```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Kisan AI - Smart Farming Assistant</title>
<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>
<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css"
rel="stylesheet">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
<link
href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&family=O
pen+Sans:wght@400;600;700&display=swap" rel="stylesheet">
<style>
:root {
    --primary-color: #2F855A;
    --secondary-color: #38A169;
    --accent-color: #4299E1;
    --light-bg: #F7FAFC;
    --dark-text: #1A202C;
    --light-text: #718096;
}

body {
    font-family: 'Open Sans', sans-serif;
    background-color: #F8F9FA;
    color: var(--dark-text);
    line-height: 1.6;
}

.professional-typography {
    font-family: 'Inter', sans-serif;
}

.btn-primary {
    background-color: var(--primary-color);
    color: white;
    transition: all 0.3s ease;
}

.btn-primary:hover {
    background-color: #276749;
    transform: translateY(-2px);
    box-shadow: 0 4px 12px rgba(47, 133, 90, 0.2);
}
```

```
.btn-secondary {
    background-color: var(--accent-color);
    color: white;
    transition: all 0.3s ease;
}

.btn-secondary:hover {
    background-color: #3182CE;
    transform: translateY(-2px);
    box-shadow: 0 4px 12px rgba(66, 153, 225, 0.2);
}

.card {
    background: white;
    border-radius: 12px;
    box-shadow: 0 4px 6px rgba(0, 0, 0, 0.05);
    transition: all 0.3s ease;
}

.card:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1);
}

.feature-card {
    border-left: 4px solid var(--primary-color);
}

.nav-item {
    position: relative;
    transition: all 0.3s ease;
}

.nav-item:hover {
    color: var(--primary-color);
}

.nav-item.active {
    color: var(--primary-color);
    font-weight: 600;
}

.nav-item.active::after {
    content: "";
    position: absolute;
    bottom: -8px;
    left: 0;
```

```
    width: 100%;
    height: 3px;
    background-color: var(--primary-color);
    border-radius: 3px;
}

.hero-gradient {
    background: linear-gradient(135deg, rgba(47, 133, 90, 0.1) 0%, rgba(56, 161, 105, 0.1)
100%);
}

.voice-wave {
    animation: voice-wave 0.8s ease-in-out infinite alternate;
}

@keyframes voice-wave {
    0% { height: 8px; opacity: 0.4; }
    100% { height: 24px; opacity: 1; }
}

.loading-spinner {
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}

.slide-in {
    animation: slide-in 0.6s ease-out forwards;
}

@keyframes slide-in {
    from {
        opacity: 0;
        transform: translateY(20px);
    }
    to {
        opacity: 1;
        transform: translateY(0);
    }
}

.pulse {
    animation: pulse 2s infinite;
}
```

```
@keyframes pulse {
    0%, 100% { opacity: 1;}
    50% { opacity: 0.5;}
}

.input-field {
    transition: all 0.3s ease;
    border: 1px solid #E2E8F0;
}

.input-field:focus {
    border-color: var(--primary-color);
    box-shadow: 0 0 0 3px rgba(47, 133, 90, 0.2);
}

.badge {
    font-size: 0.75rem;
    padding: 0.25rem 0.5rem;
    border-radius: 9999px;
}

.badge-new {
    background-color: #F6E05E;
    color: #744210;
}

.badge-live {
    background-color: #F56565;
    color: white;
}

.badge-beta {
    background-color: #4299E1;
    color: white;
}

.tab-button {
    transition: all 0.3s ease;
    padding: 7px 15px;
    border-radius: 7px
}

.tab-button.active {
    background-color: var(--primary-color);
    color: white;
    border-radius: 7px;
    padding: 7px 15px;
}
```

```
.tab-button:not(.active):hover {
    background-color: #EDF2F7;
}

.progress-bar {
    transition: width 0.6s ease;
}

/* New styles for full-page views */
.full-page-view {
    position: fixed;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background-color: white;
    z-index: 100;
    overflow-y: auto;
    padding: 20px;
}

.back-button {
    display: flex;
    align-items: center;
    color: var(--primary-color);
    margin-bottom: 20px;
    cursor: pointer;
}

.back-button i {
    margin-right: 8px;
}

/* Login page styles */
.login-bg {
    background: linear-gradient(rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)),

url('https://images.unsplash.com/photo-1500382017468-9049fed747ef?ixlib=rb-4.0.3&ixid=M
3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&
w=2070&q=80');
    background-size: cover;
    background-position: center;
    min-height: 100vh;
}

.login-card {
    backdrop-filter: blur(10px);
```

```
    background: rgba(255, 255, 255, 0.85);
}

.feature-icon {
    width: 60px;
    height: 60px;
    border-radius: 16px;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 24px;
    margin-bottom: 16px;
}

/* Enhanced container styles */
.app-container {
    max-width: 1200px;
    margin: 0 auto;
    padding: 20px;
    background-color: rgba(255, 255, 255, 0);
    border-radius: 16px;
    box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);
    transition: all 0.3s ease;
}

.app-container:hover {
    box-shadow: 0 15px 35px rgba(0, 0, 0, 0.15);
}

/* Google button fix */
.google-btn img {
    width: 35px;
    height: 35px;
    margin-right: 8px;
}
.text-gray-500 {
    --tw-text-opacity: 1;
    color: rgb(47 133 90);
    padding-bottom: 7px;
    padding-top: 7px;
    padding-left: 15px;
    padding-right: 15px;
    border-radius: 7px;
}

/* Alert section scrolling */
.alert-section {
    max-height: 400px;
```

```
    overflow-y: auto;
    scroll-behavior: smooth;
}

/* Smooth scrolling for alerts */
.alert-section::-webkit-scrollbar {
    width: 8px;
}

.alert-section::-webkit-scrollbar-track {
    background: #f1f1f1;
    border-radius: 10px;
}

.alert-section::-webkit-scrollbar-thumb {
    background: var(--primary-color);
    border-radius: 10px;
}

.alert-section::-webkit-scrollbar-thumb:hover {
    background: #276749;
}
</style>
</head>
<body class="bg-gray-50">
<div id="root"></div>

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

    // Enhanced mock data with more professional structure
    const mockCrops = [
        {
            name: 'Basmati Rice',
            price: '$3,250',
            unit: 'per quintal',
            change: '+8.5%',
            trend: 'up',
            volume: '2,340 tons',
            forecast: 'bullish',
            quality: 'Grade A',
            image: 'rice.png'
        },
        {
            name: 'Wheat (PBW 343)',
            price: '$2,850',
            unit: 'per quintal',
            change: '-2.3%',
```

```
            trend: 'down',
            volume: '1,890 tons',
            forecast: 'stable',
            quality: 'Premium',
            image: 'wheat.png'
        },
        {
            name: 'Hybrid Corn',
            price: '$2,100',
            unit: 'per quintal',
            change: '+12.8%',
            trend: 'up',
            volume: '3,120 tons',
            forecast: 'bullish',
            quality: 'Grade A',
            image: 'corn.png'
        },
        {
            name: 'Tomato (Fresh)',
            price: '$65',
            unit: 'per kg',
            change: '+25.4%',
            trend: 'up',
            volume: '890 tons',
            forecast: 'volatile',
            quality: 'Grade A',
            image: 'tomato.png'
        }
    ];

    const mockSchemes = [
        {
            name: 'PM-KISAN Samman Nidhi',
            amount: '$6,000',
            period: 'per year',
            type: 'Central',
            beneficiaries: '11.7 Cr',
            status: 'Active',
            icon: 'hand-holding-usd',
            description: 'Provides income support to all landholding farmers to enable them to
take care of expenses related to agriculture and allied activities as well as domestic needs.',
            eligibility: 'All landholding farmers families, Small and marginal farmers families,
Cultivators',
            documents: 'Aadhaar Card, Land ownership documents, Bank account details'
        },
        {
            name: 'Pradhan Mantri Fasal Bima Yojana',
            amount: '$2,00,000',
```

```
            period: 'coverage',
            type: 'Central',
            beneficiaries: '5.5 Cr',
            status: 'Active',
            icon: 'shield-alt',
            description: 'Provides comprehensive insurance cover against failure of the crop thus helping in stabilising the income of the farmers.',
            eligibility: 'All farmers including sharecroppers and tenant farmers growing notified crops in notified areas',
            documents: 'Land records, Aadhaar Card, Bank account details'
        },
        {
            name: 'Kisan Credit Card',
            amount: '$3,00,000',
            period: 'limit',
            type: 'Central',
            beneficiaries: '7.3 Cr',
            status: 'Active',
            icon: 'credit-card',
            description: 'Provides adequate and timely credit support from the banking system to farmers for their cultivation needs.',
            eligibility: 'Individual farmers, Joint borrowers, Tenant farmers, Oral lessees, Sharecroppers',
            documents: 'Land ownership documents, Identity proof, Address proof, Passport size photos'
        },
        {
            name: 'Soil Health Card Scheme',
            amount: 'Free Testing',
            period: 'service',
            type: 'Central',
            beneficiaries: '22 Cr',
            status: 'Active',
            icon: 'flask',
            description: 'Provides information to farmers on nutrient status of their soil along with recommendations on appropriate dosage of nutrients to be applied for improving soil health and its fertility.',
            eligibility: 'All farmers across the country.',
            documents: 'Aadhaar Card, Land ownership documents'
        }
    ];

    // Enhanced voice recognition hook
    const useVoiceRecognition = () => {
        const [isListening, setIsListening] = useState(false);
        const [transcript, setTranscript] = useState("");
        const [confidence, setConfidence] = useState(0);
        const recognition = useRef(null);
```

```
    useEffect(() => {
        if ('webkitSpeechRecognition' in window) {
            recognition.current = new window.webkitSpeechRecognition();
            recognition.current.continuous = true;
            recognition.current.interimResults = true;
            recognition.current.lang = 'hi-IN';

            recognition.current.onresult = (event) => {
                const transcript = Array.from(event.results)
                    .map(result => result[0])
                    .map(result => result.transcript)
                    .join('');
                setTranscript(transcript);
                if (event.results[event.results.length - 1].isFinal) {
                    setConfidence(event.results[event.results.length - 1][0].confidence);
                }
            };
            recognition.current.onend = () => {
                setIsListening(false);
            };
        }
    }, []);

    const startListening = () => {
        if (recognition.current) {
            setIsListening(true);
            recognition.current.start();
        }
    };

    const stopListening = () => {
        if (recognition.current) {
            recognition.current.stop();
            setIsListening(false);
        }
    };

    return { isListening, transcript, confidence, startListening, stopListening };
};

// Login Page Component
const LoginPage = ({ onLoginSuccess }) => {
    const [isLogin, setIsLogin] = useState(true);
    const [email, setEmail] = useState("");
    const [password, setPassword] = useState("");
    const [name, setName] = useState("");
    const [phone, setPhone] = useState("");
```

```
    const [errors, setErrors] = useState({});

    // Dummy credentials for demo
    const DEMO_EMAIL = "farmer@demo.com";
    const DEMO_PASSWORD = "kisan123";

    const validateLogin = () => {
      const newErrors = {};
      if (!email) newErrors.email = 'Email is required';
      if (!password) newErrors.password = 'Password is required';
      setErrors(newErrors);
      return Object.keys(newErrors).length === 0;
    };

    const validateRegister = () => {
      const newErrors = {};
      if (!name) newErrors.name = 'Name is required';
      if (!email) newErrors.email = 'Email is required';
      if (!phone) newErrors.phone = 'Phone is required';
      if (!password) newErrors.password = 'Password is required';
      setErrors(newErrors);
      return Object.keys(newErrors).length === 0;
    };

    const handleLogin = (e) => {
      e.preventDefault();
      if (validateLogin()) {
        // For demo purposes, accept any credentials but show success only for demo
credentials
        if (email === DEMO_EMAIL && password === DEMO_PASSWORD) {
          onLoginSuccess({
            name: "Demo Farmer",
            email: DEMO_EMAIL,
            phone: "9876543210"
          });
        } else {
          alert("Invalid credentials. For demo, use:\nEmail:
farmer@demo.com\nPassword: kisan123");
        }
      }
    };

    const handleRegister = (e) => {
      e.preventDefault();
      if (validateRegister()) {
        // Simulate registration
        onLoginSuccess({
          name,
```

```
                email,
                phone
