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

import { Search, Bell, User, Plane, CreditCard, MapPin, Plus, RefreshCw, TrendingUp, Calendar, DollarSign, Star } from 'lucide-react';

// Mock API service - connects to FastAPI backend in production

class LoyaltyAPI {

constructor() {

this.baseURL = 'http://localhost:8000/api';

this.token = localStorage.getItem('auth-token');

}

async request(endpoint, options = {}) {

const headers = {

'Content-Type': 'application/json',

...(this.token && { Authorization: `Bearer ${this.token}` }),

...options.headers,

};

// Simulate realistic API delay

await new Promise(resolve => setTimeout(resolve, 800 + Math.random() \* 1200));

return this.mockResponse(endpoint, options);

}

mockResponse(endpoint, options) {

if (endpoint === '/loyalty/balances') {

return {

success: true,

data: [

{

account\_id: '1',

program\_name: 'amex',

current\_balance: 80000,

last\_updated: new Date().toISOString(),

status: 'current',

metadata: { elite\_status: 'Platinum', expires: '2024-12-31' }

},

{

account\_id: '2',

program\_name: 'united',

current\_balance: 45000,

last\_updated: new Date().toISOString(),

status: 'current',

metadata: { elite\_status: 'Premier Gold', pqp: 4250 }

},

{

account\_id: '3',

program\_name: 'marriott',

current\_balance: 120000,

last\_updated: new Date().toISOString(),

status: 'current',

metadata: { elite\_status: 'Titanium', nights: 65 }

}

]

};

}

if (endpoint === '/optimize/recommendations') {

return {

success: true,

data: [

{

id: '1',

title: 'Use 50K Amex for a United flight worth $1,200',

description: 'Transfer Amex points to United MileagePlus at 1:1 ratio. Book business class LAX-NRT via ANA for exceptional value at 2.4¢ per point.',

program\_name: 'amex',

points\_required: 50000,

estimated\_value: 1200,

confidence\_score: 0.95,

booking\_url: 'https://united.com/book'

}

]

};

}

if (endpoint === '/optimize/query' && options.method === 'POST') {

const query = JSON.parse(options.body).query;

return {

success: true,

data: {

original\_query: query,

recommendations: [

{

id: 'paris-flight',

title: 'Paris Business Class - Air France',

description: 'Use 63K United miles for LAX-CDG in Air France business class. Transfer from Amex at 1:1 ratio.',

program\_name: 'united',

points\_required: 63000,

estimated\_value: 2400,

confidence\_score: 0.92

},

{

id: 'paris-hotel',

title: 'Le Meurice Paris - 7 Nights',

description: 'Luxury hotel in 1st Arrondissement using 420K Marriott points. Includes breakfast and late checkout.',

program\_name: 'marriott',

points\_required: 420000,

estimated\_value: 3500,

confidence\_score: 0.85

}

],

total\_value: 5900

}

};

}

return { success: true, data: [] };

}

async getBalances() {

const response = await this.request('/loyalty/balances');

return response.data;

}

async getRecommendations() {

const response = await this.request('/optimize/recommendations');

return response.data;

}

async searchWithQuery(query) {

const response = await this.request('/optimize/query', {

method: 'POST',

body: JSON.stringify({ query })

});

return response.data;

}

async syncAccount(accountId) {

await this.request(`/loyalty/sync/${accountId}`, { method: 'POST' });

}

}

const AceMyTravelApp = () => {

const [user] = useState({ name: "Chetan", email: "chetan@example.com" });

const [balances, setBalances] = useState([]);

const [recommendations, setRecommendations] = useState([]);

const [searchQuery, setSearchQuery] = useState("I want to go to Paris in June for 7 days");

const [searchResults, setSearchResults] = useState(null);

const [loading, setLoading] = useState({ balances: false, search: false, sync: {} });

const [activeTab, setActiveTab] = useState('dashboard');

const api = new LoyaltyAPI();

useEffect(() => {

loadInitialData();

}, []);

const loadInitialData = async () => {

setLoading(prev => ({ ...prev, balances: true }));

try {

const [balancesData, recommendationsData] = await Promise.all([

api.getBalances(),

api.getRecommendations()

]);

setBalances(balancesData);

setRecommendations(recommendationsData);

