-auto mb-4">

<div className="absolute inset-0 rounded-full border-8

border-gray-700"></div>

<svg className="absolute inset-0 w-32 h-32 transform

-rotate-90">

<circle

cx="64"

cy="64"

r="56"

stroke={ring.id === 'safety' ? '#06b6d4' : ring.id === 'freedom' ? '#10b981' : '#a855f7'}

strokeWidth="8"

fill="none"

strokeLinecap="round"

strokeDasharray={` \${(ring.progress / 100) \* 351.86} 351.86 `}

className="transition-all duration-1000 ease-out"

/>

</svg>

<div className="absolute inset-0 flex flex-col items-center justify-center">

<span className="text-3xl mb-1">{ring.icon}</span>

<span className="text-2xl font-bold text-

white">{ring.progress}%</span>

</div>

```

    </div>

    <h3 className="text-lg font-bold text-white
mb-1">{ring.title}</h3>
    <p className="text-sm text-gray-400">{ring.subtitle}</p>
  </div>
)}}
</div>

{showReflectionPrompt && (
  <div className="fixed inset-0 bg-black/50 backdrop-blur-sm
z-50 flex items-center justify-center p-4">
    <div className="bg-gray-800 rounded-3xl p-8 max-w-md
w-full border border-gray-700">
      <h3 className="text-2xl font-bold text-white mb-4">Daily
Financial Reflection 🧠</h3>
      <p className="text-gray-300 mb-6">How did you feel
about your money choices today?</p>

      <textarea
        value={dailyReflection}
        onChange={(e) => setDailyReflection(e.target.value)}
        placeholder="Share your thoughts..."
        className="w-full p-4 bg-gray-700 rounded-xl text-
white placeholder-gray-400 resize-none h-24 mb-4"
      />

      <div className="flex gap-3">
        <button
          onClick={() => setShowReflectionPrompt(false)}
          className="flex-1 bg-gray-600 text-white px-4 py-3
rounded-xl font-medium hover:bg-gray-500
transition-all"
        >
          Skip Today
        </button>
        <button
          onClick={() => {
            setWellnessRings(prev => ({
              ...prev,
              fulfillment: { ...prev.fulfillment, progress:
Math.min(100, prev.fulfillment.progress + 5) }
            }));
            setShowReflectionPrompt(false);

```

```

        setDailyReflection('');
      }}
      className="flex-1 bg-gradient-to-r from-teal-500 to-
purple-500 text-white px-4 py-3 rounded-xl
font-medium hover:shadow-lg transition-all"
    >
      Complete Check-in ✨
    </button>
  </div>
</div>
</div>
  )}
</div>
);
};

```

```

// Navigation component
const renderNavigation = () => (
  <nav className="bg-black/80 backdrop-blur-md border-b border-
purple-900/30 sticky top-0 z-50">
    <div className="max-w-7xl mx-auto px-6">
      <div className="flex justify-between items-center h-16">
        <button
          onClick={() => setCurrentPage('landing')}
          className="flex items-center gap-3"
        >
          <div className="text-2xl font-bold gradient-text">
            🧠💰 FinThera
          </div>
          <div className="hidden sm:block text-gray-400 text-sm">
            Feel Better About Your Financial Lifestyle
          </div>
        </button>

        <div className="hidden md:flex items-center gap-1">
          {[
            { id: 'landing', label: 'Home', icon: '🏠', available: true },
            { id: 'conversation', label: 'Voice Chat', icon: '🎙️', available:
true },
            { id: 'profile', label: 'Profile', icon: '👤', available:
userJourney.hasCompletedConversation },
            { id: 'planning', label: 'Life Plan', icon: '🎯', available:
userJourney.hasSetupProfile }
          ].map(({ id, label, icon, available }) => (

```

```

<button
  key={id}
  onClick={() => available && setCurrentPage(id)}
  disabled={!available}
  className={`px-4 py-2 rounded-lg text-sm font-medium
transition-all ${
  currentPage === id
    ? 'bg-gradient-to-r from-teal-500 to-purple-600
text-white shadow-lg'
    : available
    ? 'text-gray-300 hover:text-white hover:bg-
gray-800'
    : 'text-gray-600 cursor-not-allowed'
  }}
>
  {icon} {label}
</button>
)}}
</div>

```

```

<div className="md:hidden">
  <button
    onClick={() => setMobileMenuOpen(!mobileMenuOpen)}
    className="text-gray-300 hover:text-white"
  >
    {mobileMenuOpen ? 'X' : '≡'}
  </button>
</div>
</div>