            });
        }
    };

    const handleGoogleLogin = () => {
        // Simulate Google login
        onLoginSuccess({
            name: "Google User",
            email: "googleuser@gmail.com",
            phone: "9876543210"
        });
    };

    return (
        <div className="login-bg flex items-center justify-center p-4">
            <div className="app-container max-w-5xl w-full grid grid-cols-1 lg:grid-cols-2
gap-8">
                {/* Left side - App Info */}
                <div className="text-white">
                    <div className="flex items-center space-x-3 mb-6">
                        <div className="w-12 h-12 bg-white bg-opacity-20 rounded-lg flex
items-center justify-center">
                            <i className="fas fa-seedling text-white text-xl"></i>
                        </div>
                        <h1 className="text-3xl font-bold">Kisan AI</h1>
                    </div>

                    <h2 className="text-4xl font-bold mb-6">Your Smart Farming
Companion</h2>
                    <p className="text-xl mb-8">AI-powered agricultural assistant to maximize
your crop yield and profits</p>

                    <div className="grid grid-cols-2 gap-6">
                        <div className="bg-white bg-opacity-10 p-6 rounded-xl">
                            <div className="feature-icon bg-green-100 text-green-600">
                                <i className="fas fa-leaf"></i>
                            </div>
                            <h3 className="text-xl font-semibold mb-2">Crop Doctor</h3>
                            <p className="text-white text-opacity-80">Diagnose plant diseases
with AI</p>
                        </div>

                        <div className="bg-white bg-opacity-10 p-6 rounded-xl">
                            <div className="feature-icon bg-blue-100 text-blue-600">
                                <i className="fas fa-chart-line"></i>
                            </div>
```

```
                        <h3 className="text-xl font-semibold mb-2">Market Prices</h3>
                        <p className="text-white text-opacity-80">Real time mandi prices</p>
                    </div>

                    <div className="bg-white bg-opacity-10 p-6 rounded-xl">
                        <div className="feature-icon bg-purple-100 text-purple-600">
                            <i className="fas fa-microphone"></i>
                        </div>
                        <h3 className="text-xl font-semibold mb-2">Voice Assistant</h3>
                        <p className="text-white text-opacity-80">Ask in your local
language</p>
                    </div>

                    <div className="bg-white bg-opacity-10 p-6 rounded-xl">
                        <div className="feature-icon bg-yellow-100 text-yellow-600">
                            <i className="fas fa-seedling"></i>
                        </div>
                        <h3 className="text-xl font-semibold mb-2">Crop Advisor</h3>
                        <p className="text-white text-opacity-80">Personalized
recommendations</p>
                    </div>
                </div>
            </div>

            {/* Right side - Login/Register Form */}
            <div className="login-card rounded-2xl shadow-xl p-8">
                <div className="text-center mb-8">
                    <h2 className="text-2xl font-bold text-gray-800">
                        {isLogin ? 'Welcome Back' : 'Create Account'}
                    </h2>
                    <p className="text-gray-600">
                        {isLogin ? 'Sign in to continue to Kisan AI' : 'Join Kisan AI to access
smart farming tools'}
                    </p>
                </div>

                <button
                    onClick={handleGoogleLogin}
                    className="google-btn w-full flex items-center justify-center space-x-3
bg-white border border-gray-300 rounded-lg py-3 px-4 hover:bg-gray-50 transition mb-4"
                >
                    <img
src="https://static.vecteezy.com/system/resources/previews/013/948/549/non_2x/google-log
o-on-transparent-white-background-free-vector.jpg" alt="Google" />
                    <span>{isLogin ? 'Sign in with Google' : 'Sign up with Google'}</span>
                </button>

                <div className="relative my-6">
```

```
                    <div className="absolute inset-0 flex items-center">
                      <div className="w-full border-t border-gray-300"></div>
                    </div>
                    <div className="relative flex justify-center text-sm">
                      <span className="px-2 bg-white text-gray-500">Or continue with
email</span>
                    </div>
                  </div>

                  {isLogin ? (
                    <form onSubmit={handleLogin} className="space-y-4">
                      <div>
                        <label htmlFor="email" className="block text-sm font-medium
text-gray-700 mb-1">Email</label>
                        <input
                          type="email"
                          id="email"
                          value={email}
                          onChange={(e) => setEmail(e.target.value)}
                          className={`w-full input-field py-2 px-3 rounded-lg ${errors.email ?
'border-red-500' : ''}`}
                          placeholder="farmer@example.com"
                        />
                        {errors.email && <p className="text-red-500 text-xs
mt-1">{errors.email}</p>}
                      </div>

                      <div>
                        <label htmlFor="password" className="block text-sm font-medium
text-gray-700 mb-1">Password</label>
                        <input
                          type="password"
                          id="password"
                          value={password}
                          onChange={(e) => setPassword(e.target.value)}
                          className={`w-full input-field py-2 px-3 rounded-lg
${errors.password ? 'border-red-500' : ''}`}
                          placeholder="••••••••"
                        />
                        {errors.password && <p className="text-red-500 text-xs
mt-1">{errors.password}</p>}
                      </div>

                      <div className="flex items-center justify-between">
                        <div className="flex items-center">
                          <input
                            id="remember-me"
                            name="remember-me"
```

```
                                type="checkbox"
                                className="h-4 w-4 text-green-600 focus:ring-green-600
border-gray-300 rounded"
                            />
                            <label htmlFor="remember-me" className="ml-2 block text-sm
text-gray-700">
                                Remember me
                            </label>
                        </div>
                        <a href="#" className="text-sm text-green-600
hover:text-green-500">
                            Forgot password?
                        </a>
                    </div>

                    <button type="submit" className="btn-primary w-full py-3 px-4
rounded-lg">
                        Sign In
                    </button>

                    <div className="text-center text-sm text-gray-600">
                        Don't have an account?{' '}
                        <button
                            type="button"
                            onClick={() => setIsLogin(false)}
                            className="text-green-600 font-medium ml-1"
                        >
                            Sign Up
                        </button>
                    </div>


                </form>
            ) : (
                <form onSubmit={handleRegister} className="space-y-4">
                    <div>
                        <label htmlFor="name" className="block text-sm font-medium
text-gray-700 mb-1">Full Name</label>
                        <input
                            type="text"
                            id="name"
                            value={name}
                            onChange={(e) => setName(e.target.value)}
                            className={`w-full input-field py-2 px-3 rounded-lg ${errors.name
? 'border-red-500' : ''}`}
                            placeholder="Farmer Name"
                        />
```

```
                {errors.name && <p className="text-red-500 text-xs
mt-1">{errors.name}</p>}
              </div>

              <div>
                <label htmlFor="email" className="block text-sm font-medium
text-gray-700 mb-1">Email</label>
                <input
                  type="email"
                  id="email"
                  value={email}
                  onChange={(e) => setEmail(e.target.value)}
                  className={`w-full input-field py-2 px-3 rounded-lg ${errors.email ?
'border-red-500' : ''}`}
                  placeholder="farmer@example.com"
                />
                {errors.email && <p className="text-red-500 text-xs
mt-1">{errors.email}</p>}
              </div>

              <div>
                <label htmlFor="phone" className="block text-sm font-medium
text-gray-700 mb-1">Phone Number</label>
                <input
                  type="tel"
                  id="phone"
                  value={phone}
                  onChange={(e) => setPhone(e.target.value)}
                  className={`w-full input-field py-2 px-3 rounded-lg ${errors.phone
? 'border-red-500' : ''}`}
                  placeholder="+91 9876543210"
                />
                {errors.phone && <p className="text-red-500 text-xs
mt-1">{errors.phone}</p>}
              </div>

              <div>
                <label htmlFor="password" className="block text-sm font-medium
text-gray-700 mb-1">Password</label>
                <input
                  type="password"
                  id="password"
                  value={password}
                  onChange={(e) => setPassword(e.target.value)}
                  className={`w-full input-field py-2 px-3 rounded-lg
${errors.password ? 'border-red-500' : ''}`}
                  placeholder="••••••••"
                />
```

```
                        {errors.password && <p className="text-red-500 text-xs
mt-1">{errors.password}</p>}
                    </div>

                    <button type="submit" className="btn-primary w-full py-3 px-4
rounded-lg">
                        Create Account
                    </button>

                    <div className="text-center text-sm text-gray-600">
                        Already have an account?{' '}
                        <button
                            type="button"
                            onClick={() => setIsLogin(true)}
                            className="text-green-600 font-medium ml-1"
                        >
                            Sign In
                        </button>
                    </div>
                </form>
            )}
        </div>
    </div>
  </div>
 );
};

// Professional Navigation Component
const Navigation = ({ activeTab, setActiveTab, onLogout }) => {
  const [isScrolled, setIsScrolled] = useState(false);

  useEffect(() => {
    const handleScroll = () => {
      setIsScrolled(window.scrollY > 20);
    };
    window.addEventListener('scroll', handleScroll);
    return () => window.removeEventListener('scroll', handleScroll);
  }, []);

  const tabs = [
    { id: 'dashboard', label: 'Dashboard', icon: 'chart-pie', badge: null },
    { id: 'crop-disease', label: 'Crop Doctor', icon: 'leaf', badge: 'AI' },
    { id: 'market', label: 'Market Prices', icon: 'chart-line', badge: 'Live' },
    { id: 'voice', label: 'Voice Assistant', icon: 'microphone', badge: 'Beta' },
    { id: 'recommendations', label: 'Crop Advisor', icon: 'seedling', badge: null },
    { id: 'schemes', label: 'Govt Schemes', icon: 'file-invoice-dollar', badge: null }
  ];
```

```
      return (
        <header className={`sticky top-0 z-50 bg-white shadow-sm transition-all
duration-300 ${isScrolled ? 'py-2' : 'py-4'}`}>
          <div className="container mx-auto px-4">
            <div className="flex items-center justify-between">
              <div className="flex items-center space-x-2">
                <div className="w-10 h-10 bg-green-600 rounded-lg flex items-center
justify-center">
                  <i className="fas fa-seedling text-white"></i>
                </div>
                <h1 className="text-xl font-bold text-gray-800">Kisan AI</h1>
              </div>

              <nav className="hidden md:flex items-center space-x-8">
                {tabs.map((tab) => (
                  <button
                    key={tab.id}
                    onClick={() => setActiveTab(tab.id)}
                    className={`nav-item ${activeTab === tab.id ? 'active' :
'text-gray-600'}`}
                  >
                    <i className={`fas fa-${tab.icon} mr-2`}></i>
                    {tab.label}
                    {tab.badge && (
                      <span className={`badge badge-${tab.badge.toLowerCase()}
