

```

import React, { useState } from 'react';
import { Edit2, Download, TrendingDown, ChevronDown } from 'lucide-react';

const RealTimeAuditApp = () => {
  const [activeTab, setActiveTab] = useState('overview');
  const [editMode, setEditMode] = useState(false);
  const [expandedCategory, setExpandedCategory] = useState('kitchen');

  const [clientData, setClientData] = useState({
    restaurantName: 'Fine Dine Hotel Restaurant',
    city: 'Noida',
    auditDate: '2025-01-15',
    monthlyRevenue: 1000000,
    covers: 280,
    avgBill: 1200,
    staffCount: 42,
    foodCostPercent: 33,
    laborCostPercent: 28,
    googleRating: 4.0,
    reviewsPerMonth: 14
  });
}

const [tempClientData, setTempClientData] = useState({ ...clientData });

// CHECKLIST WITH REAL-TIME LOSS IMPACT
const [auditChecklist, setAuditChecklist] = useState([
  {
    id: 'kitchen',
    name: 'Chef Kitchen Operations & Portion Control',
    icon: 'chef-hat',
    color: 'from-amber-900/40 to-amber-800/40',
    borderColor: 'border-amber-500/50',
    items: [
      { id: 'portion_standard', question: 'Are standardized portions documented?', answer: 'No', lossPercent: 10, lossBase: 'foodcost' },
      { id: 'portion_tools', question: 'Are portion control utensils in use?', answer: 'No', lossPercent: 3, lossBase: 'foodcost' },
      { id: 'recipe_cards', question: 'Are recipe cards with plating standards available?', answer: 'No', lossPercent: 2, lossBase: 'foodcost' },
      { id: 'portion_audits', question: 'Are weekly portion audits conducted?', answer: 'No', lossPercent: 2, lossBase: 'foodcost' }
    ]
  },
  {
    id: 'storage',
    name: 'Padlock Store Security & Inventory Control',
    icon: 'padlock',
    color: 'from-red-900/40 to-red-800/40',
  }
]);