```

```

{mobileMenuOpen && (
  <div className="md:hidden bg-gray-800 border-t border-
gray-700 py-4">
    <div className="flex flex-col space-y-2">
      {[
        { id: 'landing', label: 'Home', icon: '🏠', available: true },
        { id: 'conversation', label: 'Voice Chat', icon: '🗣️',
available: true },
        { id: 'profile', label: 'Profile', icon: '👤', available:
userJourney.hasCompletedConversation },
        { id: 'planning', label: 'Life Plan', icon: '🎯', available:
userJourney.hasSetupProfile }
      ].map(({ id, label, icon, available }) => (
        <button

```

```

        key={id}
        onClick={() => {
          if (available) {
            setCurrentPage(id);
            setMobileMenuOpen(false);
          }
        }}
        disabled={!available}
        className={`px-4 py-2 rounded-lg text-sm font-
medium transition-all text-left ${
          currentPage === id
            ? 'bg-gradient-to-r from-teal-500 to-purple-600
text-white'
            : available
            ? 'text-gray-300 hover:text-white hover:bg-
gray-700'
            : 'text-gray-600 cursor-not-allowed'
        }}
      >
        {icon} {label}
      </button>
    )}}
  </div>
</div>
)}
</div>
</nav>
);

// Landing Page
const renderLandingPage = () => (
  <div className="min-h-screen bg-gradient-to-br from-black via-
purple-900/30 to-black text-white relative overflow-hidden">
    <div className="absolute inset-0">
      {[...Array(30)].map((_, i) => (
        <div
          key={i}
          className="absolute w-1 h-1 bg-white rounded-full animate-
pulse"
          style={{
            left: `${Math.random() * 100}%`,
            top: `${Math.random() * 100}%`,
            animationDelay: `${Math.random() * 3}s`
          }}
        >

```

```
    />
  })}
</div>
```

```
<div className="relative z-10 max-w-4xl mx-auto px-6 py-20 text-center">
```

```
  <div className="mb-16">
```

```
    <div className="inline-flex items-center px-6 py-3 rounded-full bg-white/10 backdrop-blur-md border border-white/20 mb-8">
```

```
      <span className="text-2xl mr-3">🧠</span>
```

```
      <span className="text-lg font-medium">Your Personal Financial Therapist</span>
```

```
    </div>
```

```
  <h1 className="text-6xl md:text-8xl font-black mb-8">
```

```
    <span className="block gradient-text">
```

```
      Feel better about
```

```
    </span>
```

```
    <span className="block text-white">
```

```
      your money
```

```
    </span>
```

```
  </h1>
```

```
  <p className="text-2xl text-gray-300 mb-8 max-w-3xl mx-auto">
```

```
    I'm your AI financial therapist. In a <span className="text-cyan-400 font-semibold">5-minute voice conversation</span>,&br/>
```

```
    I'll help you understand how your lifestyle choices impact your financial future.
```

```
  </p>
```

```
  <div className="bg-white/5 backdrop-blur-sm rounded-2xl p-6 max-w-2xl mx-auto mb-12 border border-white/10">
```

```
    <p className="text-lg text-gray-300">
```

```
      <span className="text-purple-400 font-semibold">🎙️</span> Talk naturally.</span><br/>
```

```
      <span className="text-cyan-400 font-semibold">💬 I'll talk back.</span><br/>
```

```
      Understanding your relationship with money through conversation.
```

```
    </p>
```

```
  </div>
```



```

<button
  onClick={() => setCurrentPage('conversation')}
  className="group relative px-12 py-6 bg-gradient-to-r from-
purple-600 via-pink-600 to-cyan-500 rounded-2xl
  font-bold text-2xl hover:shadow-2xl hover:shadow-purple-500/40 transform
hover:scale-105 transition-all duration-300 overflow-hidden"
  >
    <div className="absolute inset-0 bg-gradient-to-r from-
purple-600 via-pink-600 to-cyan-500 opacity-0
group-hover:opacity-100 blur-xl transition-opacity duration-300"></div>
    <span className="relative z-10 flex items-center gap-3">
      <span>Start Voice Conversation</span>
      <span>💬</span>
    </span>
  </button>
</div>

```

```

<div className="grid grid-cols-1 md:grid-cols-3 gap-8">
  {[
    {
      icon: "🗣️",
      title: "Voice Conversation",
      desc: "Talk naturally - I'll listen and respond with my voice
too"
    },
    {
      icon: "💡",
      title: "Eye-Opening Insights",
      desc: "See the real financial impact of your daily choices"
    },
    {
      icon: "🎯",
      title: "Your Money Story",
      desc: "Understand your relationship with money - no
judgment"
    }
  ]}.map((item, i) => (
    <div key={i} className="group bg-gradient-to-br from-
gray-800/50 to-purple-900/30 backdrop-blur-sm rounded-2xl
p-6 border border-gray-700 hover:scale-105 transition-all duration-300">
      <div className="text-4xl mb-4">{item.icon}</div>
      <h3 className="text-xl font-bold text-white
mb-3">{item.title}</h3>