ml-2`}>
                        {tab.badge}
                      </span>
                    )}
                  </button>
                ))}
              </nav>

              <div className="flex items-center space-x-4">
                <button
                  onClick={onLogout}
                  className="hidden md:block bg-gray-100 text-gray-800 px-4 py-2
rounded-lg hover:bg-gray-200 transition"
                >
                  <i className="fas fa-sign-out-alt mr-2"></i>
                  Logout
                </button>
                <button className="md:hidden text-gray-600">
                  <i className="fas fa-bars text-xl"></i>
                </button>
              </div>
            </div>
          </div>
```

```
        </header>
    );
};

// Enhanced Dashboard Component
const Dashboard = ({ setActiveTab }) => {
    const [currentTime, setCurrentTime] = useState(new Date());
    const [weatherData, setWeatherData] = useState({
        temp: 28,
        condition: 'sunny',
        humidity: 65,
        windSpeed: 12
    });
    const [reminders, setReminders] = useState([]);

    useEffect(() => {
        const timer = setInterval(() => setCurrentTime(new Date()), 1000);
        return () => clearInterval(timer);
    }, []);

    const handleAddReminder = (reminderText) => {
        if (reminderText.trim()) {
            setReminders([...reminders, {
                id: Date.now(),
                text: reminderText,
                date: new Date().toLocaleDateString()
            }]);
        }
    };

    return (
        <div className="container mx-auto px-4 py-8 space-y-8 slide-in">
            {/* Hero Section */}
            <div className="hero-gradient rounded-2xl p-8 text-center">
                <h1 className="text-3xl md:text-4xl font-bold text-gray-800 mb-4">
                    Welcome to Your Smart Farming Assistant
                </h1>
                <p className="text-lg text-gray-600 max-w-2xl mx-auto">
                    Get AI-powered insights to maximize your crop yield and profits
                </p>

                <div className="flex flex-wrap items-center justify-center gap-4 mt-6">
                    <div className="flex items-center bg-white px-4 py-2 rounded-lg
shadow-sm">
                        <i className="fas fa-map-marker-alt text-green-600 mr-2"></i>
                        <span>Tirupati, Andhra Pradesh</span>
                    </div>
```

```
                <div className="flex items-center bg-white px-4 py-2 rounded-lg
shadow-sm">
                  <i className="fas fa-clock text-green-600 mr-2"></i>
                  <span>{currentTime.toLocaleTimeString()}</span>
                </div>
                <div className="flex items-center bg-white px-4 py-2 rounded-lg
shadow-sm">
                  <i className="fas fa-calendar-day text-green-600 mr-2"></i>
                  <span>{currentTime.toLocaleDateString()}</span>
                </div>
              </div>
            </div>

            {/* Feature Cards */}
            <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-6">
              {[
                {
                  title: 'Crop Doctor',
                  desc: 'Diagnose plant diseases with AI',
                  icon: 'leaf',
                  color: 'text-green-500',
                  stats: '95% Accuracy',
                  onClick: () => setActiveTab('crop-disease')
                },
                {
                  title: 'Market Prices',
                  desc: 'Real-time prices from 500+ markets',
                  icon: 'chart-line',
                  color: 'text-blue-500',
                  stats: 'Live Updates',
                  onClick: () => setActiveTab('market')
                },
                {
                  title: 'Voice Assistant',
                  desc: 'Ask questions in your language',
                  icon: 'microphone',
                  color: 'text-purple-500',
                  stats: '12 Languages',
                  onClick: () => setActiveTab('voice')
                },
                {
                  title: 'Crop Advisor',
                  desc: 'Personalized planting advice',
                  icon: 'seedling',
                  color: 'text-yellow-500',
                  stats: 'ML Powered',
                  onClick: () => setActiveTab('recommendations')
                }
```

```
            ].map((card, index) => (
              <button
                key={index}
                onClick={card.onClick}
                className="text-left"
              >
                <div className="card feature-card p-6">
                  <div className={`text-3xl mb-4 ${card.color}`}>
                    <i className={`fas fa-${card.icon}`}></i>
                  </div>
                  <h3 className="text-lg font-semibold text-gray-800
mb-2">{card.title}</h3>
                  <p className="text-gray-600 text-sm mb-3">{card.desc}</p>
                  <span className="text-xs font-medium bg-gray-100 text-gray-700 px-2
py-1 rounded-full">
                    {card.stats}
                  </span>
                </div>
              </button>
            ))}
          </div>

          {/* Market Overview */}
          <div className="bg-white rounded-2xl shadow-sm p-6">
            <div className="flex flex-wrap items-center justify-between mb-6">
              <h3 className="text-xl font-semibold text-gray-800 flex items-center">
                <i className="fas fa-chart-bar text-blue-500 mr-3"></i>
                Today's Market Prices
              </h3>
              <button
                onClick={() => setActiveTab('market')}
                className="text-green-600 text-sm hover:underline"
              >
                View All Market Data
              </button>
            </div>

            <div className="overflow-x-auto">
              <table className="min-w-full divide-y divide-gray-200">
                <thead className="bg-gray-50">
                  <tr>
                    <th className="px-6 py-3 text-left text-xs font-medium text-gray-500
uppercase tracking-wider">Crop</th>
                    <th className="px-6 py-3 text-left text-xs font-medium text-gray-500
uppercase tracking-wider">Quality</th>
                    <th className="px-6 py-3 text-left text-xs font-medium text-gray-500
uppercase tracking-wider">Price</th>
```

```
                        <th className="px-6 py-3 text-left text-xs font-medium text-gray-500
uppercase tracking-wider">Change</th>
                        <th className="px-6 py-3 text-left text-xs font-medium text-gray-500
uppercase tracking-wider">Volume</th>
                        <th className="px-6 py-3 text-left text-xs font-medium text-gray-500
uppercase tracking-wider">Forecast</th>
                    </tr>
                </thead>
                <tbody className="bg-white divide-y divide-gray-200">
                    {mockCrops.map((crop, index) => (
                        <tr key={index} className="hover:bg-gray-50 transition">
                            <td className="px-6 py-4 whitespace-nowrap">
                                <div className="flex items-center">
                                    <div className="flex-shrink-0 h-10 w-10 bg-green-100
rounded-lg flex items-center justify-center">
                                        <i className="fas fa-seedling text-green-600"></i>
                                    </div>
                                    <div className="ml-4">
                                        <div className="text-sm font-medium
text-gray-900">{crop.name}</div>
                                    </div>
                                </div>
                            </td>
                            <td className="px-6 py-4 whitespace-nowrap">
                                <span className="px-2 py-1 text-xs rounded-full bg-blue-100
text-blue-800">{crop.quality}</span>
                            </td>
                            <td className="px-6 py-4 whitespace-nowrap">
                                <div className="text-sm font-semibold
text-gray-900">{crop.price}</div>
                                <div className="text-xs text-gray-500">{crop.unit}</div>
                            </td>
                            <td className="px-6 py-4 whitespace-nowrap">
                                <span className={`flex items-center ${crop.trend === 'up' ?
'text-green-600' : 'text-red-600'}`}>
                                    <i className={`fas ${crop.trend === 'up' ? 'fa-arrow-up' :
'fa-arrow-down'} mr-1`}></i>
                                    {crop.change}
                                </span>
                            </td>
                            <td className="px-6 py-4 whitespace-nowrap text-sm
text-gray-500">
                                {crop.volume}
                            </td>
                            <td className="px-6 py-4 whitespace-nowrap">
                                <span className={`px-2 py-1 text-xs rounded-full font-medium
${
                                    crop.forecast === 'bullish' ? 'bg-green-100 text-green-800' :
```

```
                                    crop.forecast === 'stable' ? 'bg-blue-100 text-blue-800' :
'bg-yellow-100 text-yellow-800'
                                }`}>
                                    {crop.forecast}
                                </span>
                            </td>
                        </tr>
                    ))}
                </tbody>
            </table>
        </div>
    </div>

    {/* Weather & Insights */}
    <div className="grid grid-cols-1 lg:grid-cols-2 gap-6">
        {/* Weather Card */}
        <div className="bg-white rounded-2xl shadow-sm p-6">
            <h3 className="text-xl font-semibold text-gray-800 flex items-center mb-4">
                <i className="fas fa-cloud-sun text-yellow-500 mr-3"></i>
                Weather Forecast
            </h3>

            <div className="grid grid-cols-3 gap-4">
                <div className="bg-green-50 rounded-lg p-4 text-center">
                    <div className="text-3xl font-bold text-gray-800
mb-1">{weatherData.temp}°C</div>
                    <div className="text-sm text-gray-600">Temperature</div>
                </div>

                <div className="bg-blue-50 rounded-lg p-4 text-center">
                    <div className="text-3xl font-bold text-gray-800
mb-1">{weatherData.humidity}%</div>
                    <div className="text-sm text-gray-600">Humidity</div>
                </div>

                <div className="bg-purple-50 rounded-lg p-4 text-center">
                    <div className="text-3xl font-bold text-gray-800
mb-1">{weatherData.windSpeed} km/h</div>
                    <div className="text-sm text-gray-600">Wind Speed</div>
                </div>
            </div>

            <div className="mt-4 pt-4 border-t border-gray-100">
                <p className="text-green-600 font-medium">
                    <i className="fas fa-check-circle mr-2"></i>
                    Good weather for field operations today
                </p>
            </div>
```

```
            </div>

            {/* AI Insights */}
            <div className="bg-white rounded-2xl shadow-sm p-6">
              <h3 className="text-xl font-semibold text-gray-800 flex items-center mb-4">
                <i className="fas fa-lightbulb text-green-500 mr-3"></i>
                AI Farming Insights
              </h3>

              <div className="space-y-4">
                <div className="flex items-start">
                  <div className="flex-shrink-0 mt-1">
                    <div className="w-2 h-2 bg-green-500 rounded-full pulse"></div>
                  </div>
                  <div className="ml-3">
                    <p className="text-gray-700">Optimal planting time for wheat in
next 7 days</p>
                  </div>
                </div>

                <div className="flex items-start">
                  <div className="flex-shrink-0 mt-1">
                    <div className="w-2 h-2 bg-blue-500 rounded-full pulse"></div>
                  </div>
                  <div className="ml-3">
                    <p className="text-gray-700">Rice prices expected to rise by 12%
this month</p>
                  </div>
                </div>

                <div className="flex items-start">
                  <div className="flex-shrink-0 mt-1">
                    <div className="w-2 h-2 bg-yellow-500 rounded-full pulse"></div>
                  </div>
                  <div className="ml-3">
                    <p className="text-gray-700">Apply for PM-KISAN scheme -
deadline in 15 days</p>
                  </div>
                </div>
              </div>

              <div className="mt-4 pt-4 border-t border-gray-100">
                <button
                  onClick={() => setActiveTab('schemes')}
                  className="btn-primary px-4 py-2 rounded-lg"
                >
                  <i className="fas fa-file-invoice-dollar mr-2"></i>
                  View Govt Schemes
```