```

```
borderColor: 'border-red-500/50',
items: [
  { id: 'store_locked', question: 'Is raw material store locked during service?', answer: 'No', lossPercent: 5, lossBase: 'foodcost' },
  { id: 'access_log', question: 'Is there a single access point with logging?', answer: 'No', lossPercent: 3, lossBase: 'foodcost' },
  { id: 'inventory_checks', question: 'Are daily inventory spot checks done?', answer: 'No', lossPercent: 1.5, lossBase: 'foodcost' },
  { id: 'variance_tracking', question: 'Is stock variance tracked monthly?', answer: 'No', lossPercent: 1.5, lossBase: 'foodcost' }
]
},
{
  id: 'wastage',
  name: ' Wastage Management & Tracking',
  icon: '',
  color: 'from-orange-900/40 to-orange-800/40',
  borderColor: 'border-orange-500/50',
  items: [
    { id: 'wastage_register', question: 'Is daily wastage register maintained?', answer: 'No', lossPercent: 2, lossBase: 'foodcost' },
    { id: 'waste_quantified', question: 'Is wastage quantified by category?', answer: 'No', lossPercent: 2, lossBase: 'foodcost' },
    { id: 'root_cause', question: 'Are root cause analysis done for high waste?', answer: 'No', lossPercent: 1.5, lossBase: 'foodcost' },
    { id: 'plate_waste', question: 'Is plate waste measured & tracked?', answer: 'No', lossPercent: 1.5, lossBase: 'foodcost' }
  ]
},
{
  id: 'cash',
  name: ' Cash & Financial Controls',
  icon: '',
  color: 'from-emerald-900/40 to-emerald-800/40',
  borderColor: 'border-emerald-500/50',
  items: [
    { id: 'daily_reconciliation', question: 'Is daily cash reconciliation performed?', answer: 'Partial', lossPercent: 0.3, lossBase: 'revenue' },
    { id: 'two_person', question: 'Is reconciliation done by two people?', answer: 'No', lossPercent: 0.2, lossBase: 'revenue' },
    { id: 'discount_approval', question: 'Do discounts require manager approval?', answer: 'No', lossPercent: 0.5, lossBase: 'revenue' },
    { id: 'unauthorized_discounts', question: 'Are unauthorized discounts observed?', answer: 'Yes', lossPercent: 0.5, lossBase: 'revenue' }
  ]
},
{
  id: 'revenue',
```

```
name: '📊 Revenue Capture & Upselling',
icon: '⬆',
color: 'from-green-900/40 to-green-800/40',
borderColor: 'border-green-500/50',
items: [
  { id: 'addon_compliance', question: 'Do staff request addons with orders (compliance %)?', answer: '20%', lossPercent: 12, lossBase: 'revenue' },
  { id: 'review_request', question: 'Do staff request Google reviews from customers?', answer: 'No', lossPercent: 3, lossBase: 'revenue' },
  { id: 'delivery_optimization', question: 'Are delivery channels optimized (multi-platform)?', answer: 'Partial', lossPercent: 1.5, lossBase: 'revenue' },
  { id: 'premium_items', question: 'Are premium items & positioning offered?', answer: 'No', lossPercent: 3, lossBase: 'revenue' }
]
},
{
id: 'staff',
name: '⌚ Staff Discipline & Attendance',
icon: '🕒',
color: 'from-blue-900/40 to-blue-800/40',
borderColor: 'border-blue-500/50',
items: [
  { id: 'attendance_tracked', question: 'Is attendance tracked digitally?', answer: 'No', lossPercent: 2, lossBase: 'labor' },
  { id: 'late_coming', question: 'How frequent is late coming?', answer: 'Common', lossPercent: 1.5, lossBase: 'labor' },
  { id: 'absenteeism', question: 'What is monthly absenteeism percentage?', answer: '15-20%', lossPercent: 2, lossBase: 'labor' },
  { id: 'policy_enforcement', question: 'Is attendance policy enforced?', answer: 'Weak', lossPercent: 1, lossBase: 'labor' }
]
},
{
id: 'reputation',
name: '⭐ Digital Presence & Reputation',
icon: '⭐',
color: 'from-yellow-900/40 to-yellow-800/40',
borderColor: 'border-yellow-500/50',
items: [
  { id: 'google_rating', question: 'What is current Google rating?', answer: '4.0/5.0', lossPercent: 4, lossBase: 'revenue' },
  { id: 'review_gap', question: 'Reviews per month vs target (45+)?', answer: '14', lossPercent: 3, lossBase: 'revenue' },
  { id: 'reputation_system', question: 'Is reputation management system active?', answer: 'No', lossPercent: 1.5, lossBase: 'revenue' },
  { id: 'platform_presence', question: 'Multi-platform presence (Zomato, Swiggy)?', answer: 'Partial', lossPercent: 1, lossBase: 'revenue' }
]
]
```

```

    }
  ]);

// CALCULATE LOSSES BASED ON CHECKLIST
const calculateLosses = () => {
  const monthlyRev = clientData.monthlyRevenue;
  const foodCost = (monthlyRev * clientData.foodCostPercent) / 100;
  const laborCost = (monthlyRev * clientData.laborCostPercent) / 100;

  const losses = {};
  let totalLoss = 0;

  auditChecklist.forEach(category => {
    let categoryLoss = 0;
    const itemLosses = [];

    category.items.forEach(item => {
      let itemLoss = 0;

      // Map answer to loss multiplier
      let multiplier = 1;
      if (item.answer === 'Yes') multiplier = 0;
      else if (item.answer === 'Partial') multiplier = 0.5;
      else if (item.answer === 'No') multiplier = 1;
      else if (item.answer.includes('%')) {
        const num = parseFloat(item.answer);
        if (!isNaN(num)) {
          multiplier = num / 100;
        }
      }
    });

    // Calculate loss based on base
    if (item.lossBase === 'foodcost') {
      itemLoss = (foodCost * item.lossPercent / 100) * multiplier;
    } else if (item.lossBase === 'revenue') {
      itemLoss = (monthlyRev * item.lossPercent / 100) * multiplier;
    } else if (item.lossBase === 'labor') {
      itemLoss = (laborCost * item.lossPercent / 100) * multiplier;
    }

    categoryLoss += itemLoss;
    itemLosses.push({ question: item.question, loss: itemLoss });
  });

  losses[category.id] = {
    name: category.name,
    icon: category.icon,
    color: category.color,
  }
}