```

```

        <p className="text-gray-300">{item.desc}</p>
      </div>
    )}
  </div>
</div>
</div>
);

// Voice Conversation Page
const renderConversationPage = () => (
  <div className="min-h-screen bg-gradient-to-br from-black via-
purple-900/30 to-black">
    {showConversationResults ? renderConversationResults() : (
      <div className="max-w-4xl mx-auto p-6">
        <div className="text-center mb-8">
          <h1 className="text-3xl font-bold text-white mb-2">
            🗨️ Financial Therapy Session
          </h1>
          <p className="text-gray-300 mb-4">
            A safe space to explore your relationship with money
          </p>

          <div className="flex justify-center gap-4 flex-wrap">
            {speechRecognition && (
              <div className="inline-flex items-center px-3 py-1
rounded-full bg-green-500/20 border
border-green-500/30 text-green-300 text-sm">
                🎤 Voice conversation ready
              </div>
            )}

            <div className={`inline-flex items-center px-3 py-1
rounded-full text-sm border ${
              voiceProvider === 'openai' && apiKey
                ? 'bg-green-500/20 border-green-500/30 text-
green-300 animate-pulse'
                : 'bg-gray-500/20 border-gray-500/30 text-gray-300'
            }`} >
              🎧 {voiceProvider === 'openai' && apiKey ? '✅ OpenAI
HD Voice Active' : '🖥️ Browser Voice (Default)'}
            </div>

            <div className="inline-flex items-center gap-2">
              <select

```

```

        value={voiceProvider}
        onChange={(e) => switchVoiceProvider(e.target.value)}
        className="bg-gray-700 text-white text-xs rounded
px-2 py-1 border border-gray-600"
      >
        <option value="browser">Browser Voice</option>
        <option value="openai">OpenAI TTS</option>
      </select>

      {voiceProvider === 'openai' && !apiKey && (
        <button
          onClick={configureOpenAI}
          className="text-xs bg-blue-500 hover:bg-
blue-600 text-white px-2 py-1 rounded"
        >
          Setup API
        </button>
      )}

      {voiceProvider === 'openai' && apiKey && (
        <select
          value={selectedVoice}
          onChange={(e) => changeVoice(e.target.value)}
          className="bg-gray-700 text-white text-xs
rounded px-2 py-1 border border-gray-600"
        >
          <option value="nova">🌟 Nova (Warm)</option>
          <option value="alloy">⚖️ Alloy (Balanced)</option>
          <option value="echo">📢 Echo (Clear)</option>
          <option value="fable">😊 Fable (Friendly)</option>
          <option value="onyx">👁️ Onyx (Deep)</option>
          <option value="shimmer">✨ Shimmer (Bright)</
option>
        </select>
      )}
    </div>
  </div>
</div>
</div>

<div className="bg-gray-800/50 backdrop-blur-sm
rounded-3xl border border-gray-700 h-[600px] flex flex-col">
  <div className="flex-1 overflow-y-auto p-6 space-y-4">
    {chatMessages.map((msg, index) => (
      <div key={index} className={`flex ${msg.type ===

```

```

'user' ? 'justify-end' : 'justify-start'}}`>
      <div className={`max-w-xs lg:max-w-md px-4 py-3
rounded-2xl ${
      msg.type === 'user'
      ? 'bg-gradient-to-r from-teal-500 to-purple-500
text-white'
      : 'bg-gray-700 text-gray-100'
    }}`>
    {msg.type === 'therapist' && (
      <div className="flex items-center gap-2 mb-2">
        <span className="text-lg">🧠</span>
        <span className="text-xs text-gray-300 font-
medium">Your Financial Therapist</span>
        {isAISpeaking && index ===
chatMessages.length - 1 && (
          <div className="flex items-center gap-1">
            <div className="w-1 h-3 bg-green-400
rounded animate-pulse"></div>
            <div className="w-1 h-4 bg-green-400
rounded animate-pulse"
style={{animationDelay: '0.2s'}}></div>
            <div className="w-1 h-3 bg-green-400
rounded animate-pulse"
style={{animationDelay: '0.4s'}}></div>
          </div>
        )}
      </div>
    )}
    <p className="text-sm leading-
relaxed">{msg.message}</p>
  </div>
</div>
)}}
</div>

<div className="p-6 border-t border-gray-600">
  <div className="flex justify-between items-center mb-4">
    <div className="flex items-center gap-3">
      <button
        onClick={toggleVoice}
        className={`p-2 rounded-lg transition-all ${
          voiceEnabled
            ? 'bg-green-500/20 text-green-400 hover:bg-
green-500/30'