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

              {/* Reminders List */}
              {reminders.length > 0 && (
                <div className="bg-white rounded-2xl shadow-sm p-6">
                  <h3 className="text-xl font-semibold text-gray-800 flex items-center mb-4">
                    <i className="fas fa-bell text-yellow-500 mr-3"></i>
                    Your Reminders
                  </h3>

                  <div className="space-y-3">
                    {reminders.map((reminder) => (
                      <div key={reminder.id} className="border border-gray-200 rounded-lg
p-4 flex justify-between items-center">
                        <div>
                          <p className="text-gray-800">{reminder.text}</p>
                          <p className="text-gray-500 text-sm mt-1">{reminder.date}</p>
                        </div>
                        <button
                          onClick={() => setReminders(reminders.filter(r => r.id !==
reminder.id))}
                          className="text-red-500 hover:text-red-700"
                        >
                          <i className="fas fa-trash"></i>
                        </button>
                      </div>
                    ))}
                  </div>
                </div>
              )}
            </div>
          );
        };

        // Enhanced Crop Disease Detection Component
        const CropDiseaseDetection = ({ setActiveTab }) => {
          const [selectedImage, setSelectedImage] = useState(null);
          const [isAnalyzing, setIsAnalyzing] = useState(false);
          const [results, setResults] = useState(null);
          const [confidence, setConfidence] = useState(0);
          const fileInputRef = useRef(null);

          const handleImageUpload = (event) => {
            const file = event.target.files[0];
            if (file) {
```

```
        const reader = new FileReader();
        reader.onload = (e) => {
            setSelectedImage(e.target.result);
        };
        reader.readAsDataURL(file);
    }
};

const analyzeImage = () => {
    setIsAnalyzing(true);
    setResults(null);

    // Simulate AI analysis
    setTimeout(() => {
        const mockResults = [
            {
                disease: "Leaf Blight",
                probability: 87,
                severity: "Moderate",
                treatment: "Apply copper-based fungicide within 24 hours",
                prevention: "Ensure proper drainage and avoid overhead watering"
            },
            {
                disease: "Healthy Plant",
                probability: 13,
                severity: "None",
                treatment: "Continue regular care routine",
                prevention: "Maintain current practices"
            }
        ];
        setResults(mockResults);
        setConfidence(87);
        setIsAnalyzing(false);
    }, 3000);
};

return (
    <div className="container mx-auto px-4 py-8 space-y-8 slide-in">
        <div className="back-button" onClick={() => setActiveTab('dashboard')}>
            <i className="fas fa-arrow-left"></i>
            <span>Back to Dashboard</span>
        </div>

        <div className="text-center">
            <h2 className="text-2xl md:text-3xl font-bold text-gray-800 mb-2">
                AI Crop Disease Detection
            </h2>
            <p className="text-gray-600 max-w-2xl mx-auto">
```

```
                Upload a photo of your crop and get instant diagnosis with treatment
recommendations
              </p>
            </div>

            <div className="grid grid-cols-1 lg:grid-cols-2 gap-8">
              {/* Image Upload Section */}
              <div className="bg-white rounded-2xl shadow-sm p-6">
                <h3 className="text-lg font-semibold text-gray-800 mb-4 flex items-center">
                  <i className="fas fa-camera text-green-500 mr-3"></i>
                  Upload Crop Image
                </h3>

                <div
                  className="border-2 border-dashed border-gray-300 rounded-xl p-8
text-center cursor-pointer transition hover:border-green-500 hover:bg-green-50"
                  onClick={() => fileInputRef.current?.click()}
                >
                  {selectedImage ? (
                    <div className="space-y-4">
                      <img
                        src={selectedImage}
                        alt="Uploaded crop"
                        className="w-full h-64 object-contain rounded-lg"
                      />
                      <p className="text-green-600 font-medium">Image uploaded
successfully</p>
                    </div>
                  ) : (
                    <div className="space-y-4">
                      <div className="w-16 h-16 bg-green-100 rounded-2xl flex
items-center justify-center mx-auto">
                        <i className="fas fa-plus text-green-500 text-2xl"></i>
                      </div>
                      <div>
                        <p className="text-gray-800 font-medium mb-2">Click to upload
crop image</p>
                        <p className="text-gray-500 text-sm">Supports JPG, PNG up to
10MB</p>
                      </div>
                    </div>
                  )}
                </div>

