import React, { useState, useEffect } from 'react';

// IMPORTANT SECURITY DISCLAIMER:

// This application is a CONCEPTUAL FRONTEND DEMONSTRATION ONLY.

// It DOES NOT implement the necessary backend security, encryption,

// audit trails, or robust authentication required for a real Electronic Health Record (EHR) system.

// DO NOT use this code with real patient data.

// A production-ready EHR requires a highly secure backend, proper database encryption,

// server-side access control, and compliance with data protection regulations (e.g., NDPR, HIPAA).

const App = () => {

  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const [username, setUsername] = useState('');

  const [password, setPassword] = useState('');

  const [userRole, setUserRole] = useState(null); // 'clinician' or 'admin'

  const [activePage, setActivePage] = useState('dashboard');

  const [selectedPatient, setSelectedPatient] = useState(null);

  const [isEditingClinicalNotes, setIsEditingClinicalNotes] = useState(false);

  // Mock Patient Data (In-memory for demonstration - NOT secure or persistent)

  const [patients, setPatients] = useState([

    {

      id: 'CVEC-001',

      fullName: 'Aisha Bello',

      dob: '1985-04-20',

      gender: 'Female',

      phone: '08012345678',

      email: 'aisha.bello@example.com',

      address: '123 Abuja Crescent, Utako',

      emergencyContact: 'Musa Bello (Husband) - 08087654321',

      visits: [

        {

          id: 'V001', date: '2023-01-15', reason: 'Routine Check-up',

          notes: {

            chiefComplaint: 'Blurred vision, occasional headaches',

            va: 'OD: 6/12, OS: 6/18',

            refraction: 'OD: -1.00DS, OS: -1.50DS',

            slitLamp: 'Clear media, healthy anterior segment',

            iop: 'OD: 15 mmHg, OS: 16 mmHg',

            diagnosis: 'Myopia',

            treatmentPlan: 'Prescribe glasses, review in 1 year.',

          },

          prescriptions: [{ date: '2023-01-15', type: 'Glasses', details: 'OD: -1.00, OS: -1.50' }],

          documents: [{ name: 'RetinalScan\_20230115.pdf', url: '#' }]

        },

        {

          id: 'V002', date: '2024-02-28', reason: 'Follow-up for headaches',

          notes: {

            chiefComplaint: 'Headaches reduced with glasses.',

            va: 'OD: 6/6 with correction, OS: 6/6 with correction',

            refraction: 'No change',

            slitLamp: 'Normal',

            iop: 'OD: 14 mmHg, OS: 15 mmHg',

            diagnosis: 'Stable Myopia',

            treatmentPlan: 'Continue current glasses, advise ergonomic setup for work.',

          },

          prescriptions: [],

          documents: []

        },

      ],

    },

    {

      id: 'CVEC-002',

      fullName: 'Chike Okoro',

      dob: '1970-11-01',

      gender: 'Male',

      phone: '07098765432',

      email: 'chike.okoro@example.com',

      address: '456 Maitama Street, Abuja',

      emergencyContact: 'Ngozi Okoro (Wife) - 08034567890',

      visits: [

        {

          id: 'V003', date: '2023-05-10', reason: 'Cataract evaluation',

          notes: {

            chiefComplaint: 'Blurry vision, difficulty driving at night.',

            va: 'OD: 6/24, OS: 6/36',

            refraction: 'OD: +1.00DS, OS: +2.00DS',

            slitLamp: 'Dense nuclear sclerosis OU',

            iop: 'OD: 18 mmHg, OS: 19 mmHg',

            diagnosis: 'Bilateral Cataracts',

            treatmentPlan: 'Discuss surgical options.',

          },

          prescriptions: [],

          documents: [{ name: 'CataractScan\_20230510.jpg', url: '#' }]

        },

      ],

    },

    {

      id: 'CVEC-003',

      fullName: 'Fatima Abdullahi',

      dob: '2015-08-12',

      gender: 'Female',

      phone: '09011223344',

      email: 'fatima.a@example.com',

      address: '789 Garki Road, Abuja',

      emergencyContact: 'Mr. Abdullahi (Father) - 08055443322',

      visits: [

        {

          id: 'V004', date: '2024-01-22', reason: 'School vision screening follow-up',

          notes: {

            chiefComplaint: 'Difficulty seeing blackboard at school.',

            va: 'OD: 6/9, OS: 6/12',

            refraction: 'OD: -0.75DS, OS: -1.00DS',

            slitLamp: 'Normal',

            iop: 'OD: 12 mmHg, OS: 11 mmHg',

            diagnosis: 'Mild Myopia',

            treatmentPlan: 'Prescribe glasses for school, review in 6 months.',

          },

          prescriptions: [{ date: '2024-01-22', type: 'Glasses', details: 'OD: -0.75, OS: -1.00' }],

          documents: []

        },

      ],

    },

  ]);

  // Handle Login

  const handleLogin = (e) => {

    e.preventDefault();

    // In a real app: Send username/password to backend for authentication and 2FA.