```

```

        borderColor: category.borderColor,
        totalLoss: categoryLoss,
        items: itemLosses
    );
    totalLoss += categoryLoss;
});

return { losses, totalLoss, totalMonthlyRevenue: monthlyRev };
};

const { losses, totalLoss, totalMonthlyRevenue } = calculateLosses();
const leakagePercent = (totalLoss / totalMonthlyRevenue) * 100;

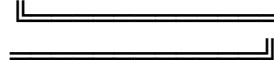
const formatCurrency = (value) => {
    if (value >= 10000000) return `₹${(value / 10000000).toFixed(1)}Cr`;
    if (value >= 100000) return `₹${(value / 100000).toFixed(1)}L`;
    if (value >= 1000) return `₹${(value / 1000).toFixed(1)}K`;
    return `₹${value.toFixed(0)}`;
};

const handleSave = () => {
    setClientData({ ...tempClientData });
    setEditMode(false);
};

const updateChecklistAnswer = (categoryId, itemId, newAnswer) => {
    setAuditChecklist(auditChecklist.map(cat => {
        if (cat.id === categoryId) {
            return {
                ...cat,
                items: cat.items.map(item => {
                    if (item.id === itemId) {
                        return { ...item, answer: newAnswer };
                    }
                    return item;
                })
            };
        }
        return cat;
    }));
};

const generateReport = () => {
    let reportContent =

```



RESTAURANT INFORMATION

Name: \${clientData.restaurantName}
Location: \${clientData.city}
Audit Date: \${clientData.auditDate}
Monthly Revenue: \${formatCurrency(clientData.monthlyRevenue)}
Daily Covers: \${clientData.covers} | Avg Bill: ₹\${clientData.avgBill}
Staff Count: \${clientData.staffCount}

EXECUTIVE SUMMARY

Total Monthly Profit Leakage: \${formatCurrency(totalLoss)}
Percentage of Revenue Lost: \${leakagePercent.toFixed(2)}%
Annual Profit Exposure: \${formatCurrency(totalLoss * 12)}

CATEGORY-BY-CATEGORY LOSS BREAKDOWN

\${Object.entries(losses).map(([key, category]) => ` \${category.name}
Loss Amount: \${formatCurrency(category.totalLoss)} per month
Annual Exposure: \${formatCurrency(category.totalLoss * 12)}

Details:

\${category.items.map(item => ` • \${item.question}: \${formatCurrency(item.loss)}`).join('\n')}`.join('\n')}

RECOVERY POTENTIAL

Conservative (60%): \${formatCurrency(totalLoss * 0.60)} over 90 days
Realistic (85%): \${formatCurrency(totalLoss * 0.85)} over 90 days
Optimistic (100%): \${formatCurrency(totalLoss)} over 90 days

Monthly Recurring Benefit: \${formatCurrency(totalLoss)} after implementation

Generated: \${new Date().toLocaleDateString()}
 Professional Hospitality Consulting | Transforming Restaurants

`;

```

const element = document.createElement('a');
element.setAttribute('href', `data:text/plain;charset=utf-8,${encodeURIComponent(reportContent)}`);
element.setAttribute('download', `${clientData.restaurantName.replace(/\s/g, '_')}_Real_Time_Audit_${clientData.auditDate}.txt`);
element.style.display = 'none';
document.body.appendChild(element);
element.click();
document.body.removeChild(element);
};

return (
<div className="min-h-screen bg-gradient-to-br from-slate-950 via-blue-950 to-slate-950 text-white pb-32">
  {/* Header with Live Loss Counter */}
  <div className="sticky top-0 z-50 bg-gradient-to-r from-slate-900 to-blue-900 border-b border-blue-500/30 p-4 md:p-6">
    <div className="flex justify-between items-start gap-4 mb-4">
      <div className="flex-1">
        <h1 className="text-2xl md:text-3xl font-bold mb-1">_REAL TIME Audit Suite</h1>
        <p className="text-blue-300 text-sm">{clientData.restaurantName} • {clientData.city}</p>
      </div>
      <div className="flex gap-2">
        <button
          onClick={() => setEditMode(!editMode)}
          className="flex items-center gap-2 bg-blue-600 hover:bg-blue-700 px-3 py-2 rounded-lg font-semibold text-sm transition-all">
          <Edit2 className="w-4 h-4" />
          {editMode ? 'Done' : 'Edit'}
        </button>
        <button
          onClick={generateReport}
          className="flex items-center gap-2 bg-green-600 hover:bg-green-700 px-3 py-2 rounded-lg font-semibold text-sm transition-all">
          <Download className="w-4 h-4" />
          Report
        </button>
      </div>
    </div>
  </div>
  {/* LIVE LOSS DISPLAY */}
  <div className="grid grid-cols-1 md:grid-cols-3 gap-3">
    <div className="bg-red-900/40 border border-red-500/50 rounded-lg p-3">
      <p className="text-red-300 text-xs mb-1">TOTAL MONTHLY LOSS</p>
    </div>
  </div>
)