                <input
                  ref={fileInputRef}
                  type="file"
                  accept="image/*"
```

```
                    onChange={handleImageUpload}
                    className="hidden"
                />

                {selectedImage && (
                    <div className="flex space-x-3 mt-4">
                        <button
                            onClick={() => fileInputRef.current?.click()}
                            className="flex-1 bg-gray-100 text-gray-800 py-3 px-6 rounded-lg
font-medium hover:bg-gray-200 transition"
                        >
                            <i className="fas fa-sync-alt mr-2"></i>
                            Change Image
                        </button>
                        <button
                            onClick={analyzeImage}
                            disabled={isAnalyzing}
                            className="flex-1 btn-primary py-3 px-6 rounded-lg font-medium
disabled:opacity-70"
                        >
                            {isAnalyzing ? (
                                <>
                                    <i className="fas fa-spinner fa-spin mr-2"></i>
                                    Analyzing...
                                </>
                            ) : (
                                <>
                                    <i className="fas fa-search mr-2"></i>
                                    Analyze
                                </>
                            )}
                        </button>
                    </div>
                )}
            </div>

            {/* Results Section */}
            <div className="bg-white rounded-2xl shadow-sm p-6">
                <h3 className="text-lg font-semibold text-gray-800 mb-4 flex items-center">
                    <i className="fas fa-chart-pie text-blue-500 mr-3"></i>
                    Analysis Results
                </h3>

                {isAnalyzing && (
                    <div className="space-y-4">
                        <div className="flex items-center space-x-3">
                            <div className="w-8 h-8 border-2 border-blue-500
border-t-transparent rounded-full animate-spin"></div>
```

```
                        <span className="text-gray-600">AI is analyzing your crop
image...</span>
                    </div>
                    <div className="space-y-3">
                        {[1, 2, 3].map((i) => (
                            <div key={i} className="h-4 bg-gray-200 rounded-lg
animate-pulse"></div>
                        ))}
                    </div>
                </div>
            )}

            {results && !isAnalyzing && (
                <div className="space-y-6">
                    <div className="bg-blue-50 rounded-xl p-4">
                        <div className="flex items-center justify-between mb-3">
                            <span className="text-gray-700 font-medium">Confidence
Level</span>
                            <span className="text-green-600
font-bold">{confidence}%</span>
                        </div>
                        <div className="w-full bg-gray-200 rounded-full h-2">
                            <div
                                className="bg-green-500 h-2 rounded-full"
                                style={{ width: `${confidence}%` }}
                            ></div>
                        </div>
                    </div>

                    <div className="space-y-4">
                        {results.map((result, index) => (
                            <div key={index} className="border border-gray-200 rounded-xl
p-4">
                                <div className="flex items-start justify-between mb-3">
                                    <div>
                                        <h4 className="text-gray-800
font-semibold">{result.disease}</h4>
                                        <p className="text-gray-500 text-sm">Probability:
{result.probability}%</p>
                                    </div>
                                    <div className={`px-3 py-1 rounded-full text-xs font-medium
${
                                        result.severity === 'None' ? 'bg-green-100 text-green-800' :
                                        result.severity === 'Moderate' ? 'bg-yellow-100
text-yellow-800' :
                                        'bg-red-100 text-red-800'
                                    }`}>
                                        {result.severity}
```

```
                                    </div>
                                </div>
                                <div className="space-y-2">
                                    <div>
                                        <span className="text-gray-700 text-sm
font-medium">Treatment: </span>
                                        <span className="text-gray-600
text-sm">{result.treatment}</span>
                                    </div>
                                    <div>
                                        <span className="text-gray-700 text-sm
font-medium">Prevention: </span>
                                        <span className="text-gray-600
text-sm">{result.prevention}</span>
                                    </div>
                                </div>
                            </div>
                        ))}
                    </div>
                </div>
            )}

            {!selectedImage && !isAnalyzing && (
                <div className="text-center py-12">
                    <div className="w-16 h-16 bg-gray-100 rounded-2xl flex items-center
justify-center mx-auto mb-4">
                        <i className="fas fa-microscope text-gray-400 text-2xl"></i>
                    </div>
                    <p className="text-gray-500">Upload an image to see AI analysis
results</p>
                </div>
            )}
        </div>
    </div>
  </div>
);
};

// Market Intelligence Component
const MarketIntelligence = ({ setActiveTab }) => {
  const [selectedMarket, setSelectedMarket] = useState('all');
  const [priceAlerts, setPriceAlerts] = useState([]);
  const [showAlertForm, setShowAlertForm] = useState(false);
  const [alertCrop, setAlertCrop] = useState("");
  const [alertPrice, setAlertPrice] = useState("");
  const [alertCondition, setAlertCondition] = useState('above');
  const alertSectionRef = useRef(null);
```

```
const markets = [
    { id: 'all', name: 'All Markets', count: 500 },
    { id: 'ap', name: 'Andhra Pradesh', count: 45 },
    { id: 'tn', name: 'Tamil Nadu', count: 38 },
    { id: 'ka', name: 'Karnataka', count: 42 },
    { id: 'mh', name: 'Maharashtra', count: 55 }
];

const handleSetAlert = (crop) => {
    setAlertCrop(crop.name);
    setAlertPrice(crop.price.replace('$', '').replace(',', ''));
    setShowAlertForm(true);
    // Scroll to alert section
    setTimeout(() => {
        alertSectionRef.current?.scrollIntoView({ behavior: 'smooth' });
    }, 100);
};

const handleSaveAlert = () => {
    if (alertCrop && alertPrice) {
        setPriceAlerts([...priceAlerts, {
            id: Date.now(),
            crop: alertCrop,
            price: alertPrice,
            condition: alertCondition
        }]);
        setShowAlertForm(false);
        setAlertCrop("");
        setAlertPrice("");
    }
};

return (
    <div className="container mx-auto px-4 py-8 space-y-8 slide-in">
        <div className="back-button" onClick={() => setActiveTab('dashboard')}>
            <i className="fas fa-arrow-left"></i>
            <span>Back to Dashboard</span>
        </div>

        <div className="text-center">
            <h2 className="text-2xl md:text-3xl font-bold text-gray-800 mb-2">
                Market Intelligence
            </h2>
            <p className="text-gray-600 max-w-2xl mx-auto">
                Real-time market data from 500+ mandis across India with price predictions
            </p>
        </div>
```

```
            {/* Market Filters */}
            <div className="bg-white rounded-2xl shadow-sm p-6">
              <div className="flex flex-wrap gap-2">
                {markets.map((market) => (
                  <button
                    key={market.id}
                    onClick={() => setSelectedMarket(market.id)}
                    className={`tab-button ${selectedMarket === market.id ? 'active' : ''}`}
                  >
                    {market.name}
                    <span className="ml-1 text-xs opacity-70">({market.count})</span>
                  </button>
                ))}
              </div>
            </div>

            {/* Price Cards */}
            <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
              {mockCrops.map((crop, index) => (
                <div key={index} className="card bg-white rounded-2xl shadow-sm p-6">
                  <div className="flex items-center justify-between mb-4">
                    <div className="flex items-center space-x-4">
                      <div className="w-12 h-12 bg-green-100 rounded-xl flex
items-center justify-center">
                        <i className="fas fa-seedling text-green-600"></i>
                      </div>
                      <div>
                        <h3 className="text-gray-800 font-semibold">{crop.name}</h3>
                        <p className="text-gray-500 text-sm">{crop.quality}</p>
                      </div>
                    </div>
                    <div className="text-right">
                      <div className="text-xl font-bold text-gray-800">{crop.price}</div>
                      <div className="text-gray-500 text-sm">{crop.unit}</div>
                    </div>
                  </div>

                  <div className="space-y-3">
                    <div className="flex items-center justify-between">
                      <span className="text-gray-600 text-sm">24h Change</span>
                      <span className={`font-semibold ${crop.trend === 'up' ?
'text-green-600' : 'text-red-600'}`}>
                        {crop.change}
                      </span>
                    </div>
                    <div className="flex items-center justify-between">
                      <span className="text-gray-600 text-sm">Volume</span>
```

```
                        <span className="text-gray-800
font-medium">{crop.volume}</span>
                    </div>
                    <div className="flex items-center justify-between">
                        <span className="text-gray-600 text-sm">Forecast</span>
                        <span className={`px-2 py-1 rounded-full text-xs font-medium ${
                            crop.forecast === 'bullish' ? 'bg-green-100 text-green-800' :
                            crop.forecast === 'stable' ? 'bg-blue-100 text-blue-800' :
'bg-yellow-100 text-yellow-800'
                        }`}>
                            {crop.forecast}
                        </span>
                    </div>
                </div>

                <div className="mt-4 pt-4 border-t border-gray-100">
                    <button
                        onClick={() => handleSetAlert(crop)}
                        className="btn-secondary w-full py-2 px-4 rounded-lg"
                    >
                        <i className="fas fa-bell mr-2"></i>
                        Set Price Alert
                    </button>
                </div>
            </div>
        ))}
    </div>