    // Backend returns JWT token and user role.

    if (username === 'clinician' && password === 'clinicianpass') {

      setIsLoggedIn(true);

      setUserRole('clinician');

      setActivePage('dashboard');

      console.log("Clinician logged in. (Simulated)");

      // In a real app: Log successful login for audit trail.

    } else if (username === 'admin' && password === 'adminpass') {

      setIsLoggedIn(true);

      setUserRole('admin');

      setActivePage('dashboard');

      console.log("Admin logged in. (Simulated)");

      // In a real app: Log successful login for audit trail.

    } else {

      alert('Invalid credentials. Please try "clinician/clinicianpass" or "admin/adminpass".');

      // In a real app: Log failed login attempts for audit trail.

    }

  };

  // Handle Logout

  const handleLogout = () => {

    setIsLoggedIn(false);

    setUsername('');

    setPassword('');

    setUserRole(null);

    setActivePage('login');

    setSelectedPatient(null);

    console.log("User logged out. (Simulated)");

    // In a real app: Invalidate session token, log logout for audit trail.

  };

  // Handle Patient Selection

  const handleSelectPatient = (patientId) => {

    const patient = patients.find(p => p.id === patientId);

    setSelectedPatient(patient);

    setActivePage('patientRecord');

    console.log(`Patient ${patientId} selected. (Simulated view log for audit trail)`);

    // In a real app: Log patient record view by userRole and userId for audit trail.

  };

  // Handle Saving Clinical Notes (Simulated)

  const handleSaveClinicalNotes = (visitId, updatedNotes) => {

    // In a real app: This would send updatedNotes to a secure backend API.

    // The backend would validate user role, encrypt data, and save to database.

    // The backend would also create an audit log entry for the modification.

    setPatients(prevPatients => prevPatients.map(p =>

      p.id === selectedPatient.id

        ? {

            ...p,

            visits: p.visits.map(v =>

              v.id === visitId ? { ...v, notes: updatedNotes } : v

            )

          }

        : p

    ));

    setIsEditingClinicalNotes(false);

    console.log(`Clinical notes for visit ${visitId} saved. (Simulated edit log for audit trail)`);

  };

  // Handle Adding New Patient (Simulated)

  const handleAddNewPatient = (newPatientData) => {

    // In a real app: This would send newPatientData to a secure backend API.

    // The backend would validate user role, generate ID, encrypt data, and save.

    // The backend would also create an audit log entry for the creation.

    const newPatientId = `CVEC-${String(patients.length + 1).padStart(3, '0')}`;

    const newPatient = {

      id: newPatientId,

      fullName: newPatientData.fullName,

      dob: newPatientData.dob,

      gender: newPatientData.gender,

      phone: newPatientData.phone,

      email: newPatientData.email,

      address: newPatientData.address,

      emergencyContact: newPatientData.emergencyContact,

      visits: [],

    };

    setPatients(prevPatients => [...prevPatients, newPatient]);

    alert(`New patient ${newPatient.fullName} (ID: ${newPatientId}) added! (Simulated)`);

    console.log(`New patient ${newPatientId} created. (Simulated creation log for audit trail)`);

  };

  // Login Page

  const LoginPage = () => (

    <div className="flex items-center justify-center min-h-screen bg-gradient-to-br from-blue-50 to-indigo-100 p-4">

      <div className="bg-white p-8 rounded-xl shadow-2xl w-full max-w-md text-center">

        <h2 className="text-4xl font-extrabold text-blue-800 mb-6">Central Vision EHR</h2>

        <p className="text-md text-gray-600 mb-6">Secure Patient Record & Management Portal</p>

        <form onSubmit={handleLogin} className="space-y-6">

          <div>

            <label className="block text-gray-700 text-sm font-bold mb-2 text-left" htmlFor="username">

              Username:

            </label>

            <input

              type="text"

              id="username"

              className="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700 leading-tight focus:outline-none focus:ring-2 focus:ring-blue-500"

              value={username}

              onChange={(e) => setUsername(e.target.value)}

              placeholder="clinician or admin"

              required

            />

          </div>

          <div>

            <label className="block text-gray-700 text-sm font-bold mb-2 text-left" htmlFor="password">

              Password:

            </label>

            <input

              type="password"

              id="password"

              className="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700 leading-tight focus:outline-none focus:ring-2 focus:ring-blue-500"

              value={password}

              onChange={(e) => setPassword(e.target.value)}

              placeholder="clinicianpass or adminpass"

              required

            />

          </div>

          <button

            type="submit"

            className="w-full bg-blue-700 hover:bg-blue-800 text-white font-bold py-3 px-4 rounded-lg focus:outline-none focus:shadow-outline transition duration-300 ease-in-out transform hover:scale-105"

          >

            Login

          </button>

        </form>

        <p className="text-sm text-gray-500 mt-6">

          For demo: Use "clinician/clinicianpass" or "admin/adminpass".

        </p>

        <p className="text-red-600 font-bold mt-4">

          THIS IS A DEMO. DO NOT ENTER REAL PATIENT DATA.

        </p>

      </div>

    </div>

  );

  // Dashboard Page

  const DashboardPage = () => (

    <div className="p-6 bg-white rounded-xl shadow-lg mb-8">

      <h2 className="text-3xl font-bold text-blue-800 mb-6">Dashboard - {userRole === 'clinician' ? 'Clinician View' : 'Admin View'}</h2>

      {userRole === 'clinician' && (

        <div className="space-y-6">

          <div className="bg-blue-50 p-4 rounded-lg shadow-sm">

            <h3 className="text-xl font-bold text-gray-800 mb-3">Today's Appointments</h3>

            <ul className="list-disc list-inside text-gray-700">

              <li>10:00 AM - Aisha Bello (CVEC-001) - Routine Check-up</li>

              <li>11:30 AM - Chike Okoro (CVEC-002) - Cataract Follow-up</li>

              <li>02:00 PM - Fatima Abdullahi (CVEC-003) - New Patient</li>

            </ul>

          </div>

          <div className="bg-blue-50 p-4 rounded-lg shadow-sm">

            <h3 className="text-xl font-bold text-gray-800 mb-3">Recent Patients</h3>

            <ul className="list-disc list-inside text-gray-700">

              <li>Aisha Bello (CVEC-001) - Last Visit: 2024-02-28</li>

              <li>Chike Okoro (CVEC-002) - Last Visit: 2023-05-10</li>

            </ul>

          </div>

        </div>

      )}

      {userRole === 'admin' && (

        <div className="space-y-6">

          <div className="bg-blue-50 p-4 rounded-lg shadow-sm">

            <h3 className="text-xl font-bold text-gray-800 mb-3">Daily Schedule Overview</h3>

            <ul className="list-disc list-inside text-gray-700">

              <li>10:00 AM - Aisha Bello (CVEC-001) - Dr. Adebayo</li>

              <li>11:30 AM - Chike Okoro (CVEC-002) - Dr. Fatima</li>

              <li>02:00 PM - Fatima Abdullahi (CVEC-003) - Dr. Adebayo</li>

            </ul>

          </div>

          <div className="bg-blue-50 p-4 rounded-lg shadow-sm">

            <h3 className="text-xl font-bold text-gray-800 mb-3">New Booking Requests</h3>

            <p className="text-gray-700">No new requests at this time.</p>

          </div>

        </div>

      )}

      <button

        onClick={() => setActivePage('patientSearch')}

        className="mt-6 bg-teal-500 hover:bg-teal-600 text-white font-bold py-2 px-5 rounded-lg transition duration-300 ease-in-out"

      >

        Manage Patients

      </button>

    </div>

  );

  // Patient Search Page

  const PatientSearchPage = () => {

    const [searchTerm, setSearchTerm] = useState('');

    const filteredPatients = patients.filter(p =>

      p.fullName.toLowerCase().includes(searchTerm.toLowerCase()) ||

      p.id.toLowerCase().includes(searchTerm.toLowerCase()) ||

      p.phone.includes(searchTerm)

    );

    return (

      <div className="p-6 bg-white rounded-xl shadow-lg mb-8">

        <h2 className="text-3xl font-bold text-blue-800 mb-6">Patient Database</h2>

        <input

          type="text"

          placeholder="Search by Name, ID, or Phone..."