```

```

<p className="text-2xl md:text-3xl font-bold
text-red-300">{formatCurrency(totalLoss)}</p>
<p className="text-red-400 text-xs mt-1">{leakagePercent.toFixed(2)}% of
revenue</p>
</div>
<div className="bg-orange-900/40 border border-orange-500/50 rounded-lg p-3">
  <p className="text-orange-300 text-xs mb-1">↗ Annual Loss</p>
  <p className="text-2xl md:text-3xl font-bold
text-orange-300">{formatCurrency(totalLoss * 12)}</p>
  <p className="text-orange-400 text-xs mt-1">if no changes made</p>
</div>
<div className="bg-green-900/40 border border-green-500/50 rounded-lg p-3">
  <p className="text-green-300 text-xs mb-1">🚀 90-Day Recovery</p>
  <p className="text-2xl md:text-3xl font-bold
text-green-300">{formatCurrency(totalLoss * 0.85)}</p>
  <p className="text-green-400 text-xs mt-1">at 85% execution</p>
</div>
</div>
</div>

/* EDIT MODE */
{editMode && (
  <div className="bg-slate-900/80 border-b border-blue-500/30 p-4 md:p-6 space-y-4
max-h-80 overflow-y-auto">
    <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
      <div>
        <label className="text-slate-400 text-xs block mb-1">Restaurant Name</label>
        <input
          type="text"
          value={tempClientData.restaurantName}
          onChange={(e) => setTempClientData({...tempClientData, restaurantName:
e.target.value})}
          className="w-full bg-slate-700 border border-slate-600 rounded px-3 py-2
text-white text-sm"
        />
      </div>
      <div>
        <label className="text-slate-400 text-xs block mb-1">City</label>
        <input
          type="text"
          value={tempClientData.city}
          onChange={(e) => setTempClientData({...tempClientData, city: e.target.value})}
          className="w-full bg-slate-700 border border-slate-600 rounded px-3 py-2
text-white text-sm"
        />
      </div>
      <div>
        <label className="text-slate-400 text-xs block mb-1">Monthly Revenue (₹)</label>

```

```

<input
  type="number"
  value={tempClientData.monthlyRevenue}
  onChange={(e) => setTempClientData({...tempClientData, monthlyRevenue:
  parseInt(e.target.value) || 0})}
  className="w-full bg-slate-700 border border-slate-600 rounded px-3 py-2
text-white text-sm"
/>
</div>
<div>
  <label className="text-slate-400 text-xs block mb-1">Daily Covers</label>
  <input
    type="number"
    value={tempClientData.covers}
    onChange={(e) => setTempClientData({...tempClientData, covers:
    parseInt(e.target.value) || 0})}
    className="w-full bg-slate-700 border border-slate-600 rounded px-3 py-2
text-white text-sm"
  />
</div>
</div>
<button
  onClick={handleSave}
  className="w-full bg-green-600 hover:bg-green-700 px-4 py-2 rounded-lg
font-semibold transition-all text-sm"
>
  Save & See Live Updates
</button>
</div>
)}

/* Tabs */
<div className="bg-slate-900/50 border-b border-blue-500/20 sticky top-32 md:top-24
z-40 overflow-x-auto">
<div className="flex gap-1 p-2 md:p-4 min-w-max md:min-w-full md:justify-center">
  {[{
    { id: 'overview', label: '📊 Overview' },
    { id: 'checklist', label: '✓ Checklist' },
    { id: 'analysis', label: '📈 Analysis' },
    { id: 'summary', label: '📋 Summary' }
  ].map(tab => (
    <button
      key={tab.id}
      onClick={() => setActiveTab(tab.id)}
      className={`px-3 md:px-4 py-2 rounded-lg font-semibold text-sm
white-space nowrap transition-all ${(
        activeTab === tab.id
        ? 'bg-blue-600 text-white'
      )}`}
    )
  ))}
</div>