    {/* Price Alerts Section */}
    <div className="bg-white rounded-2xl shadow-sm p-6" ref={alertSectionRef}>
        <div className="flex justify-between items-center mb-4">
            <h3 className="text-xl font-semibold text-gray-800 flex items-center">
                <i className="fas fa-bell text-yellow-500 mr-3"></i>
                Your Price Alerts
            </h3>
            <button
                onClick={() => setShowAlertForm(true)}
                className="btn-primary px-4 py-2 rounded-lg"
            >
                <i className="fas fa-plus mr-2"></i>
                Add New Alert
            </button>
        </div>

        {priceAlerts.length > 0 ? (
            <div className="alert-section space-y-3">
                {priceAlerts.map((alert) => (
```

```
                        <div key={alert.id} className="border border-gray-200 rounded-lg p-4
flex justify-between items-center">
                            <div>
                                <p className="text-gray-800 font-medium">{alert.crop}</p>
                                <p className="text-gray-600 text-sm mt-1">
                                    Alert when price goes {alert.condition} ${alert.price}
                                </p>
                            </div>
                            <button
                                onClick={() => setPriceAlerts(priceAlerts.filter(a => a.id !==
alert.id))}
                                className="text-red-500 hover:text-red-700"
                            >
                                <i className="fas fa-trash"></i>
                            </button>
                        </div>
                    ))}
                </div>
            ) : (
                <div className="text-center py-8">
                    <div className="w-16 h-16 bg-gray-100 rounded-2xl flex items-center
justify-center mx-auto mb-4">
                        <i className="fas fa-bell-slash text-gray-400 text-2xl"></i>
                    </div>
                    <p className="text-gray-500">No price alerts set up yet</p>
                </div>
            )}
        </div>

        {/* Alert Form */}
        {showAlertForm && (
            <div className="bg-white rounded-2xl shadow-sm p-6">
                <div className="flex justify-between items-center mb-4">
                    <h3 className="text-xl font-semibold text-gray-800">Set Price Alert</h3>
                    <button
                        onClick={() => {
                            setShowAlertForm(false);
                            setAlertCrop("");
                            setAlertPrice("");
                        }}
                        className="text-gray-500 hover:text-gray-700"
                    >
                        <i className="fas fa-times"></i>
                    </button>
                </div>