```

```

        : 'bg-gray-600/50 text-gray-400 hover:bg-
gray-600/70'
    }}
    title={voiceEnabled ? 'Voice enabled' : 'Voice
disabled'}
  >
    {voiceEnabled ? '🔊' : '🔇'}
  </button>

  {voiceEnabled && (
    <span className="text-xs text-gray-400">
      {isAISpeaking ? '🗣️ AI speaking...' : 'Voice
enabled'}
    </span>
  )}
</div>
</div>

<div className="flex gap-3">
  <input
    type="text"
    value={chatInput}
    onChange={(e) => setChatInput(e.target.value)}
    onKeyDown={(e) => e.key === 'Enter' && !isRecording
&& sendMessage()}
    placeholder={isRecording ? "🎤 Listening... speak
now" : speechRecognition ? "Type or click mic to
speak..." : "Type your response..."}
    className={`flex-1 bg-gray-700 text-white rounded-
xl px-4 py-3 border transition-all
focus:outline-none ${
      isRecording
        ? 'border-red-500 bg-red-900/20'
        : 'border-gray-600 focus:border-purple-500'
    }}
    disabled={isRecording}
  />

  {speechRecognition && (
    <button
      onClick={isRecording ? stopRecording :
startRecording}
      className={`p-3 rounded-xl transition-all flex
items-center justify-center min-w-[48px] ${

```

```

        isRecording
        ? 'bg-red-500 text-white voice-recording
shadow-lg shadow-red-500/50'
        : 'bg-blue-500 hover:bg-blue-600 text-white
hover:shadow-lg'
    }}
    title={isRecording ? 'Click to stop recording' : 'Click
to start voice input'}
  >
    {isRecording ? (
      <div className="flex items-center gap-1">
        <div className="w-2 h-2 bg-white rounded-
full animate-bounce"></div>
        <div className="w-2 h-2 bg-white rounded-
full animate-bounce" style={{animationDelay:
'0.1s'}}></div>
        <div className="w-2 h-2 bg-white rounded-
full animate-bounce" style={{animationDelay:
'0.2s'}}></div>
      </div>
    ) : (
      '🎤'
    )}
  </button>
)}

<button
  onClick={sendMessage}
  disabled={!chatInput.trim() || isRecording}
  className="bg-gradient-to-r from-teal-500 to-
purple-500 text-white p-3 rounded-xl hover:shadow-lg
transition-all disabled:opacity-50 disabled:cursor-not-allowed min-w-[48px]"
  title="Send message"
>
  📤
</button>
</div>
</div>
</div>
</div>
)}
</div>
);

```

```

// Conversation Results Page
const renderConversationResults = () => {
  const impact = calculateFinancialImpact();

  return (
    <div className="max-w-4xl mx-auto p-6 text-white">
      <div className="text-center mb-12">
        <h1 className="text-4xl font-bold mb-4 gradient-text">
          ✨ Your Financial Therapy Results
        </h1>
        <p className="text-xl text-gray-300">
          Here's what your lifestyle choices reveal about your financial
future
          </p>
        </div>

        <div className="grid grid-cols-1 md:grid-cols-2 gap-8 mb-12">
          <div className="bg-gradient-to-br from-green-900/50 to-emerald-900/40 backdrop-blur-sm rounded-3xl p-8 border border-green-700/30">
            <div className="flex items-center gap-3 mb-6">
              <span className="text-2xl">💖</span>
              <h3 className="text-2xl font-bold text-white">Your Joy
Analysis</h3>
            </div>

            <div className="space-y-4">
              <div className="bg-green-800/30 rounded-xl p-4">
                <div className="text-green-300 font-medium mb-2">
💖 What Makes You Happy:</div>
                <div className="text-white italic">
                  {conversationResponses.intro || 'Your meaningful purchases'}
                </div>
              </div>

              <div className="bg-red-800/30 rounded-xl p-4">
                <div className="text-red-300 font-medium mb-2">😓
What You Regret:</div>
                <div className="text-white italic">
                  {conversationResponses.guilt_spending || 'Impulse purchases'}
                </div>
              </div>

              <div className="bg-blue-800/30 rounded-xl p-4">
                <div className="text-blue-300 font-medium mb-2">🌟