```

```

        : 'bg-slate-800 text-slate-300 hover:bg-slate-700'
    `}`}
  >
  {tab.label}
</button>
))>
</div>
</div>

/* Content */
<div className="p-4 md:p-6 max-w-6xl mx-auto">

  /* OVERVIEW TAB */
  {activeTab === 'overview' && (
    <div className="space-y-4">
      <div className="bg-gradient-to-r from-red-900/40 to-red-800/40 border-2
border-red-500/50 rounded-lg p-4 md:p-6">
        <h2 className="text-2xl md:text-3xl font-bold mb-3">💸 Total Profit Loss</h2>
        <p className="text-4xl md:text-5xl font-bold text-red-300
mb-2">{formatCurrency(totalLoss)}</p>
        <p className="text-red-200 text-sm">Monthly loss ({leakagePercent.toFixed(2)}%
of revenue)</p>
        <p className="text-red-300 text-sm mt-2">Annual exposure:
{formatCurrency(totalLoss * 12)}</p>
      </div>

      <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
        <div className="bg-blue-900/30 border border-blue-500/50 rounded-lg p-4
md:p-6">
          <p className="text-blue-300 text-sm mb-2">₹ Monthly Revenue</p>
          <p className="text-3xl font-bold">{formatCurrency(totalMonthlyRevenue)}</p>
          <p className="text-blue-300 text-xs mt-2">{clientData.covers} covers/day @
₹{clientData.avgBill}/bill</p>
        </div>
        <div className="bg-green-900/30 border border-green-500/50 rounded-lg p-4
md:p-6">
          <p className="text-green-300 text-sm mb-2">🚀 90-Day Recovery (85%)</p>
          <p className="text-3xl font-bold text-green-300">{formatCurrency(totalLoss *
0.85)}</p>
          <p className="text-green-300 text-xs mt-2">Realistic execution scenario</p>
        </div>
      </div>

      <div className="grid grid-cols-1 md:grid-cols-3 gap-4">
        <div className="bg-slate-800/50 border border-slate-700 rounded-lg p-4 md:p-6">
          <p className="text-slate-400 text-xs">⭐ Google Rating</p>
          <p className="text-2xl font-bold mt-2">{clientData.googleRating}/5.0</p>
        </div>
      </div>
    </div>
  )
)

```

```

    <p className="text-slate-400 text-xs mt-1">{clientData.reviewsPerMonth}</p>
    reviews/month</p>
    </div>
    <div className="bg-slate-800/50 border border-slate-700 rounded-lg p-4 md:p-6">
        <p className="text-slate-400 text-xs">👤 Team Size</p>
        <p className="text-2xl font-bold mt-2">{clientData.staffCount}</p>
        <p className="text-slate-400 text-xs mt-1">staff members</p>
    </div>
    <div className="bg-slate-800/50 border border-slate-700 rounded-lg p-4 md:p-6">
        <p className="text-slate-400 text-xs">📅 Audit Date</p>
        <p className="text-xl font-bold
mt-2">{clientData.auditDate.split('-').reverse().join('/')}</p>
        <p className="text-slate-400 text-xs mt-1">Format: DD/MM/YYYY</p>
    </div>
</div>

<div className="bg-orange-900/20 border border-orange-500/30 rounded-lg p-4
md:p-6">
    <h3 className="font-bold mb-3">⚠️ Key Takeaways</h3>
    <ul className="space-y-2 text-sm text-slate-300">
        <li>✓ Restaurant is losing <span className="font-bold
text-red-400">{formatCurrency(totalLoss)}/month</span> to operational gaps</li>
        <li>✓ This represents <span className="font-bold
text-red-400">{leakagePercent.toFixed(1)}%</span> of monthly revenue</li>
        <li>✓ Annual impact if not addressed: <span className="font-bold
text-red-400">{formatCurrency(totalLoss * 12)}</span></li>
        <li>✓ Recovery potential in 90 days: <span className="font-bold
text-green-400">{formatCurrency(totalLoss * 0.85)}</span></li>
    </ul>
</div>
</div>
)}

/* CHECKLIST TAB - WITH ALL QUESTIONS VISIBLE */
{activeTab === 'checklist' &&
<div className="space-y-4">
    <div className="bg-blue-900/20 border border-blue-500/30 rounded-lg p-4 mb-4">
        <h2 className="text-lg md:text-xl font-bold mb-2">📋 Interactive Audit
Checklist</h2>
        <p className="text-blue-300 text-sm">Select Yes/No/Partial for each question →
See losses update INSTANTLY above</p>
    </div>
}

{auditChecklist.map((category) => {
    const categoryLoss = losses[category.id].totalLoss;
    const isExpanded = expandedCategory === category.id;

    return (

