import React, { useState } from 'react';

import clientLogo from './images/dreddy.png';

import yourLogo from './images/pinnacle.png';

import {

  LineChart, Line, BarChart, Bar, XAxis, YAxis, CartesianGrid, Tooltip, Legend,

  ResponsiveContainer, PieChart, Pie, Cell

} from 'recharts';

import {

  Package, TrendingUp, AlertTriangle, RefreshCw

} from 'lucide-react';

import './App.css'; // Ensure this includes necessary styling

const Card = ({ children, className = '', style = {} }) => (

  <div className={`rounded-lg shadow ${className}`} style={style}>

    {children}

  </div>

);

const CardHeader = ({ children }) => (

  <div className="p-6">

    {children}

  </div>

);

const CardTitle = ({ children }) => (

  <h3 className="text-xl font-semibold mb-4">

    {children}

  </h3>

);

const monthlyData = [

  {

    month: 'Jan',

    initialList: 279,

    finalList: 279,

    delivered: 278,

    returned: 7,

    leftOrg: 1,

    lost: 0,

    preLoss: 0

  },

  {

    month: 'Feb',

    initialList: 312,

    finalList: 312,

    delivered: 311,

    returned: 5,

    leftOrg: 2,

    lost: 0,

    preLoss: 0

  },

  {

    month: 'Mar',

    initialList: 399,

    finalList: 391,

    delivered: 389,

    returned: 8,

    leftOrg: 8,

    lost: 1,

    preLoss: 8

  },

  {

    month: 'Apr',

    initialList: 631,

    finalList: 625,

    delivered: 625,

    returned: 12,

    leftOrg: 6,

    lost: 0,

    preLoss: 6

  },

  {

    month: 'May',

    initialList: 348,

    finalList: 348,

    delivered: 348,

    returned: 11,

    leftOrg: 0,

    lost: 1,

    preLoss: 0

  },

  {

    month: 'Jun',

    initialList: 441,

    finalList: 405,

    delivered: 403,

    returned: 18,

    leftOrg: 10,

    lost: 0,

    preLoss: 36

  },

  {

    month: 'Jul',

    initialList: 398,

    finalList: 391,

    delivered: 389,

    returned: 9,

    leftOrg: 7,

    lost: 0,

    preLoss: 7

  },

  {

    month: 'Aug',

    initialList: 536,

    finalList: 533,

    delivered: 531,

    returned: 10,

    leftOrg: 2,

    lost: 0,

    preLoss: 3

  },

  {

    month: 'Sep',

    initialList: 410,

    finalList: 409,

    delivered: 408,

    returned: 4,

    leftOrg: 1,

    lost: 0,

    preLoss: 1

  },

  {

    month: 'Oct',

    initialList: 274,

    finalList: 276,

    delivered: 273,

    returned: 2,

    leftOrg: 0,

    lost: 0,

    preLoss: -2

  },

  {

    month: 'Nov',

    initialList: 300,

    finalList: 295,

    delivered: 287,

    returned: 8,

    leftOrg: 5,

    lost: 0,

    preLoss: 5

  }

];

const COLORS = {

  primary: '#007BFF',

  success: '#28A745',

  warning: '#FFC107',

  danger: '#DC3545',

  info: '#17A2B8',

  backgroundKPI: '#F8F9FA',

};

function App() {

  const [selectedMonth, setSelectedMonth] = useState('All Months');

  const handleMonthChange = (event) => {

    setSelectedMonth(event.target.value);

  };

  // Filter data based on selected month

  const filteredData = selectedMonth === 'All Months'

    ? monthlyData

    : monthlyData.filter(data => data.month === selectedMonth);

  // Recalculate totals based on filtered data

  const totalInitialList = filteredData.reduce((sum, data) => sum + data.initialList, 0);

  const totalFinalList = filteredData.reduce((sum, data) => sum + data.finalList, 0);

  const totalDelivered = filteredData.reduce((sum, data) => sum + data.delivered, 0);

  const totalReturned = filteredData.reduce((sum, data) => sum + data.returned, 0);

  const totalLeftOrg = filteredData.reduce((sum, data) => sum + data.leftOrg, 0);

  const totalLost = filteredData.reduce((sum, data) => sum + data.lost, 0);