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

```
                        <label className="block text-sm font-medium text-gray-700
mb-1">Crop</label>
                        <input
                          type="text"
                          value={alertCrop}
                          onChange={(e) => setAlertCrop(e.target.value)}
                          className="w-full input-field py-2 px-3 rounded-lg"
                          placeholder="Enter crop name"
                        />
                      </div>

                      <div>
                        <label className="block text-sm font-medium text-gray-700
mb-1">Alert When Price</label>
                        <div className="flex items-center space-x-3">
                          <select
                            value={alertCondition}
                            onChange={(e) => setAlertCondition(e.target.value)}
                            className="input-field py-2 px-3 rounded-lg"
                          >
                            <option value="above">Goes Above</option>
                            <option value="below">Goes Below</option>
                          </select>
                          <div className="relative flex-1">
                            <span className="absolute left-3 top-1/2 transform
-translate-y-1/2 text-gray-500">$</span>
                            <input
                              type="number"
                              value={alertPrice}
                              onChange={(e) => setAlertPrice(e.target.value)}
                              className="w-full input-field py-2 px-3 pl-8 rounded-lg"
                              placeholder="Enter price"
                            />
                          </div>
                        </div>
                      </div>

                      <div className="flex justify-end space-x-3 mt-6">
                        <button
                          onClick={() => {
                            setShowAlertForm(false);
                            setAlertCrop("");
                            setAlertPrice("");
                          }}
                          className="bg-gray-100 text-gray-800 px-4 py-2 rounded-lg
hover:bg-gray-200 transition"
                        >
                          Cancel
```

```
                        </button>
                        <button
                          onClick={handleSaveAlert}
                          className="btn-primary px-4 py-2 rounded-lg"
                        >
                          Save Alert
                        </button>
                      </div>
                    </div>
                  </div>
                )}
              </div>
            );
          };

          // Voice Assistant Component
          const VoiceAssistant = ({ setActiveTab }) => {
            const { isListening, transcript, startListening, stopListening } = useVoiceRecognition();
            const [messages, setMessages] = useState([
              {
                type: 'assistant',
                content: 'Welcome to Your Smart Farming Assistant. Get AI-powered insights to
maximize your crop yield and profits!',
                timestamp: new Date()
              }
            ]);
            const [isProcessing, setIsProcessing] = useState(false);

            const handleSendMessage = (message) => {
              if (!message.trim()) return;

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

              setMessages(prev => [...prev, userMessage]);
              setIsProcessing(true);

              // Simulate AI response
              setTimeout(() => {
                const responses = [
                  'आपकी फसल के लिए यह एक बहुत अच्छा समय है। मौसम अनुकूल है और बाजार की स्थिति
भी अच्छी है।',
                  'वर्तमान मौसम की स्थिति के अनुसार, आपको सिंचाई की आवृत्ति बढ़ानी चाहिए।',
                  'बाजार की जानकारी के अनुसार, चावल की कीमतें अगले महीने 8-10% बढ़ सकती हैं।',
                  'आपकी फसल में कोई रोग के लक्षण नजर नहीं आ रहे। नियमित देखभाल जारी रखें।'
```

```
            ];

            const aiResponse = {
                type: 'assistant',
                content: responses[Math.floor(Math.random() * responses.length)],
                timestamp: new Date()
            };

            setMessages(prev => [...prev, aiResponse]);
            setIsProcessing(false);
        }, 2000);
    };

    useEffect(() => {
        if (transcript) {
            handleSendMessage(transcript);
        }
    }, [transcript]);

    return (
        <div className="container mx-auto px-4 py-8 space-y-8 slide-in">
            <div className="back-button" onClick={() => setActiveTab('dashboard')}>
                <i className="fas fa-arrow-left"></i>
                <span>Back to Dashboard</span>
            </div>

            <div className="text-center">
                <h2 className="text-2xl md:text-3xl font-bold text-gray-800 mb-2">
                    Voice AI Assistant
                </h2>
                <p className="text-gray-600 max-w-2xl mx-auto">
                    Speak naturally in Hindi, English, or Telugu - Your AI farming expert
understands them all
                </p>
            </div>

            <div className="bg-white rounded-2xl shadow-sm p-6 max-w-4xl mx-auto">
                {/* Chat Messages */}
                <div className="h-96 overflow-y-auto mb-6 p-4 bg-gray-50 rounded-lg">
                    <div className="space-y-4">
                        {messages.map((message, index) => (
                            <div
                                key={index}
                                className={`flex ${message.type === 'user' ? 'justify-end' :
'justify-start'}`}
                            >
                                <div
                                    className={`max-w-xs lg:max-w-md px-4 py-3 rounded-2xl ${
```

```
                                message.type === 'user'
                                    ? 'bg-blue-500 text-white'
                                    : 'bg-gray-200 text-gray-800'
                                }`}
                            >
                                <p className="text-sm">{message.content}</p>
                                <p className="text-xs opacity-70 mt-1">
                                    {message.timestamp.toLocaleTimeString()}
                                </p>
                            </div>
                        </div>
                    ))}

                    {isProcessing && (
                        <div className="flex justify-start">
                            <div className="bg-gray-200 text-gray-800 px-4 py-3 rounded-2xl">
                                <div className="flex items-center space-x-2">
                                    <div className="w-2 h-2 bg-gray-600 rounded-full
animate-pulse"></div>
                                    <div className="w-2 h-2 bg-gray-600 rounded-full
animate-pulse" style={{ animationDelay: '0.2s' }}></div>
                                    <div className="w-2 h-2 bg-gray-600 rounded-full
animate-pulse" style={{ animationDelay: '0.4s' }}></div>
                                </div>
                            </div>
                        </div>
                    )}
                </div>
            </div>

            {/* Voice Controls */}
            <div className="flex flex-col items-center space-y-4">
                <button
                    onClick={isListening ? stopListening : startListening}
                    className={`w-16 h-16 rounded-full flex items-center justify-center
transition-all ${
                        isListening
                            ? 'bg-red-500 hover:bg-red-600 animate-pulse'
                            : 'bg-green-500 hover:bg-green-600'
                    }`}
                >
                    <i className={`fas ${isListening ? 'fa-stop' : 'fa-microphone'} text-white
text-xl`}></i>
                </button>

                {isListening && (
                    <div className="flex items-center space-x-1 h-8">
                        {[1, 2, 3, 4, 5].map((i) => (
```

```
                        <div
                            key={i}
                            className="w-1 bg-green-500 rounded-full voice-wave"
                            style={{ animationDelay: `${i * 0.1}s` }}
                        ></div>
                    ))}
                </div>
            )}

            <div className="text-center">
                <p className="text-gray-600 text-sm">
                    {isListening ? 'Listening... Speak now' : 'Press the microphone button
and start speaking'}
                </p>
            </div>
        </div>
    </div>
  );
};

// Smart Recommendations Component
const SmartRecommendations = ({ setActiveTab }) => {
  const [selectedSeason, setSelectedSeason] = useState('rabi');
  const [selectedSoil, setSelectedSoil] = useState('loamy');
  const [recommendations, setRecommendations] = useState([]);
  const [selectedCrop, setSelectedCrop] = useState(null);

  const seasons = ['rabi', 'kharif', 'zaid'];
  const soilTypes = ['loamy', 'clay', 'sandy', 'red', 'black'];

  const cropRecommendations = {
    rabi: {
      loamy: [
        {
          name: 'Wheat',
          yield: '45-50 quintals/hectare',
          profit: '$45,000-60,000',
          duration: '120-150 days',
          guide: {
            planting: 'October to November',
            irrigation: '4-6 times during growth period',
            fertilizer: '120 kg N, 60 kg P205, 40 kg K20 per hectare',
            pests: 'Aphids, termites, and rust diseases',
            harvesting: 'March to April when grains are hard'
          }
        },
        {
```

```
            name: 'Barley',
            yield: '35-40 quintals/hectare',
            profit: '$35,000-45,000',
            duration: '110-140 days',
            guide: {
                planting: 'October to November',
                irrigation: '3-4 times during growth period',
                fertilizer: '80 kg N, 40 kg P205, 20 kg K20 per hectare',
                pests: 'Aphids and rust diseases',
                harvesting: 'March when grains are hard'
            }
        },
        {
            name: 'Mustard',
            yield: '15-20 quintals/hectare',
            profit: '$50,000-70,000',
            duration: '90-120 days',
            guide: {
                planting: 'October to November',
                irrigation: '2-3 times during growth period',
                fertilizer: '60 kg N, 30 kg P205, 20 kg K20 per hectare',
                pests: 'Aphids and white rust',
                harvesting: 'February to March when pods turn yellow'
            }
        }
    ],
    clay: [
        {
            name: 'Wheat',
            yield: '40-45 quintals/hectare',
            profit: '$40,000-55,000',
            duration: '120-150 days',
            guide: {
                planting: 'October to November',
                irrigation: '5-7 times during growth period',
                fertilizer: '100 kg N, 50 kg P205, 40 kg K20 per hectare',
                pests: 'Aphids, termites, and rust diseases',
                harvesting: 'March to April when grains are hard'
            }
        },
        {
            name: 'Gram',
            yield: '15-20 quintals/hectare',
            profit: '$30,000-40,000',
            duration: '100-130 days',
            guide: {
                planting: 'October to November',
                irrigation: '2-3 times during growth period',
```

```
                fertilizer: '20 kg N, 50 kg P205, 20 kg K20 per hectare',
                pests: 'Pod borer and wilt disease',
                harvesting: 'March when pods are dry'
            }
        }
    ]
},
kharif: {
    loamy: [
        {
            name: 'Rice',
            yield: '50-60 quintals/hectare',
            profit: '$50,000-70,000',
            duration: '150-180 days',
            guide: {
                planting: 'June to July',
                irrigation: 'Continuous flooding (5-7cm water depth)',
                fertilizer: '120 kg N, 60 kg P205, 60 kg K20 per hectare',
                pests: 'Stem borer, leaf folder, and blast disease',
                harvesting: 'November to December when 80% grains are ripe'
            }
        },
        {
            name: 'Maize',
            yield: '40-50 quintals/hectare',
            profit: '$45,000-60,000',
            duration: '120-150 days',
            guide: {
                planting: 'June to July',
                irrigation: '8-10 times during growth period',
                fertilizer: '150 kg N, 75 kg P205, 75 kg K20 per hectare',
                pests: 'Stem borer and leaf blight',
                harvesting: 'October when husks turn yellow'
            }
        }
    ]
}
};

useEffect(() => {
    const recs = cropRecommendations[selectedSeason]?.[selectedSoil] || [];
    setRecommendations(recs);
}, [selectedSeason, selectedSoil]);

const handleShowGuide = (crop) => {
    setSelectedCrop(crop);
};
```

```
    return (
      <div className="container mx-auto px-4 py-8 space-y-8 slide-in">
        {!selectedCrop ? (
          <>
            <div className="back-button" onClick={() => setActiveTab('dashboard')}>
              <i className="fas fa-arrow-left"></i>
              <span>Back to Dashboard</span>
            </div>

            <div className="text-center">
              <h2 className="text-2xl md:text-3xl font-bold text-gray-800 mb-2">
                Smart Crop Recommendations
              </h2>
              <p className="text-gray-600 max-w-2xl mx-auto">
                Get personalized crop suggestions based on your soil, climate, and
market conditions
              </p>
            </div>

            {/* Selection Filters */}
            <div className="bg-white rounded-2xl shadow-sm p-6">
              <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
                <div>
                  <label className="block text-gray-700 font-medium
mb-3">Season</label>
                  <div className="flex flex-wrap gap-2">
                    {seasons.map((season) => (
                      <button
                        key={season}
                        onClick={() => setSelectedSeason(season)}
                        className={`tab-button ${selectedSeason === season ?
'active' : ''}`}
                      >
                        {season.charAt(0).toUpperCase() + season.slice(1)}
                      </button>
                    ))}
                  </div>
                </div>

                <div>
                  <label className="block text-gray-700 font-medium mb-3">Soil
Type</label>
                  <div className="flex flex-wrap gap-2">
                    {soilTypes.map((soil) => (
                      <button
                        key={soil}
                        onClick={() => setSelectedSoil(soil)}
                        className={`tab-button ${selectedSoil === soil ? 'active' : ''}`}
```

```
                                >
                                    {soil.charAt(0).toUpperCase() + soil.slice(1)}
                                </button>
                            ))}
                        </div>
                    </div>
                </div>
            </div>

            {/* Recommendations */}
            {recommendations.length > 0 ? (
                <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
                    {recommendations.map((crop, index) => (
                        <div key={index} className="card bg-white rounded-2xl shadow-sm
p-6">
                            <div className="flex items-center justify-between mb-4">
                                <h3 className="text-gray-800 font-semibold
text-lg">{crop.name}</h3>
                                <div className="w-10 h-10 bg-green-100 rounded-xl flex
items-center justify-center">
                                    <i className="fas fa-seedling text-green-600"></i>
                                </div>
                            </div>

                            <div className="space-y-3">