} catch (error) {

console.error('Failed to load data:', error);

}

setLoading(prev => ({ ...prev, balances: false }));

};

const handleSearch = async () => {

if (!searchQuery.trim()) return;

setLoading(prev => ({ ...prev, search: true }));

try {

const results = await api.searchWithQuery(searchQuery);

setSearchResults(results);

setActiveTab('search-results');

} catch (error) {

console.error('Search failed:', error);

}

setLoading(prev => ({ ...prev, search: false }));

};

const handleSyncAccount = async (accountId) => {

setLoading(prev => ({ ...prev, sync: { ...prev.sync, [accountId]: true } }));

try {

await api.syncAccount(accountId);

await loadInitialData();

} catch (error) {

console.error('Sync failed:', error);

}

setLoading(prev => ({ ...prev, sync: { ...prev.sync, [accountId]: false } }));

};

const formatPoints = (points) => points.toLocaleString();

return (

<div className="min-h-screen bg-gray-50">

{/\* Header - Exact match to design \*/}

<header className="bg-white shadow-sm border-b">

<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

<div className="flex justify-between items-center h-16">

{/\* Logo - Using orange brand colors from your image \*/}

<div className="flex items-center space-x-3">

<div className="w-10 h-10 bg-gradient-to-r from-orange-500 to-orange-600 rounded-lg flex items-center justify-center shadow-md">

<Plane className="w-6 h-6 text-white transform -rotate-45" />

</div>

<span className="text-2xl font-bold text-gray-900">AceMyTravel</span>

</div>

{/\* Navigation - Exact positioning \*/}

<nav className="hidden md:flex space-x-8">

<button

onClick={() => setActiveTab('dashboard')}

className={`font-medium transition-colors ${activeTab === 'dashboard' ? 'text-gray-900' : 'text-gray-500 hover:text-gray-700'}`}

>

Dashboard

</button>

<button

onClick={() => setActiveTab('search')}

className="text-gray-500 hover:text-gray-700 font-medium transition-colors"

>

Search

</button>

<button

onClick={() => setActiveTab('alerts')}

className="text-gray-500 hover:text-gray-700 font-medium transition-colors"

>

Alerts

</button>

</nav>

{/\* User Menu - Exact match \*/}

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

<button className="relative">

<Bell className="w-5 h-5 text-gray-500 hover:text-gray-700 transition-colors" />

</button>

<div className="flex items-center space-x-2">

<div className="w-8 h-8 bg-orange-500 rounded-full flex items-center justify-center">

<span className="text-white text-sm font-semibold">C</span>

</div>

<span className="font-medium text-gray-900">{user.name}</span>

</div>

</div>

</div>

</div>

</header>

{/\* Main Content \*/}

<main className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-8">

{/\* Welcome Section - Exact typography \*/}

<div className="mb-8">

<h1 className="text-4xl font-bold text-gray-900 mb-2">

Welcome, {user.name}

</h1>

</div>

{/\* Points Overview - Exact card design \*/}

<section className="mb-12">

<h2 className="text-2xl font-bold text-gray-900 mb-6">Your Points</h2>

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

{/\* AMEX Card - Exact design match \*/}

<div className="bg-white rounded-2xl p-8 shadow-sm hover:shadow-md transition-shadow">

<div className="flex items-center space-x-3 mb-6">

<div className="w-14 h-10 bg-blue-600 rounded-md flex items-center justify-center">

<span className="text-white text-xs font-bold tracking-wide">AMEX</span>

</div>

<span className="font-bold text-lg text-gray-900">AMEX</span>

<button

onClick={() => handleSyncAccount('1')}

disabled={loading.sync['1']}

className="ml-auto p-2 text-gray-400 hover:text-blue-600 transition-colors"

>

<RefreshCw className={`w-4 h-4 ${loading.sync['1'] ? 'animate-spin' : ''}`} />

</button>

</div>

<div className="text-4xl font-bold text-gray-900">

80,000

</div>

</div>

{/\* United Card \*/}

<div className="bg-white rounded-2xl p-8 shadow-sm hover:shadow-md transition-shadow">

<div className="flex items-center space-x-3 mb-6">

<div className="w-14 h-10 bg-blue-800 rounded-md flex items-center justify-center">