  //const preDispatchLoss = totalInitialList - totalFinalList;

  // Calculate rates

  const preDispatchRetentionRate = ((totalFinalList / totalInitialList) \* 100 || 0).toFixed(2);

  const deliverySuccessRate = ((totalDelivered / totalFinalList) \* 100 || 0).toFixed(2);

  const returnRate = ((totalReturned / totalFinalList) \* 100 || 0).toFixed(2);

  const lossRate = ((totalLost / totalFinalList) \* 100 || 0).toFixed(2);

  const statusDistribution = [

    { name: 'Delivered', value: totalDelivered, color: COLORS.success },

    { name: 'Returns', value: totalReturned, color: COLORS.warning },

    { name: 'Left Organization', value: totalLeftOrg, color: COLORS.info },

    { name: 'Lost/Disparity', value: totalLost, color: COLORS.danger }

  ];

  const monthOptions = ['All Months', ...monthlyData.map(data => data.month)];

  return (

    <div className="min-h-screen bg-gray-100 font-sans">

      {/\* Header \*/}

      <header className="bg-white shadow py-4 px-6 flex items-center justify-between">

        <div className="flex items-center">

          {/\* Replace 'client-logo.png' with the actual path to the client's logo \*/}

          <img src={clientLogo} alt="Check" className="h-10 mr-3" />

          <h1 className="text-2xl font-bold text-gray-800">Gift Delivery Dashboard FY 2024</h1>

        </div>

        <div className="flex items-center">

          {/\* Month Selector \*/}

          <label htmlFor="monthSelect" className="text-gray-700 font-medium mr-2">Select Month:</label>

          <select

            id="monthSelect"

            value={selectedMonth}

            onChange={handleMonthChange}

            className="border border-gray-300 rounded-md p-2 bg-white focus:outline-none focus:ring-2 focus:ring-blue-500 text-sm"

          >

            {monthOptions.map(month => (

              <option key={month} value={month}>{month}</option>

            ))}

          </select>

          {/\* Company Logo \*/}

          <span className="text-sm text-gray-500 mx-4">Powered by</span>

          {/\* Replace 'your-logo.png' with the actual path to your logo \*/}

          <img src={yourLogo} alt="Your Logo" className="h-8" />

        </div>

      </header>

      {/\* KPI Cards \*/}

      <div className="p-6">

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

          {/\* Total Initial List \*/}

          <Card style={{ backgroundColor: '#E9F7EF' }}>

            <CardHeader>

              <div className="flex items-center">

                <Package className="h-12 w-12 text-blue-500 mr-4" />

                <div>

                  <p className="text-sm font-medium text-gray-700">Total Initial List</p>

                  <p className="text-3xl font-bold text-gray-900">{totalInitialList.toLocaleString()}</p>

                  <p className="text-sm text-gray-600 mt-1">Retention: {preDispatchRetentionRate}%</p>

                </div>

              </div>

            </CardHeader>

          </Card>

          {/\* Total Final List \*/}

          <Card style={{ backgroundColor: '#D4EDDA' }}>

            <CardHeader>

              <div className="flex items-center">

                <TrendingUp className="h-12 w-12 text-green-500 mr-4" />

                <div>

                  <p className="text-sm font-medium text-gray-700">Total Final List</p>

                  <p className="text-3xl font-bold text-gray-900">{totalFinalList.toLocaleString()}</p>

                  <p className="text-sm text-gray-600 mt-1">Delivery Success: {deliverySuccessRate}%</p>

                </div>

              </div>

            </CardHeader>

          </Card>

          {/\* Total Returned \*/}

          <Card style={{ backgroundColor: '#FFF3CD' }}>

            <CardHeader>

              <div className="flex items-center">

                <RefreshCw className="h-12 w-12 text-yellow-500 mr-4" />

                <div>

                  <p className="text-sm font-medium text-gray-700">Total Returned</p>

                  <p className="text-3xl font-bold text-gray-900">{totalReturned.toLocaleString()}</p>