                                <div className="bg-gray-50 rounded-lg p-3">
                                    <div className="text-gray-600 text-sm">Expected
Yield</div>
                                    <div className="text-gray-800
font-semibold">{crop.yield}</div>
                                </div>

                                <div className="bg-gray-50 rounded-lg p-3">
                                    <div className="text-gray-600 text-sm">Profit Potential</div>
                                    <div className="text-green-600
font-semibold">{crop.profit}</div>
                                </div>

                                <div className="bg-gray-50 rounded-lg p-3">
                                    <div className="text-gray-600 text-sm">Duration</div>
                                    <div className="text-gray-800
font-semibold">{crop.duration}</div>
                                </div>
                            </div>

                            <button
                                onClick={() => handleShowGuide(crop)}
                                className="btn-primary w-full mt-4 py-2 px-4 rounded-lg"
```

```
                        >
                            <i className="fas fa-info-circle mr-2"></i>
                            Detailed Guide
                        </button>
                    </div>
                ))}
            </div>
        ) : (
            <div className="bg-white rounded-2xl shadow-sm p-8 text-center">
                <div className="w-16 h-16 bg-gray-100 rounded-2xl flex items-center
justify-center mx-auto mb-4">
                    <i className="fas fa-seedling text-gray-400 text-2xl"></i>
                </div>
                <h3 className="text-gray-800 font-semibold mb-2">No
recommendations found</h3>
                <p className="text-gray-600">Try different season and soil type
combinations</p>
            </div>
        )}
    </>
) : (
    <div className="bg-white rounded-2xl shadow-sm p-6">
        <div className="back-button" onClick={() => setSelectedCrop(null)}>
            <i className="fas fa-arrow-left"></i>
            <span>Back to Recommendations</span>
        </div>

        <h3 className="text-xl font-semibold text-gray-800
mb-4">{selectedCrop.name} Detailed Guide</h3>

        <div className="space-y-4">
            <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
                <div className="bg-gray-50 rounded-lg p-4">
                    <h4 className="text-gray-800 font-medium mb-2">Planting
Time</h4>
                    <p className="text-gray-600">{selectedCrop.guide.planting}</p>
                </div>
                <div className="bg-gray-50 rounded-lg p-4">
                    <h4 className="text-gray-800 font-medium mb-2">Irrigation</h4>
                    <p className="text-gray-600">{selectedCrop.guide.irrigation}</p>
                </div>
                <div className="bg-gray-50 rounded-lg p-4">
                    <h4 className="text-gray-800 font-medium mb-2">Fertilizer
Recommendation</h4>
                    <p className="text-gray-600">{selectedCrop.guide.fertilizer}</p>
                </div>
                <div className="bg-gray-50 rounded-lg p-4">
```

```
                        <h4 className="text-gray-800 font-medium mb-2">Common
Pests/Diseases</h4>
                        <p className="text-gray-600">{selectedCrop.guide.pests}</p>
                      </div>
                    </div>

                    <div className="bg-yellow-50 rounded-lg p-4 border border-yellow-200">
                      <h4 className="text-yellow-800 font-medium mb-2">Harvesting
Time</h4>
                      <p className="text-yellow-700">{selectedCrop.guide.harvesting}</p>
                    </div>
                  </div>
                </div>
              )}
            </div>
        );
    };

    // Government Schemes Component
    const GovernmentSchemes = ({ setActiveTab }) => {
        const [selectedCategory, setSelectedCategory] = useState('all');
        const [appliedSchemes, setAppliedSchemes] = useState([]);
        const [selectedScheme, setSelectedScheme] = useState(null);

        const categories = [
            { id: 'all', name: 'All Schemes' },
            { id: 'central', name: 'Central Schemes' },
            { id: 'state', name: 'State Schemes' },
            { id: 'insurance', name: 'Insurance' },
            { id: 'credit', name: 'Credit' }
        ];

        const handleApplyScheme = (schemeId) => {
            if (!appliedSchemes.includes(schemeId)) {
                setAppliedSchemes([...appliedSchemes, schemeId]);
            }
        };

        const handleShowSchemeDetails = (scheme) => {
            setSelectedScheme(scheme);
        };

        const filteredSchemes = selectedCategory === 'all'
            ? mockSchemes
            : mockSchemes.filter(scheme => scheme.type.toLowerCase() ===
selectedCategory);

        return (
```

```
        <div className="container mx-auto px-4 py-8 space-y-8 slide-in">
          {!selectedScheme ? (
            <>
              <div className="back-button" onClick={() => setActiveTab('dashboard')}>
                <i className="fas fa-arrow-left"></i>
                <span>Back to Dashboard</span>
              </div>

              <div className="text-center">
                <h2 className="text-2xl md:text-3xl font-bold text-gray-800 mb-2">
                  Government Schemes
                </h2>
                <p className="text-gray-600 max-w-2xl mx-auto">
                  Discover and apply for government schemes tailored for farmers across
India
                </p>
              </div>

              {/* Category Filters */}
              <div className="bg-white rounded-2xl shadow-sm p-6">
                <div className="flex flex-wrap gap-2">
                  {categories.map((category) => (
                    <button
                      key={category.id}
                      onClick={() => setSelectedCategory(category.id)}
                      className={`tab-button ${selectedCategory === category.id ?
'active' : ''}`}
                    >
                      {category.name}
                    </button>
                  ))}
                </div>
              </div>

              {/* Schemes Grid */}
              <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
                {filteredSchemes.map((scheme, index) => (
                  <div key={index} className="card bg-white rounded-2xl shadow-sm
p-6">
                    <div className="flex items-center justify-between mb-4">
                      <div className="flex items-center space-x-3">
                        <div className="w-12 h-12 bg-green-100 rounded-xl flex
items-center justify-center">
                          <i className={`fas fa-${scheme.icon} text-green-600`}></i>
                        </div>
                        <div>
                          <h3 className="text-gray-800
font-semibold">{scheme.name}</h3>
```

```
                                    <p className="text-gray-500 text-sm">{scheme.type}
Government</p>
                                </div>
                            </div>
                            <div className={`px-3 py-1 rounded-full text-xs font-medium ${
                                scheme.status === 'Active' ? 'bg-green-100 text-green-800' :
'bg-red-100 text-red-800'
                            }`}>
                                {scheme.status}
                            </div>
                        </div>

                        <div className="space-y-3 mb-6">
                            <div className="bg-gray-50 rounded-lg p-3">
                                <div className="flex items-center justify-between">
                                    <span className="text-gray-600 text-sm">Benefit
Amount</span>
                                    <span className="text-gray-800
font-semibold">{scheme.amount}</span>
                                </div>
                                <div className="text-gray-500 text-xs
mt-1">{scheme.period}</div>
                            </div>

                            <div className="bg-gray-50 rounded-lg p-3">
                                <div className="flex items-center justify-between">
                                    <span className="text-gray-600 text-sm">Total
Beneficiaries</span>
                                    <span className="text-blue-600
font-semibold">{scheme.beneficiaries}</span>
                                </div>
                            </div>
                        </div>

                        <div className="space-y-2">
                            <button
                                onClick={() => handleApplyScheme(index)}
                                disabled={appliedSchemes.includes(index)}
                                className={`w-full py-2 px-4 rounded-lg font-medium transition
${
                                    appliedSchemes.includes(index)
                                        ? 'bg-green-100 text-green-800 cursor-not-allowed'
                                        : 'btn-primary'
                                }`}
                            >
                                {appliedSchemes.includes(index) ? (
                                    <>
                                        <i className="fas fa-check mr-2"></i>
```

```
                                    Applied Successfully
                                  </>
                                ) : (
                                  <>
                                    <i className="fas fa-paper-plane mr-2"></i>
                                    Apply Now
                                  </>
                                )}
                              </button>
                              <button
                                onClick={() => handleShowSchemeDetails(scheme)}
                                className="w-full bg-gray-100 text-gray-800 py-2 px-4
rounded-lg font-medium hover:bg-gray-200 transition"
                              >
                                <i className="fas fa-info-circle mr-2"></i>
                                Learn More
                              </button>
                            </div>
                          </div>
                        ))}
                      </div>
                    </>
                  ) : (
                    <div className="bg-white rounded-2xl shadow-sm p-6">
                      <div className="back-button" onClick={() => setSelectedScheme(null)}>
                        <i className="fas fa-arrow-left"></i>
                        <span>Back to Schemes</span>
                      </div>

                      <div className="flex items-center space-x-3 mb-6">
                        <div className="w-12 h-12 bg-green-100 rounded-xl flex items-center
justify-center">
                          <i className={`fas fa-${selectedScheme.icon} text-green-600`}></i>
                        </div>
                        <div>
                          <h3 className="text-xl font-semibold
text-gray-800">{selectedScheme.name}</h3>
                          <p className="text-gray-600">{selectedScheme.type} Government
Scheme</p>
                        </div>
                      </div>

                      <div className="space-y-4">
                        <div className="bg-gray-50 rounded-lg p-4">
                          <h4 className="text-gray-800 font-medium mb-2">Description</h4>
                          <p className="text-gray-600">{selectedScheme.description}</p>
                        </div>
```

```
                    <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
                      <div className="bg-gray-50 rounded-lg p-4">
                        <h4 className="text-gray-800 font-medium mb-2">Benefits</h4>
                        <p className="text-gray-600">{selectedScheme.amount}
{selectedScheme.period}</p>
                      </div>
                      <div className="bg-gray-50 rounded-lg p-4">
                        <h4 className="text-gray-800 font-medium
mb-2">Beneficiaries</h4>
                        <p className="text-gray-600">{selectedScheme.beneficiaries}
farmers</p>
                      </div>
                    </div>

                    <div className="bg-blue-50 rounded-lg p-4">
                      <h4 className="text-blue-800 font-medium mb-2">Eligibility
Criteria</h4>
                      <p className="text-blue-700">{selectedScheme.eligibility}</p>
                    </div>

                    <div className="bg-yellow-50 rounded-lg p-4">
                      <h4 className="text-yellow-800 font-medium mb-2">Required
Documents</h4>
                      <p className="text-yellow-700">{selectedScheme.documents}</p>
                    </div>
                  </div>
                </div>
              )}
            </div>
        );
    };

    // Main App Component
    const KisanAI = () => {
      const [activeTab, setActiveTab] = useState('login');
      const [isLoggedIn, setIsLoggedIn] = useState(false);
      const [farmerDetails, setFarmerDetails] = useState(null);

      const handleLoginSuccess = (details) => {
        setFarmerDetails(details);
        setIsLoggedIn(true);
        setActiveTab('dashboard');
      };

      const handleLogout = () => {
        setIsLoggedIn(false);
        setFarmerDetails(null);
        setActiveTab('login');
```

```
    };

    if (!isLoggedIn) {
      return <LoginPage onLoginSuccess={handleLoginSuccess} />;
    }

    return (
      <div className="min-h-screen bg-gray-50">
        <Navigation
          activeTab={activeTab}
          setActiveTab={setActiveTab}
          onLogout={handleLogout}
        />
        <main className="transition-all duration-500">
          {activeTab === 'dashboard' && <Dashboard setActiveTab={setActiveTab} />}
          {activeTab === 'crop-disease' && <CropDiseaseDetection
setActiveTab={setActiveTab} />}
          {activeTab === 'market' && <MarketIntelligence setActiveTab={setActiveTab} />}
          {activeTab === 'voice' && <VoiceAssistant setActiveTab={setActiveTab} />}
          {activeTab === 'recommendations' && <SmartRecommendations
setActiveTab={setActiveTab} />}
          {activeTab === 'schemes' && <GovernmentSchemes
setActiveTab={setActiveTab} />}
        </main>

        <footer className="bg-white border-t border-gray-200 py-8">
          <div className="container mx-auto px-4">
            <div className="flex flex-col md:flex-row justify-between items-center">
              <div className="flex items-center space-x-2 mb-4 md:mb-0">
                <div className="w-8 h-8 bg-green-600 rounded-lg flex items-center
justify-center">
                  <i className="fas fa-seedling text-white"></i>
                </div>
                <h3 className="text-lg font-bold text-gray-800">Kisan AI</h3>
              </div>

              <div className="flex space-x-6">
                <a href="#" className="text-gray-600 hover:text-green-600">
                  <i className="fab fa-facebook-f"></i>
                </a>
                <a href="#" className="text-gray-600 hover:text-green-600">
                  <i className="fab fa-twitter"></i>
                </a>
                <a href="#" className="text-gray-600 hover:text-green-600">
                  <i className="fab fa-instagram"></i>
                </a>
                <a href="#" className="text-gray-600 hover:text-green-600">
                  <i className="fab fa-youtube"></i>
```

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

                <div className="border-t border-gray-200 mt-6 pt-6 text-center
md:text-left">
                    <p className="text-gray-500 text-sm">
                        © 2023 Kisan AI. All rights reserved. Empowering farmers with
AI-powered insights.
                    </p>
                </div>
            </div>
        </footer>
    </div>
  );
};

// Render the app
ReactDOM.render(<KisanAI />, document.getElementById('root'));
</script>
</body>
</html>
```