          className="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700 leading-tight focus:outline-none focus:ring-2 focus:ring-blue-500 mb-6"

          value={searchTerm}

          onChange={(e) => setSearchTerm(e.target.value)}

        />

        <button

          onClick={() => setActivePage('addNewPatient')}

          className="mb-6 bg-blue-600 hover:bg-blue-700 text-white font-bold py-2 px-5 rounded-lg transition duration-300 ease-in-out"

        >

          Add New Patient

        </button>

        <div className="overflow-x-auto rounded-lg shadow-md">

          <table className="min-w-full bg-white">

            <thead className="bg-blue-100">

              <tr>

                <th className="py-3 px-4 text-left text-sm font-semibold text-gray-700">Patient ID</th>

                <th className="py-3 px-4 text-left text-sm font-semibold text-gray-700">Full Name</th>

                <th className="py-3 px-4 text-left text-sm font-semibold text-gray-700">Phone</th>

                <th className="py-3 px-4 text-left text-sm font-semibold text-gray-700">Actions</th>

              </tr>

            </thead>

            <tbody>

              {filteredPatients.map(patient => (

                <tr key={patient.id} className="border-b border-gray-200 last:border-b-0 hover:bg-gray-50">

                  <td className="py-3 px-4 text-gray-800">{patient.id}</td>

                  <td className="py-3 px-4 text-gray-800">{patient.fullName}</td>

                  <td className="py-3 px-4 text-gray-800">{patient.phone}</td>

                  <td className="py-3 px-4">

                    <button

                      onClick={() => handleSelectPatient(patient.id)}

                      className="bg-blue-500 hover:bg-blue-600 text-white text-sm py-1 px-3 rounded-lg transition duration-300"

                    >

                      View Record

                    </button>

                  </td>

                </tr>

              ))}

            </tbody>

          </table>

        </div>

      </div>

    );

  };

  // Add New Patient Page

  const AddNewPatientPage = () => {

    const [newPatientData, setNewPatientData] = useState({

      fullName: '', dob: '', gender: '', phone: '', email: '', address: '', emergencyContact: ''

    });

    const handleChange = (e) => {

      const { id, value } = e.target;

      setNewPatientData(prev => ({ ...prev, [id]: value }));

    };

    const handleSubmit = (e) => {

      e.preventDefault();

      handleAddNewPatient(newPatientData);

      setNewPatientData({ fullName: '', dob: '', gender: '', phone: '', email: '', address: '', emergencyContact: '' }); // Clear form

      setActivePage('patientSearch');

    };

    return (

      <div className="p-6 bg-white rounded-xl shadow-lg mb-8">

        <h2 className="text-3xl font-bold text-blue-800 mb-6">Add New Patient</h2>

        <form onSubmit={handleSubmit} className="space-y-4">

          <div className="grid grid-cols-1 md:grid-cols-2 gap-4">

            <div>

              <label htmlFor="fullName" className="block text-gray-700 text-sm font-bold mb-2">Full Name:</label>

              <input type="text" id="fullName" value={newPatientData.fullName} onChange={handleChange} required className="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700" />

            </div>

            <div>

              <label htmlFor="dob" className="block text-gray-700 text-sm font-bold mb-2">Date of Birth:</label>

              <input type="date" id="dob" value={newPatientData.dob} onChange={handleChange} required className="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700" />

            </div>

            <div>

              <label htmlFor="gender" className="block text-gray-700 text-sm font-bold mb-2">Gender:</label>

              <select id="gender" value={newPatientData.gender} onChange={handleChange} required className="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700">

                <option value="">Select</option>

                <option value="Male">Male</option>

                <option value="Female">Female</option>

                <option value="Other">Other</option>

              </select>

            </div>

            <div>

              <label htmlFor="phone" className="block text-gray-700 text-sm font-bold mb-2">Phone Number:</label>

              <input type="tel" id="phone" value={newPatientData.phone} onChange={handleChange} required className="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700" />

            </div>

            <div>

              <label htmlFor="email" className="block text-gray-700 text-sm font-bold mb-2">Email:</label>

              <input type="email" id="email" value={newPatientData.email} onChange={handleChange} className="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700" />

            </div>

            <div>

              <label htmlFor="address" className="block text-gray-700 text-sm font-bold mb-2">Address:</label>

              <input type="text" id="address" value={newPatientData.address} onChange={handleChange} className="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700" />

            </div>

            <div className="md:col-span-2">

              <label htmlFor="emergencyContact" className="block text-gray-700 text-sm font-bold mb-2">Emergency Contact:</label>