```

```

        <div key={category.id} className={'bg-gradient-to-r ${category.color} border-2
${category.borderColor} rounded-lg overflow-hidden'>
    /* Category Header */
    <button
        onClick={() => setExpandedCategory(isExpanded ? null : category.id)}
        className="w-full p-4 md:p-5 hover:opacity-80 transition-all text-left flex
justify-between items-center"
    >
        <div className="flex-1">
            <h3 className="text-base md:text-lg font-bold mb-2">{category.name}</h3>
            <p className="text-slate-300 text-xs md:text-sm">{category.items.length}
audit questions</p>
        </div>
        <div className="text-right flex-shrink-0">
            <p className="text-lg md:text-2xl font-bold
text-red-300">{formatCurrency(categoryLoss)}</p>
            <p className="text-slate-400 text-xs">monthly loss</p>
            <ChevronDown className={`w-4 h-4 mt-1 mx-auto transition-transform
${isExpanded ? 'rotate-180' : ''}`}>
        </div>
    </button>

    /* Expanded Items - ALL QUESTIONS VISIBLE */
    {isExpanded && (
        <div className="bg-slate-900/50 border-t-2 divide-y divide-slate-700">
            {category.items.map((item, idx) => {
                const itemLoss = losses[category.id].items[idx].loss;
                return (
                    <div key={item.id} className="p-4 space-y-3">
                        /* Question */
                        <div className="flex justify-between items-start gap-2">
                            <p className="text-sm md:text-base font-semibold
flex-1">{item.question}</p>
                            <div className="text-right flex-shrink-0">
                                <p className="text-red-400 font-bold
text-sm">{formatCurrency(itemLoss)}</p>
                                <p className="text-slate-400 text-xs">loss/month</p>
                            </div>
                        </div>
                    </div>
                )
            )}
            /* Answer Buttons */
            <div className="flex gap-2 flex-wrap">
                {'Yes', 'Partial', 'No'].map(option => (
                    <button
                        key={option}
                        onClick={() => updateChecklistAnswer(category.id, item.id, option)}
                        className={`px-4 py-2 rounded font-semibold transition-all text-sm ${
                            item.answer === option
                        }`}
                    >{option}</button>
                ))
            </div>
        </div>
    )
)