```

Your Ideal Weekend:</div>

```
      <div className="text-white italic text-sm">"{conversationResponses.ideal_weekend || 'Quality time and experiences'}"</div>
```

```
    </div>
```

```
  </div>
```

```
</div>
```

```
      <div className="bg-gradient-to-br from-purple-900/50 to-pink-900/40 backdrop-blur-sm rounded-3xl p-8 border border-purple-700/30">
```

```
        <div className="flex items-center gap-3 mb-6">
```

```
          <span className="text-2xl">🏠</span>
```

```
          <h3 className="text-2xl font-bold text-white">The Money Story</h3>
```

```
        </div>
```

```
      <div className="space-y-4">
```

```
        <div className="text-center">
```

```
          <div className="text-4xl font-bold text-purple-400 mb-2">
```

```
            ${impact.monthlyWaste}
```

```
          </div>
```

```
          <div className="text-gray-300 text-sm">Monthly redirectable spending</div>
```

```
        </div>
```

```
      <div className="text-center">
```

```
        <div className="text-3xl font-bold text-pink-400 mb-2">
```

```
          ${impact.yearlyWaste.toLocaleString()}
```

```
        </div>
```

```
        <div className="text-gray-300 text-sm">Yearly opportunity</div>
```

```
      </div>
```

```
      <div className="bg-gradient-to-r from-purple-600 to-pink-600 rounded-xl p-4 text-center">
```

```
        <div className="text-2xl font-bold text-white mb-1">
```

```
          ${impact.tenYearInvested.toLocaleString()}
```

```
        </div>
```

```
        <div className="text-purple-100 text-sm">
```

```
          If invested for 10 years at 7% return
```

```
        </div>
```



```
    </div>
  </div>
</div>
</div>
```

```
    <div className="bg-gradient-to-r from-cyan-900/50 to-
blue-900/40 backdrop-blur-sm rounded-3xl p-8 border
border-cyan-700/30 mb-12">
```

```
    <div className="text-center">
      <h3 className="text-2xl font-bold text-white mb-4">🎯
Your Path Forward</h3>
      <div className="bg-cyan-800/30 rounded-xl p-6 mb-6">
        <div className="text-cyan-300 font-medium mb-2">Your
Goal:</div>
        <div className="text-white text-lg
italic">"{conversationResponses.big_want || 'Building a life you truly
love'}"</div>
```

```
    </div>
```

```
    <div className="grid grid-cols-1 md:grid-cols-3 gap-4">
      <div className="bg-gray-800/50 rounded-xl p-4">
        <div className="text-2xl font-bold text-cyan-400">${
{impact.yearlyWaste.toLocaleString()}}</div>
```

```
        <div className="text-sm text-gray-300">Available
yearly by redirecting spending</div>
      </div>
```

```
      <div className="bg-gray-800/50 rounded-xl p-4">
        <div className="text-2xl font-bold text-
green-400">${2,400}</div>
        <div className="text-sm text-gray-300">You already
spend this on joy yearly</div>
```

```
      </div>
      <div className="bg-gray-800/50 rounded-xl p-4">
        <div className="text-2xl font-bold text-purple-400">${
{(impact.yearlyWaste + 2400).toLocaleString()}}</div>
        <div className="text-sm text-gray-300">Total yearly for
goals + joy</div>
```

```
    </div>
  </div>
</div>
</div>
```

```
{personalizedInsights.length > 0 && (
  <div className="mb-12">
```

```

<div className="text-center mb-8">
  <h2 className="text-3xl font-bold text-white mb-4
gradient-text">
     Your Life Values Report
  </h2>
  <p className="text-gray-300">Deep insights into your
financial personality and patterns</p>
</div>

<div className="grid grid-cols-1 lg:grid-cols-2 gap-8">
  {personalizedInsights.map((insight, index) => (
    <div
      key={index}
      className="bg-gradient-to-br from-gray-800/60 to-
gray-900/60 backdrop-blur-sm rounded-3xl p-8 border
border-gray-600/30"
    >
      <div className="mb-6">
        <div className="text-sm text-teal-400 font-
medium mb-2">{insight.category}</div>
        <h3 className="text-2xl font-bold text-white
mb-4">{insight.title}</h3>
        <p className="text-gray-300 leading-relaxed
mb-4">{insight.description}</p>

        {insight.deepInsight && (
          <div className="bg-purple-900/30 rounded-xl
p-4 mb-4 border border-purple-700/30">
            <div className="text-purple-300 font-medium
mb-2 text-sm"> Deep Insight:</div>
            <p className="text-gray-200 text-
sm">{insight.deepInsight}</p>
          </div>
        )}
      </div>
    </div>
  )}
</div>

{insight.actionItems && insight.actionItems.length > 0
&& (
  <div>
    <div className="text-white font-medium mb-3
flex items-center gap-2">
       Action Steps:
    </div>
    <div className="space-y-2">