              <input type="text" id="emergencyContact" value={newPatientData.emergencyContact} onChange={handleChange} className="shadow appearance-none border rounded-lg w-full py-3 px-4 text-gray-700" />

            </div>

          </div>

          <button type="submit" className="w-full bg-blue-600 hover:bg-blue-700 text-white font-bold py-3 px-4 rounded-lg">Add Patient</button>

        </form>

      </div>

    );

  };

  // Patient Record Page

  const PatientRecordPage = () => {

    if (!selectedPatient) {

      return <div className="text-center p-6 bg-white rounded-xl shadow-lg">No patient selected.</div>;

    }

    const [activeRecordTab, setActiveRecordTab] = useState('profile');

    const [currentClinicalNotes, setCurrentClinicalNotes] = useState(selectedPatient.visits[0]?.notes || {});

    const [currentVisitId, setCurrentVisitId] = useState(selectedPatient.visits[0]?.id || null);

    useEffect(() => {

      // Update notes when selected patient or visit changes

      const latestVisit = selectedPatient.visits.length > 0 ? selectedPatient.visits[0] : null;

      setCurrentClinicalNotes(latestVisit?.notes || {});

      setCurrentVisitId(latestVisit?.id || null);

      setIsEditingClinicalNotes(false); // Reset edit mode

    }, [selectedPatient]);

    const handleNotesChange = (e) => {

      const { id, value } = e.target;

      setCurrentClinicalNotes(prev => ({ ...prev, [id]: value }));

    };

    const handleVisitSelect = (visit) => {

      setCurrentClinicalNotes(visit.notes || {});

      setCurrentVisitId(visit.id);

      setIsEditingClinicalNotes(false);

    };

    return (

      <div className="p-6 bg-white rounded-xl shadow-lg mb-8">

        <h2 className="text-3xl font-bold text-blue-800 mb-4">Patient Record: {selectedPatient.fullName} ({selectedPatient.id})</h2>

        <button

          onClick={() => setActivePage('patientSearch')}

          className="mb-6 bg-gray-500 hover:bg-gray-600 text-white text-sm py-1 px-3 rounded-lg transition duration-300"

        >

          ← Back to Patient List

        </button>

        <div className="border-b border-gray-200 mb-6">

          <nav className="-mb-px flex space-x-8" aria-label="Tabs">

            <button

              onClick={() => setActiveRecordTab('profile')}

              className={`whitespace-nowrap py-3 px-1 border-b-2 font-medium text-lg ${

                activeRecordTab === 'profile' ? 'border-blue-500 text-blue-600' : 'border-transparent text-gray-500 hover:text-gray-700 hover:border-gray-300'

              }`}

            >

              Patient Profile

            </button>

            <button

              onClick={() => setActiveRecordTab('visitHistory')}

              className={`whitespace-nowrap py-3 px-1 border-b-2 font-medium text-lg ${

                activeRecordTab === 'visitHistory' ? 'border-blue-500 text-blue-600' : 'border-transparent text-gray-500 hover:text-gray-700 hover:border-gray-300'

              }`}

            >

              Visit History

            </button>

            {userRole === 'clinician' && (

              <>

                <button

                  onClick={() => setActiveRecordTab('clinicalNotes')}

                  className={`whitespace-nowrap py-3 px-1 border-b-2 font-medium text-lg ${

                    activeRecordTab === 'clinicalNotes' ? 'border-blue-500 text-blue-600' : 'border-transparent text-gray-500 hover:text-gray-700 hover:border-gray-300'

                  }`}

                >

                  Clinical Notes

                </button>

                <button

                  onClick={() => setActiveRecordTab('prescriptions')}

                  className={`whitespace-nowrap py-3 px-1 border-b-2 font-medium text-lg ${

                    activeRecordTab === 'prescriptions' ? 'border-blue-500 text-blue-600' : 'border-transparent text-gray-500 hover:text-gray-700 hover:border-gray-300'

                  }`}

                >

                  Prescriptions

                </button>

                <button

                  onClick={() => setActiveRecordTab('documents')}

                  className={`whitespace-nowrap py-3 px-1 border-b-2 font-medium text-lg ${

                    activeRecordTab === 'documents' ? 'border-blue-500 text-blue-600' : 'border-transparent text-gray-500 hover:text-gray-700 hover:border-gray-300'