<div className="w-8 h-6 bg-white rounded-sm flex items-center justify-center">

<div className="w-6 h-4 bg-blue-800 rounded-sm"></div>

</div>

</div>

<span className="font-bold text-lg text-gray-900">UNITED</span>

<button

onClick={() => handleSyncAccount('2')}

disabled={loading.sync['2']}

className="ml-auto p-2 text-gray-400 hover:text-blue-600 transition-colors"

>

<RefreshCw className={`w-4 h-4 ${loading.sync['2'] ? 'animate-spin' : ''}`} />

</button>

</div>

<div className="text-4xl font-bold text-gray-900">

45,000

</div>

</div>

{/\* Marriott Card \*/}

<div className="bg-white rounded-2xl p-8 shadow-sm hover:shadow-md transition-shadow">

<div className="flex items-center space-x-3 mb-6">

<div className="w-14 h-10 bg-red-700 rounded-md flex items-center justify-center">

<span className="text-white text-lg font-bold">M</span>

</div>

<span className="font-bold text-lg text-gray-900">MARRIOTT</span>

<button

onClick={() => handleSyncAccount('3')}

disabled={loading.sync['3']}

className="ml-auto p-2 text-gray-400 hover:text-blue-600 transition-colors"

>

<RefreshCw className={`w-4 h-4 ${loading.sync['3'] ? 'animate-spin' : ''}`} />

</button>

</div>

<div className="text-4xl font-bold text-gray-900">

120,000

</div>

</div>

</div>

</section>

{/\* Best Use Recommendations Section - Exact layout \*/}

<div className="grid grid-cols-1 lg:grid-cols-2 gap-12 mb-12">

{/\* Recommendations \*/}

<section>

<h2 className="text-2xl font-bold text-gray-900 mb-6">Best Use Recommendations</h2>

<div className="bg-white rounded-2xl p-8 shadow-sm">

{recommendations.map((rec) => (

<div key={rec.id} className="mb-6 last:mb-0">

<h3 className="text-2xl font-bold text-gray-900 mb-4 leading-tight">

{rec.title}

</h3>

<p className="text-gray-600 mb-6 text-base leading-relaxed">

{rec.description}

</p>

<button className="bg-blue-600 hover:bg-blue-700 text-white font-bold px-8 py-4 rounded-xl transition-colors text-lg">

Book Now

</button>

</div>

))}

</div>

</section>

{/\* Travel Illustration - Enhanced version \*/}

<section className="flex items-center justify-center">

<div className="bg-gradient-to-br from-blue-100 to-blue-200 rounded-2xl p-8 w-full h-80 flex items-center justify-center relative overflow-hidden">

{/\* Animated clouds \*/}

<div className="absolute top-6 left-8 w-16 h-8 bg-white rounded-full opacity-80 animate-pulse"></div>

<div className="absolute top-10 left-20 w-12 h-6 bg-white rounded-full opacity-60 animate-pulse" style={{ animationDelay: '1s' }}></div>

<div className="absolute top-8 right-16 w-10 h-5 bg-white rounded-full opacity-70 animate-pulse" style={{ animationDelay: '2s' }}></div>

{/\* City skyline - Exact match to design \*/}

<div className="absolute bottom-0 left-0 right-0 flex justify-center items-end space-x-1">

<div className="w-8 h-16 bg-gradient-to-t from-blue-400 to-blue-300 rounded-t"></div>

<div className="w-6 h-12 bg-gradient-to-t from-blue-500 to-blue-400 rounded-t"></div>

<div className="w-10 h-20 bg-gradient-to-t from-blue-600 to-blue-500 rounded-t"></div>

<div className="w-12 h-24 bg-gradient-to-t from-blue-700 to-blue-600 rounded-t"></div>

<div className="w-14 h-28 bg-gradient-to-t from-blue-800 to-blue-700 rounded-t"></div>

<div className="w-10 h-22 bg-gradient-to-t from-blue-600 to-blue-500 rounded-t"></div>

<div className="w-8 h-18 bg-gradient-to-t from-blue-500 to-blue-400 rounded-t"></div>

<div className="w-6 h-14 bg-gradient-to-t from-blue-400 to-blue-300 rounded-t"></div>