```

```

=> (
    {insight.actionItems.map((action, actionIndex)
        <div
            key={actionIndex}
            className="flex items-start gap-3 text-
sm text-gray-300 bg-gray-700/30 rounded-lg
p-3"
        >
            <div className="text-teal-400
mt-0.5">•</div>
            <div>{action}</div>
        </div>
    )}}
</div>
</div>
    )}
</div>
    )}}
</div>
</div>
    )}
    <div className="bg-gradient-to-r from-yellow-900/50 to-
orange-900/40 backdrop-blur-sm rounded-3xl p-8 border
border-yellow-700/30 mb-12 text-center">
        <div className="text-4xl mb-4">💡 </div>
        <h3 className="text-2xl font-bold text-white mb-4">Your Key
Realization</h3>
        <p className="text-lg text-gray-300 leading-relaxed">
            You're not broke. You're not behind. You just need to <span
className="text-yellow-400 font-semibold">redirect
money you're already spending</span> from
            things that don't bring joy to things that build your future. The
money is there - it's just going to the wrong
            places.
        </p>
    </div>

    <div className="bg-gray-800/50 backdrop-blur-sm rounded-3xl
p-8 border border-gray-700 text-center">
        <h3 className="text-2xl font-bold text-white mb-6">🚀 Your
Next Steps</h3>

        <div className="grid grid-cols-1 md:grid-cols-3 gap-6 mb-8">

```

```
<div className="bg-gradient-to-br from-green-500/20 to-emerald-500/20 rounded-xl p-6 border border-green-500/30">
  <div className="text-3xl mb-3">✍️ </div>
  <h4 className="font-bold text-green-400 mb-2">Track
for 1 Week</h4>
  <p className="text-sm text-gray-300">Notice what
brings joy vs what you regret</p>
</div>
```

```
<div className="bg-gradient-to-br from-blue-500/20 to-cyan-500/20 rounded-xl p-6 border border-blue-500/30">
  <div className="text-3xl mb-3">✂️ </div>
  <h4 className="font-bold text-blue-400 mb-2">Cut One
Thing</h4>
  <p className="text-sm text-gray-300">Cancel one
subscription or habit that doesn't spark joy</p>
</div>
```

```
<div className="bg-gradient-to-br from-purple-500/20 to-pink-500/20 rounded-xl p-6 border border-purple-500/30">
  <div className="text-3xl mb-3">🎯 </div>
  <h4 className="font-bold text-purple-400
mb-2">Redirect</h4>
  <p className="text-sm text-gray-300">Put that money
toward your goal automatically</p>
</div>
</div>
```

```
<div className="flex gap-4 justify-center">
  <button
    onClick={() => setCurrentPage('profile')}
    className="bg-gradient-to-r from-cyan-500 to-blue-600
text-white px-8 py-4 rounded-xl font-bold text-lg
hover:shadow-lg transform hover:scale-105 transition-all"
  >
    Set Up Your Profile 👤
  </button>
  <button
    onClick={() => {
      setCurrentPage('landing');
      setChatMessages([]);
      setConversationResponses({});
      setCurrentConversationStep('intro');
      setShowConversationResults(false);
    }}
  >
```

```

    }}
    className="bg-gradient-to-r from-teal-500 to-purple-600
text-white px-8 py-4 rounded-xl font-bold text-lg
hover:shadow-lg transform hover:scale-105 transition-all"
    >
        Start Fresh Session 🚀
    </button>
</div>
</div>
</div>
);
};

```

```

// Profile Page
const renderProfilePage = () => (
    <div className="min-h-screen bg-gradient-to-br from-black via-
purple-900/30 to-black text-white p-6">
        <div className="max-w-6xl mx-auto">
            <div className="text-center mb-8">
                <h1 className="text-4xl font-bold mb-3 gradient-text">
                    👤 Your Financial Profile
                </h1>
                <p className="text-xl text-gray-300">
                    Now let's understand your current financial situation
                </p>
            </div>

            {renderWellnessRings()}

            <div className="grid grid-cols-1 lg:grid-cols-2 gap-8">
                <div className="bg-gray-800/50 backdrop-blur-sm
rounded-2xl p-6 border border-gray-700">
                    <h3 className="text-2xl font-bold mb-6 text-white flex
items-center gap-2">
                        <span className="text-2xl"> 🧑 </span>
                        Personal & Income
                    </h3>

                    <div className="space-y-6">
                        <div>
                            <label className="block font-medium mb-2 text-
gray-300">🎂 Age</label>
                            <input
                                type="range"

```

```

        min="18"
        max="70"
        value={financialProfile.age}
        onChange={(e) => setFinancialProfile(prev =>
({ ...prev, age: parseInt(e.target.value) })))}
        className="w-full accent-teal-500"
      />
    <div className="text-center text-teal-400 font-
semibold text-lg mt-2">
      {financialProfile.age} years old
    </div>
  </div>

  <div>
    <label className="block font-medium mb-2 text-
gray-300">💰 Your Annual Income</label>
    <input
      type="range"
      min="25000"
      max="200000"
      step="5000"
      value={financialProfile.currentIncome}
      onChange={(e) => setFinancialProfile(prev =>
({ ...prev, currentIncome: parseInt(e.target.value) })))}
      className="w-full accent-teal-500"
    />
    <div className="text-center text-green-400 font-
semibold text-lg mt-2">
      ${financialProfile.currentIncome.toLocaleString()}
    </div>
  </div>

  <div>
    <label className="block font-medium mb-2 text-
gray-300">💸 Monthly Expenses</label>
    <input
      type="range"
      min="1000"
      max="15000"
      step="100"
      value={financialProfile.monthlyExpenses}
      onChange={(e) => setFinancialProfile(prev =>
({ ...prev, monthlyExpenses: parseInt(e.target.value)
})))}
  </div>