                  }`}

                >

                  Documents

                </button>

              </>

            )}

          </nav>

        </div>

        {activeRecordTab === 'profile' && (

          <div className="space-y-4 text-gray-700">

            <h3 className="text-2xl font-bold text-blue-800 mb-4">Patient Profile</h3>

            <p><strong>Full Name:</strong> {selectedPatient.fullName}</p>

            <p><strong>Date of Birth:</strong> {selectedPatient.dob}</p>

            <p><strong>Gender:</strong> {selectedPatient.gender}</p>

            <p><strong>Phone:</strong> {selectedPatient.phone}</p>

            <p><strong>Email:</strong> {selectedPatient.email}</p>

            <p><strong>Address:</strong> {selectedPatient.address}</p>

            <p><strong>Emergency Contact:</strong> {selectedPatient.emergencyContact}</p>

            {/\* In a real app, there would be an "Edit Profile" button here \*/}

          </div>

        )}

        {activeRecordTab === 'visitHistory' && (

          <div className="space-y-4 text-gray-700">

            <h3 className="text-2xl font-bold text-blue-800 mb-4">Visit History</h3>

            {selectedPatient.visits.length === 0 ? (

              <p>No past visits recorded.</p>

            ) : (

              <ul className="space-y-2">

                {selectedPatient.visits.map(visit => (

                  <li key={visit.id} className="bg-blue-50 p-3 rounded-lg shadow-sm flex justify-between items-center">

                    <span><strong>{visit.date}</strong> - {visit.reason}</span>

                    {userRole === 'clinician' && (

                      <button

                        onClick={() => handleVisitSelect(visit)}

                        className="bg-blue-500 hover:bg-blue-600 text-white text-sm py-1 px-3 rounded-lg transition duration-300"

                      >

                        View Notes

                      </button>

                    )}

                  </li>

                ))}

              </ul>

            )}

            {userRole === 'clinician' && (

              <button

                onClick={() => {

                  // Simulate adding a new blank visit

                  const newVisitId = `V${String(selectedPatient.visits.length + 1).padStart(3, '0')}`;

                  const newVisit = {

                    id: newVisitId,

                    date: new Date().toISOString().split('T')[0],

                    reason: 'New Consultation',

                    notes: {

                      chiefComplaint: '', va: '', refraction: '', slitLamp: '', iop: '', diagnosis: '', treatmentPlan: ''

                    },

                    prescriptions: [],

                    documents: []

                  };

                  setPatients(prevPatients => prevPatients.map(p =>

                    p.id === selectedPatient.id

                      ? { ...p, visits: [newVisit, ...p.visits] } // Add to the beginning

                      : p

                  ));

                  setSelectedPatient(prev => ({ ...prev, visits: [newVisit, ...prev.visits] })); // Update selected patient

                  handleVisitSelect(newVisit); // Select the new visit

                  setActiveRecordTab('clinicalNotes'); // Go to clinical notes tab

                  setIsEditingClinicalNotes(true); // Automatically enable edit mode for new visit

                  console.log(`New visit ${newVisitId} added for ${selectedPatient.fullName}. (Simulated)`);

                }}

                className="mt-4 bg-teal-500 hover:bg-teal-600 text-white font-bold py-2 px-5 rounded-lg transition duration-300 ease-in-out"

              >

                Add New Visit

              </button>

            )}

          </div>

        )}

        {activeRecordTab === 'clinicalNotes' && userRole === 'clinician' && (

          <div className="space-y-4 text-gray-700">

            <h3 className="text-2xl font-bold text-blue-800 mb-4">Clinical Encounter Notes (Visit: {currentVisitId || 'N/A'})</h3>

            <p className="text-red-500 text-sm font-semibold">

              (Note: In a real EHR, all data here would be securely encrypted and saved to a backend database with audit trails for every change.)

            </p>

            <div className="flex justify-end mb-4">

              <button

                onClick={() => setIsEditingClinicalNotes(!isEditingClinicalNotes)}

                className="bg-blue-500 hover:bg-blue-600 text-white text-sm py-2 px-4 rounded-lg transition duration-300"

              >

                {isEditingClinicalNotes ? 'View Mode' : 'Edit Notes'}

              </button>

              {isEditingClinicalNotes && (

                <button

                  onClick={() => handleSaveClinicalNotes(currentVisitId, currentClinicalNotes)}

                  className="ml-2 bg-teal-500 hover:bg-teal-600 text-white text-sm py-2 px-4 rounded-lg transition duration-300"

                >

                  Save Changes

                </button>

              )}

            </div>

            <div className="space-y-3">

              <label className="block text-gray-700 text-sm font-bold mb-1">Chief Complaint:</label>