```

```

        ? 'bg-blue-600 text-white ring-2 ring-blue-400'
        : 'bg-slate-700 text-slate-300 hover:bg-slate-600'
    `}`}
    >
    {option}
    </button>
)}
</div>

/* Loss Impact */
<div className="bg-slate-800/50 rounded p-2 text-xs text-slate-300 border border-slate-700">
    <span className="text-slate-400">Current Loss: </span>
    <span className={itemLoss > 0 ? 'text-red-400 font-bold' :
    'text-green-400 font-bold'}>
        {formatCurrency(itemLoss)}/month
    </span>
    {item.answer === 'Yes' && <span className="text-green-400 ml-2">✓
    No Loss!</span>}
    </div>
    </div>
);
`}`}
</div>
`}
</div>
);
`}`}
</div>
`}
</div>
);
`}`}
</div>
`}
</div>
`);

/* ANALYSIS TAB */
{activeTab === 'analysis' && (
    <div className="space-y-4">
        <div className="bg-gradient-to-r from-red-900/40 to-red-800/40 border-2 border-red-500/50 rounded-lg p-4 md:p-6">
            <div className="flex items-center gap-3 mb-3">
                <TrendingDown className="w-8 h-8 text-red-400" />
                <h2 className="text-2xl md:text-3xl font-bold">Total Profit Loss Breakdown</h2>
            </div>
            <p className="text-4xl md:text-5xl font-bold text-red-300 mb-2">{formatCurrency(totalLoss)}</p>
            <p className="text-red-200 text-sm">Monthly loss ({leakagePercent.toFixed(2)}% of revenue)</p>
        </div>
    
```

```

        <div key={key} className={'bg-gradient-to-r ${category.color} border-2
${category.borderColor} rounded-lg p-4 md:p-6'}>
            <div className="flex justify-between items-start mb-3">
                <h3 className="text-lg md:text-xl font-bold flex items-center gap-2">
                    <span>{category.icon}</span>
                    {category.name}
                </h3>
                <div className="text-right">
                    <p className="text-2xl md:text-3xl
font-bold">{formatCurrency(category.totalLoss)}</p>
                    <p className="text-xs opacity-75">per month</p>
                </div>
            </div>

            <div className="space-y-2 mb-3">
                {category.items.map((item, idx) => (
                    <div key={idx} className="flex justify-between items-center p-2
bg-slate-800/50 rounded text-sm">
                        <span className="flex-1">{item.question}</span>
                        <span className="font-bold ml-2">{formatCurrency(item.loss)}</span>
                    </div>
                )))
            </div>

            <div className="p-3 bg-slate-900/50 border-t rounded mt-3">
                <p className="text-xs opacity-75 mb-1">Annual Exposure:</p>
                <p className="font-bold">{formatCurrency(category.totalLoss * 12)}</p>
            </div>
        </div>
    ))}

/* Recovery Summary */
<div className="bg-gradient-to-r from-green-900/40 to-emerald-900/40 border-2
border-green-500/50 rounded-lg p-4 md:p-6">
    <h3 className="text-lg font-bold mb-4">🚀 Recovery Potential (90 Days)</h3>
    <div className="space-y-3">
        <div className="flex justify-between items-center p-3 bg-yellow-900/30 border
border-yellow-500/20 rounded-lg">
            <span className="font-semibold">Conservative (60%)</span>
            <span className="font-bold text-yellow-400">{formatCurrency(totalLoss *
0.60)}</span>
        </div>
        <div className="flex justify-between items-center p-3 bg-green-900/40 border
border-green-500/30 rounded-lg ring-2 ring-green-500/50">
            <span className="font-semibold">Realistic (85%) ✓</span>
            <span className="font-bold text-green-400">{formatCurrency(totalLoss *
0.85)}</span>
        </div>
    </div>

```

```

        <div className="flex justify-between items-center p-3 bg-blue-900/30 border border-blue-500/20 rounded-lg">
            <span className="font-semibold">Optimistic (100%)</span>
            <span className="font-bold text-blue-400">{formatCurrency(totalLoss)}</span>
        </div>
    </div>
</div>
</div>
)}
/* SUMMARY TAB */
{activeTab === 'summary' && (
    <div className="space-y-4">
        <div className="bg-gradient-to-r from-purple-900/40 to-blue-900/40 border-2 border-purple-500/50 rounded-lg p-4 md:p-6">
            <h2 className="text-2xl font-bold mb-4"> Complete Audit Summary</h2>

            <div className="space-y-4">
                /* Restaurant Info */
                <div className="bg-slate-800/50 border border-slate-700 rounded-lg p-4">
                    <h3 className="font-bold text-lg mb-3">Restaurant Details</h3>
                    <div className="grid grid-cols-2 gap-3 text-sm">
                        <div>
                            <p className="text-slate-400">Name</p>
                            <p className="font-bold">{clientData.restaurantName}</p>
                        </div>
                        <div>
                            <p className="text-slate-400">City</p>
                            <p className="font-bold">{clientData.city}</p>
                        </div>
                        <div>
                            <p className="text-slate-400">Monthly Revenue</p>
                            <p className="font-bold">{formatCurrency(totalMonthlyRevenue)}</p>
                        </div>
                        <div>
                            <p className="text-slate-400">Daily Covers</p>
                            <p className="font-bold">{clientData.covers}</p>
                        </div>
                        <div>
                            <p className="text-slate-400">Avg Bill</p>
                            <p className="font-bold">₹{clientData.avgBill}</p>
                        </div>
                        <div>
                            <p className="text-slate-400">Staff</p>
                            <p className="font-bold">{clientData.staffCount}</p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    )
)