```

```

        className="w-full accent-teal-500"
      />
      <div className="text-center text-orange-400 font-
semibold text-lg mt-2">
        ${financialProfile.monthlyExpenses.toLocaleString()}/
month
      </div>
    </div>
  </div>
</div>

```

```

    <div className="bg-gray-800/50 backdrop-blur-sm
rounded-2xl p-6 border border-gray-700">
      <h3 className="text-2xl font-bold mb-6 text-white flex
items-center gap-2">
        <span className="text-2xl"> 💎 </span>
        Assets & Savings
      </h3>

```

```

    <div className="space-y-6">
      <div>
        <label className="block font-medium mb-2 text-
gray-300"> 🏠 Emergency Fund</label>
        <input
          type="range"
          min="0"
          max="50000"
          step="500"
          value={financialProfile.emergencyFund}
          onChange={(e) => setFinancialProfile(prev =>
({ ...prev, emergencyFund: parseInt(e.target.value) }) )}
          className="w-full accent-teal-500"
        />
        <div className="text-center text-teal-400 font-
semibold text-lg mt-2">
          ${financialProfile.emergencyFund.toLocaleString()}
        </div>
      </div>

```

```

      <div>
        <label className="block font-medium mb-2 text-
gray-300"> 📈 Total Investments</label>
        <input
          type="range"

```

```

        min="0"
        max="100000"
        step="1000"
        value={financialProfile.investments}
        onChange={(e) => setFinancialProfile(prev =>
({ ...prev, investments: parseInt(e.target.value) })))}
        className="w-full accent-teal-500"
      />
      <div className="text-center text-green-400 font-
semibold text-lg mt-2">
        ${financialProfile.investments.toLocaleString()}
      </div>
    </div>

    <div>
      <label className="block font-medium mb-2 text-
gray-300">🎯 Financial Goals</label>
      <div className="grid grid-cols-2 gap-3">
        {[
          { key: 'buyHome', label: '🏠 Buy Home', active:
financialProfile.financialGoals.buyHome },
          { key: 'travel', label: '✈️ Travel', active:
financialProfile.financialGoals.travel },
          { key: 'earlyRetirement', label: '🌴 Early Retirement',
active:
financialProfile.financialGoals.earlyRetirement },
          { key: 'payOffDebt', label: '💳 Pay Off Debt', active:
financialProfile.financialGoals.payOffDebt
        }
      ].map(goal => (
        <button
          key={goal.key}
          onClick={() => setFinancialProfile(prev => ({
            ...prev,
            financialGoals: {
              ...prev.financialGoals,
              [goal.key]: !prev.financialGoals[goal.key]
            }
          })))}
          className={`p-3 rounded-xl text-center
transition-all text-sm ${
            goal.active
              ? 'bg-gradient-to-br from-green-500 to-
teal-600 text-white shadow-lg'

```



```

    }}
  >
    {goal.label}
  </button>
)}}
</div>
</div>
</div>
</div>

```

```

<div className="text-center mt-8">
  <button
    onClick={() => {
      setUserJourney(prev => ({
        ...prev,
        hasSetupProfile: true,
        currentStep: 'planning'
      }));
      setCurrentPage('planning');
    }}
    className="bg-gradient-to-r from-teal-500 to-purple-600
text-white px-8 py-4 rounded-xl font-bold text-lg
hover:shadow-lg transform hover:scale-105 transition-all"
  >
    Continue to Life Planning 🎯
  </button>
</div>
</div>
</div>
);

```

```
// Planning Page
const renderPlanningPage = () => (
  <div className="min-h-screen bg-gradient-to-br from-black via-purple-900/30 to-black text-white p-6">
    <div className="max-w-6xl mx-auto">
      <div className="text-center mb-8">
        <h1 className="text-4xl font-bold mb-3 gradient-text">
          🎯 Life Planner
        </h1>
        <p className="text-xl text-gray-300">Design your ideal
```

lifestyle for each decade</p>  
</div>

{renderWellnessRings()}

<div className="bg-gray-800/50 backdrop-blur-sm rounded-2xl  
p-8 border border-gray-700 text-center">

<h2 className="text-3xl font-bold text-white mb-4"><img alt="rocket icon" data-bbox="780 206 805 225"/> Your  
Financial Future</h2>