              <textarea id="chiefComplaint" value={currentClinicalNotes.chiefComplaint || ''} onChange={handleNotesChange} readOnly={!isEditingClinicalNotes} rows="2" className="shadow appearance-none border rounded-lg w-full py-2 px-3 text-gray-700 bg-gray-50"></textarea>

              <label className="block text-gray-700 text-sm font-bold mb-1">Visual Acuity (VA):</label>

              <input type="text" id="va" value={currentClinicalNotes.va || ''} onChange={handleNotesChange} readOnly={!isEditingClinicalNotes} className="shadow appearance-none border rounded-lg w-full py-2 px-3 text-gray-700 bg-gray-50" />

              <label className="block text-gray-700 text-sm font-bold mb-1">Refraction Data:</label>

              <input type="text" id="refraction" value={currentClinicalNotes.refraction || ''} onChange={handleNotesChange} readOnly={!isEditingClinicalNotes} className="shadow appearance-none border rounded-lg w-full py-2 px-3 text-gray-700 bg-gray-50" />

              <label className="block text-gray-700 text-sm font-bold mb-1">Slit Lamp Examination:</label>

              <textarea id="slitLamp" value={currentClinicalNotes.slitLamp || ''} onChange={handleNotesChange} readOnly={!isEditingClinicalNotes} rows="2" className="shadow appearance-none border rounded-lg w-full py-2 px-3 text-gray-700 bg-gray-50"></textarea>

              <label className="block text-gray-700 text-sm font-bold mb-1">Intraocular Pressure (IOP):</label>

              <input type="text" id="iop" value={currentClinicalNotes.iop || ''} onChange={handleNotesChange} readOnly={!isEditingClinicalNotes} className="shadow appearance-none border rounded-lg w-full py-2 px-3 text-gray-700 bg-gray-50" />

              <label className="block text-gray-700 text-sm font-bold mb-1">Diagnosis / Assessment:</label>

              <textarea id="diagnosis" value={currentClinicalNotes.diagnosis || ''} onChange={handleNotesChange} readOnly={!isEditingClinicalNotes} rows="3" className="shadow appearance-none border rounded-lg w-full py-2 px-3 text-gray-700 bg-gray-50"></textarea>

              <label className="block text-gray-700 text-sm font-bold mb-1">Treatment Plan:</label>

              <textarea id="treatmentPlan" value={currentClinicalNotes.treatmentPlan || ''} onChange={handleNotesChange} readOnly={!isEditingClinicalNotes} rows="3" className="shadow appearance-none border rounded-lg w-full py-2 px-3 text-gray-700 bg-gray-50"></textarea>

            </div>

          </div>

        )}

        {activeRecordTab === 'prescriptions' && userRole === 'clinician' && (

          <div className="space-y-4 text-gray-700">

            <h3 className="text-2xl font-bold text-blue-800 mb-4">Prescriptions</h3>

            {selectedPatient.visits.flatMap(v => v.prescriptions).length === 0 ? (

              <p>No prescriptions recorded.</p>

            ) : (

              <ul className="space-y-2">

                {selectedPatient.visits.flatMap(v => v.prescriptions).map((rx, idx) => (

                  <li key={idx} className="bg-blue-50 p-3 rounded-lg shadow-sm">

                    <strong>Date:</strong> {rx.date}, <strong>Type:</strong> {rx.type}, <strong>Details:</strong> {rx.details}

                  </li>

                ))}

              </ul>

            )}

            <button

              onClick={() => alert('Simulated: Add New Prescription functionality. Requires backend.')}

              className="mt-4 bg-teal-500 hover:bg-teal-600 text-white font-bold py-2 px-5 rounded-lg transition duration-300 ease-in-out"

            >

              Add New Prescription

            </button>

          </div>

        )}

        {activeRecordTab === 'documents' && userRole === 'clinician' && (

          <div className="space-y-4 text-gray-700">

            <h3 className="text-2xl font-bold text-blue-800 mb-4">Document Uploads</h3>

            {selectedPatient.visits.flatMap(v => v.documents).length === 0 ? (

              <p>No documents uploaded.</p>

            ) : (

              <ul className="space-y-2">

                {selectedPatient.visits.flatMap(v => v.documents).map((doc, idx) => (

                  <li key={idx} className="bg-blue-50 p-3 rounded-lg shadow-sm">

                    <a href={doc.url} target="\_blank" rel="noopener noreferrer" className="text-blue-600 hover:underline">