```

```

/* Loss Summary */


<h3 className="font-bold text-lg mb-3 text-red-300"> Loss Summary</h3>
  <div className="space-y-2 text-sm">
    {Object.entries(losses).map(([key, category]) => (
      <div key={key} className="flex justify-between items-center p-2 bg-slate-800/50 rounded">
        <span>{category.icon} {category.name.split(' ')[1]}</span>
        <span className="font-bold text-red-400">{formatCurrency(category.totalLoss)}</span>
      </div>
    )))
    <div className="flex justify-between items-center p-3 bg-red-900/40 border border-red-500/30 rounded-lg font-bold mt-2">
      <span>Total Monthly Loss</span>
      <span className="text-lg text-red-300">{formatCurrency(totalLoss)}</span>
    </div>
    <div className="flex justify-between items-center p-2 bg-slate-800/50 rounded">
      <span>Annual Loss</span>
      <span className="font-bold">{formatCurrency(totalLoss * 12)}</span>
    </div>
    <div className="flex justify-between items-center p-2 bg-slate-800/50 rounded">
      <span>% of Revenue</span>
      <span className="font-bold">{leakagePercent.toFixed(2)}%</span>
    </div>
  </div>


```

```

/* Recovery Plan */


<h3 className="font-bold text-lg mb-3 text-green-300"> Recovery Plan (90 Days)</h3>
  <div className="space-y-2 text-sm">
    <div className="flex justify-between items-center p-2 bg-yellow-900/30 rounded">
      <span>Conservative (60%)</span>
      <span className="font-bold text-yellow-400">{formatCurrency(totalLoss * 0.60)}</span>
    </div>
    <div className="flex justify-between items-center p-2 bg-green-900/40 rounded font-bold ring-1 ring-green-500/50">
      <span>Realistic (85%) ✓</span>
      <span className="text-green-400">{formatCurrency(totalLoss * 0.85)}</span>
    </div>
  </div>


```

```

        <div className="flex justify-between items-center p-2 bg-blue-900/30 rounded">
            <span>Optimistic (100%)</span>
            <span className="font-bold text-blue-400">{formatCurrency(totalLoss)}</span>
        </div>
        <div className="mt-3 p-3 bg-slate-800/50 border border-slate-700 rounded">
            <p className="text-xs text-slate-400 mb-1">Monthly Recurring Benefit After Day 90:</p>
            <p className="font-bold text-green-400 text-lg">{formatCurrency(totalLoss)}/month</p>
            <p className="text-xs text-slate-400 mt-1">Annual Recurring: {formatCurrency(totalLoss * 12)}</p>
        </div>
    </div>

    {/* Next Steps */}
    <div className="bg-blue-900/20 border border-blue-500/30 rounded-lg p-4">
        <h3 className="font-bold text-lg mb-3">📋 Next Steps</h3>
        <ul className="space-y-2 text-sm text-slate-300">
            <li>✓ Review this audit report with stakeholders</li>
            <li>✓ Identify priority control implementations (1-2 weeks)</li>
            <li>✓ Train staff on new procedures and standards</li>
            <li>✓ Implement tracking systems and monitoring</li>
            <li>✓ Monitor and track monthly improvements</li>
            <li>✓ Conduct follow-up audit after 90 days</li>
        </ul>
    </div>
    </div>
    </div>
    </div>
}

}
</div>

 {/* Professional Footer */}
<div className="fixed bottom-0 left-0 right-0 bg-gradient-to-r from-slate-900 to-blue-900 border-t border-blue-500/30 p-4 text-center">
    <p className="text-slate-300 text-xs md:text-sm mb-1">
        🎯 <span className="font-semibold">Professional Hospitality Consulting</span>
    </p>
    <p className="text-slate-400 text-xs">
        Transforming Restaurants Into Profit-Driven Businesses | Real-Time Audit Analysis |
    Confidential
    </p>
    <p className="text-slate-500 text-xs mt-1">
        🔒 Data Protection Assured | 🔒 Secure & Professional | 💼 Enterprise-Grade
    Auditing
    </p>

```

```
</p>
</div>
</div>
);
};

export default RealTimeAuditApp;
```