                  <p className="text-sm text-gray-600 mt-1">Return Rate: {returnRate}%</p>

                </div>

              </div>

            </CardHeader>

          </Card>

          {/\* Total Lost/Disparity \*/}

          <Card style={{ backgroundColor: '#F8D7DA' }}>

            <CardHeader>

              <div className="flex items-center">

                <AlertTriangle className="h-12 w-12 text-red-500 mr-4" />

                <div>

                  <p className="text-sm font-medium text-gray-700">Total Lost/Disparity</p>

                  <p className="text-3xl font-bold text-gray-900">{totalLost}</p>

                  <p className="text-sm text-gray-600 mt-1">Loss Rate: {lossRate}%</p>

                </div>

              </div>

            </CardHeader>

          </Card>

        </div>

      </div>

      {/\* Charts \*/}

      <div className="p-6 pt-0">

        <div className="grid grid-cols-1 lg:grid-cols-2 gap-6 mb-6">

          {/\* Monthly Performance \*/}

          <Card>

            <CardHeader>

              <CardTitle>Monthly Performance</CardTitle>

              <div className="h-80">

                <ResponsiveContainer width="100%" height="100%">

                  <LineChart data={filteredData}>

                    <CartesianGrid strokeDasharray="3 3" />

                    <XAxis dataKey="month" />

                    <YAxis />

                    <Tooltip />

                    <Legend />

                    <Line type="monotone" dataKey="initialList" name="Initial List" stroke={COLORS.primary} />

                    <Line type="monotone" dataKey="delivered" name="Delivered" stroke={COLORS.success} />

                    <Line type="monotone" dataKey="returned" name="Returns" stroke={COLORS.warning} />

                  </LineChart>

                </ResponsiveContainer>

              </div>

            </CardHeader>

          </Card>

          {/\* Issues Analysis \*/}

          <Card>

            <CardHeader>

              <CardTitle>Monthly Issues Analysis</CardTitle>

              <div className="h-80">

                <ResponsiveContainer width="100%" height="100%">

                  <BarChart data={filteredData}>

                    <CartesianGrid strokeDasharray="3 3" />

                    <XAxis dataKey="month" />

                    <YAxis />

                    <Tooltip />

                    <Legend />

                    <Bar dataKey="preLoss" name="Pre-dispatch Loss" fill={COLORS.info} />

                    <Bar dataKey="returned" name="Returns" fill={COLORS.warning} />

                    <Bar dataKey="leftOrg" name="Left Organization" fill={COLORS.primary} />

                    <Bar dataKey="lost" name="Lost" fill={COLORS.danger} />

                  </BarChart>

                </ResponsiveContainer>

              </div>

            </CardHeader>

          </Card>

        </div>

        {/\* Overall Distribution \*/}

        <Card>

          <CardHeader>

            <CardTitle>Overall Status Distribution</CardTitle>

            <div className="h-96">

              <ResponsiveContainer width="100%" height="100%">

                <PieChart>

                  <Pie

                    data={statusDistribution}

                    cx="50%"

                    cy="50%"

                    innerRadius={70}

                    outerRadius={100}

                    paddingAngle={5}

                    dataKey="value"

                    label={(entry) => `${entry.name}: ${((entry.value / totalFinalList) \* 100).toFixed(1)}%`}

                  >

                    {statusDistribution.map((entry, index) => (

                      <Cell key={`cell-${index}`} fill={entry.color} />

                    ))}

                  </Pie>

                  <Tooltip formatter={(value, name) => [`${value}`, `${name}`]} />

                  <Legend verticalAlign="bottom" height={36} />

                </PieChart>

              </ResponsiveContainer>

            </div>

          </CardHeader>

        </Card>

      </div>

    </div>

  );

}

export default App;