                      {doc.name}

                    </a>

                  </li>

                ))}

              </ul>

            )}

            <button

              onClick={() => alert('Simulated: Upload Document functionality. Requires secure cloud storage and backend processing.')}

              className="mt-4 bg-teal-500 hover:bg-teal-600 text-white font-bold py-2 px-5 rounded-lg transition duration-300 ease-in-out"

            >

              Upload Document

            </button>

          </div>

        )}

        {/\* Admin/Receptionist view for sensitive sections \*/}

        {(activeRecordTab === 'clinicalNotes' || activeRecordTab === 'prescriptions' || activeRecordTab === 'documents') && userRole === 'admin' && (

          <div className="p-6 bg-red-50 border border-red-300 rounded-lg text-center">

            <p className="text-red-700 font-bold text-lg">Access Denied</p>

            <p className="text-red-600 mt-2">

              As an Admin/Receptionist, you do not have permission to view or edit sensitive clinical information.

            </p>

          </div>

        )}

      </div>

    );

  };

  const renderCurrentPage = () => {

    if (!isLoggedIn) {

      return <LoginPage />;

    }

    switch (activePage) {

      case 'dashboard':

        return <DashboardPage />;

      case 'patientSearch':

        return <PatientSearchPage />;

      case 'addNewPatient':

        return <AddNewPatientPage />;

      case 'patientRecord':

        return <PatientRecordPage />;

      default:

        return <DashboardPage />;

    }

  };

  return (

    <div className="min-h-screen bg-gradient-to-br from-blue-50 to-indigo-100 font-sans text-gray-900 p-4 sm:p-6 flex flex-col items-center">

      <script src="https://cdn.tailwindcss.com"></script>

      <link href="https://fonts.googleapis.com/css2?family=Inter:wght@400;600;700&family=Poppins:wght@400;600;700&display=swap" rel="stylesheet" />

      <style>

        {`

          body {

            font-family: 'Inter', 'Poppins', sans-serif;

          }

          /\* Custom scrollbar \*/

          .overflow-x-auto::-webkit-scrollbar {

            height: 8px;

          }

          .overflow-x-auto::-webkit-scrollbar-track {

            background: #f1f1f1;

            border-radius: 10px;

          }

          .overflow-x-auto::-webkit-scrollbar-thumb {

            background: #888;

            border-radius: 10px;

          }

          .overflow-x-auto::-webkit-scrollbar-thumb:hover {

            background: #555;

          }

          /\* Custom colors based on blueprint \*/

          .bg-blue-800 { background-color: #005A9E; }

          .text-blue-800 { color: #005A9E; }

          .bg-teal-500 { background-color: #00A99D; }

          .hover\\:bg-teal-600:hover { background-color: #008C80; } /\* Slightly darker teal for hover \*/

          .text-teal-500 { color: #00A99D; }

          .bg-blue-400 { background-color: #66B2FF; } /\* Lighter blue for active tab \*/

          .text-blue-900 { color: #003D66; } /\* Darker blue for active tab text \*/

        `}

      </style>

      {isLoggedIn && (

        <header className="w-full max-w-6xl bg-white rounded-xl shadow-lg p-6 mb-6 flex flex-col sm:flex-row justify-between items-center">

          <div className="text-center sm:text-left mb-4 sm:mb-0">

            <h1 className="text-4xl font-extrabold text-blue-800">Central Vision EHR</h1>

            <p className="text-lg text-gray-600">Secure Patient Record & Management Portal</p>

          </div>

          <div className="flex items-center space-x-4">

            <span className="text-gray-700 font-semibold">Logged in as: {username} ({userRole})</span>

            <button

              onClick={handleLogout}

              className="bg-red-500 hover:bg-red-600 text-white font-bold py-2 px-4 rounded-lg text-md transition duration-300 ease-in-out"

            >

              Logout

            </button>

          </div>

        </header>

      )}

      <main className="w-full max-w-6xl flex-grow">

        {renderCurrentPage()}

      </main>

      {isLoggedIn && (

        <footer className="w-full max-w-6xl bg-blue-800 rounded-xl shadow-lg p-4 mt-6 text-center text-white text-sm">

          <p>&copy; {new Date().getFullYear()} Central Vision Clinic. All rights reserved.</p>

          <p className="mt-2">

            This is a conceptual demo. For a real EHR, robust security, backend systems, and compliance are essential.

          </p>

        </footer>

      )}

    </div>

  );

};

export default App;