</div>

{/\* Airplane - Exact positioning \*/}

<div className="absolute top-8 right-8 transform rotate-12">

<Plane className="w-16 h-16 text-blue-800" />

<div className="absolute -top-1 -right-1 w-2 h-2 bg-blue-800 rounded-full animate-ping"></div>

</div>

</div>

</section>

</div>

{/\* Search Section - Exact match to design \*/}

<section className="mb-12">

<h2 className="text-2xl font-bold text-gray-900 mb-6">Search</h2>

<div className="bg-white rounded-2xl p-6 shadow-sm">

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

<div className="flex-1">

<input

type="text"

value={searchQuery}

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

placeholder="I want to go to Paris in June for 7 days"

className="w-full px-6 py-4 text-lg border border-gray-300 rounded-xl focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-transparent"

onKeyPress={(e) => e.key === 'Enter' && handleSearch()}

/>

</div>

<button

onClick={handleSearch}

disabled={loading.search || !searchQuery.trim()}

className="bg-blue-600 hover:bg-blue-700 disabled:bg-blue-400 text-white px-8 py-4 rounded-xl font-semibold transition-colors flex items-center space-x-2"

>

{loading.search ? (

<>

<div className="w-5 h-5 border-2 border-white border-t-transparent rounded-full animate-spin"></div>

<span>Searching...</span>

</>

) : (

<>

<Search className="w-5 h-5" />

<span>Search</span>

</>

)}

</button>

</div>

</div>

</section>

{/\* Search Results Section \*/}

{searchResults && (

<section className="mb-12">

<h2 className="text-2xl font-bold text-gray-900 mb-6">Search Results</h2>

<div className="bg-blue-50 rounded-xl p-6 mb-6">

<div className="flex items-center space-x-2 mb-2">

<Search className="w-5 h-5 text-blue-600" />

<h3 className="font-semibold text-blue-900">Your Query:</h3>

</div>

<p className="text-blue-800 text-lg">"{searchResults.original\_query}"</p>

</div>

<div className="grid grid-cols-1 lg:grid-cols-2 gap-8">

{searchResults.recommendations?.map((rec) => (

<div key={rec.id} className="bg-white rounded-2xl p-6 shadow-md border hover:shadow-lg transition-all">

<div className="flex justify-between items-start mb-4">

<h3 className="text-xl font-bold text-gray-900">{rec.title}</h3>

<div className="text-right">

<div className="text-2xl font-bold text-green-600">${rec.estimated\_value?.toLocaleString()}</div>

<div className="text-sm text-gray-500">est. value</div>

</div>

</div>

<p className="text-gray-600 mb-4">{rec.description}</p>

<div className="bg-gray-50 rounded-lg p-4 mb-4">

<div className="flex justify-between items-center">

<div>

<span className="text-gray-500 text-sm">Points Required:</span>

<div className="font-bold text-blue-600">{formatPoints(rec.points\_required)}</div>

</div>

<div>

<span className="text-gray-500 text-sm">Program:</span>

<div className="font-bold text-gray-900 uppercase">{rec.program\_name}</div>

</div>

</div>

</div>

<button className="w-full bg-gradient-to-r from-orange-500 to-orange-600 hover:from-orange-600 hover:to-orange-700 text-white font-bold px-6 py-3 rounded-xl transition-all">

Book This Trip

</button>

</div>

))}

</div>

{searchResults.total\_value && (

<div className="mt-8 text-center">

<div className="inline-flex items-center space-x-2 bg-green-100 text-green-800 px-6 py-3 rounded-full">

<DollarSign className="w-5 h-5" />

<span className="font-bold text-lg">Total Trip Value: ${searchResults.total\_value?.toLocaleString()}</span>

</div>

</div>

)}

</section>

)}

{/\* Loading States \*/}

{loading.balances && (

<div className="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50">

<div className="bg-white rounded-lg p-6 flex items-center space-x-3">

<div className="w-6 h-6 border-2 border-orange-500 border-t-transparent rounded-full animate-spin"></div>

<span className="text-gray-900 font-medium">Loading your points data...</span>

</div>

</div>

)}

</main>

</div>

);

};

export default AceMyTravelApp;