<p className="text-xl text-gray-300 mb-8">  
Based on your profile and goals, here's your personalized path  
to financial wellness  
</p>

<div className="grid grid-cols-1 md:grid-cols-3 gap-6 mb-8">  
<div className="bg-green-900/30 rounded-xl p-6 border  
border-green-700/30">

<div className="text-green-400 font-bold text-3xl  
mb-2">

65

</div>

<div className="text-green-300">Projected Retirement  
Age</div>

</div>

<div className="bg-purple-900/30 rounded-xl p-6 border  
border-purple-700/30">

<div className="text-purple-400 font-bold text-3xl  
mb-2">

\$1.2M

</div>

<div className="text-purple-300">Estimated Net Worth at  
Retirement</div>

</div>

<div className="bg-teal-900/30 rounded-xl p-6 border  
border-teal-700/30">

<div className="text-teal-400 font-bold text-3xl mb-2">



</div>

<div className="text-teal-300">On Track for Financial  
Freedom</div>

</div>

</div>

```

<div className="bg-gradient-to-r from-cyan-900/30 to-
blue-900/30 rounded-xl p-8 border border-cyan-700/30">
  <h3 className="text-2xl font-bold text-white mb-4">🎯
Your Recommended Action Plan</h3>
  <div className="grid grid-cols-1 md:grid-cols-2 gap-6 text-
left">
    <div>
      <h4 className="text-lg font-bold text-cyan-400
mb-3">Immediate Steps (This Month)</h4>
      <ul className="space-y-2 text-gray-300">
        <li>• Set up automatic 401k contribution (15% of
income)</li>
        <li>• Open high-yield savings account for emergency
fund</li>
        <li>• Review and optimize monthly subscriptions</li>
        <li>• Start tracking expenses with your wellness
rings</li>
      </ul>
    </div>
    <div>
      <h4 className="text-lg font-bold text-purple-400
mb-3">Long-term Strategy (Next Year)</h4>
      <ul className="space-y-2 text-gray-300">
        <li>• Build emergency fund to $27,000 (6 months
expenses)</li>
        <li>• Increase investments to $25,000</li>
        <li>• Research home buying in your target area</li>
        <li>• Consider side income streams for faster growth</
li>
      </ul>
    </div>
  </div>
</div>
</div>
</div>
</div>
</div>
);

// Main render
return (
  <div className="min-h-screen bg-black relative overflow-hidden">
    <div className="absolute inset-0 bg-gradient-to-b from-gray-900
via-black to-purple-900/20 pointer-events-none"></div>

```

```

        <div className="absolute inset-0 opacity-30 pointer-events-none">
            <div className="absolute top-10 left-10 w-1 h-1 bg-white
rounded-full animate-pulse"></div>
            <div className="absolute top-20 left-32 w-0.5 h-0.5 bg-teal-200
rounded-full animate-pulse" style={{animationDelay:
'1s'}}></div>
            <div className="absolute top-32 left-64 w-1 h-1 bg-purple-200
rounded-full animate-pulse" style={{animationDelay:
'2s'}}></div>
            <div className="absolute top-16 right-20 w-0.5 h-0.5 bg-white
rounded-full animate-pulse" style={{animationDelay:
'0.5s'}}></div>
            <div className="absolute top-40 right-40 w-1 h-1 bg-teal-300
rounded-full animate-pulse" style={{animationDelay:
'1.5s'}}></div>
            <div className="absolute top-24 right-64 w-0.5 h-0.5 bg-
purple-300 rounded-full animate-pulse" style={{animationDelay:
'2.5s'}}></div>
            <div className="absolute bottom-32 left-20 w-1 h-1 bg-white
rounded-full animate-pulse" style={{animationDelay:
'3s'}}></div>
            <div className="absolute bottom-48 left-48 w-0.5 h-0.5 bg-
teal-200 rounded-full animate-pulse" style={{animationDelay:
'0.8s'}}></div>
            <div className="absolute bottom-20 right-32 w-1 h-1 bg-
purple-200 rounded-full animate-pulse" style={{animationDelay:
'1.8s'}}></div>
            <div className="absolute bottom-40 right-16 w-0.5 h-0.5 bg-
white rounded-full animate-pulse" style={{animationDelay:
'2.8s'}}></div>
        </div>
        <div className="relative z-10">
            {renderNavigation()}

            {currentPage === 'landing' && renderLandingPage()}
            {currentPage === 'conversation' && renderConversationPage()}
            {currentPage === 'profile' && renderProfilePage()}
            {currentPage === 'planning' && renderPlanningPage()}
        </div>
    </div>
);
};

// Render the app

```

```
    ReactDOM.render(<FinancialTherapyPlatform />,
document.getElementById('root'));
  </script>
</body>
</html>
```