1. Project Setup

npx create-react-app profile-mapping-app

cd profile-mapping-app

npm install @react-google-maps/api axios

2. Sample Profiles and API Setup

// src/services/profileService.js

export const getProfiles = () => {

return [

{

id: 1,

name: 'John Doe',

description: 'Web Developer',

photo: 'https://via.placeholder.com/100',

address: '1600 Amphitheatre Parkway, Mountain View, CA',

},

{

id: 2,

name: 'Jane Smith',

description: 'Graphic Designer',

photo: 'https://via.placeholder.com/100',

address: '1 Infinite Loop, Cupertino, CA',

},

// More profiles here

];

};

3. Profile Display Component.

// src/components/ProfileCard.js

import React from 'react';

const ProfileCard = ({ profile, onSummaryClick }) => {

return (

<div className="profile-card">

<img src={profile.photo} alt={profile.name} />

<h3>{profile.name}</h3>

<p>{profile.description}</p>

<button onClick={() => onSummaryClick(profile.address)}>Show on Map</button>

</div>

);

};

export default ProfileCard;

4. Interactive Map Component

// src/components/MapComponent.js

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

import { GoogleMap, Marker, useJsApiLoader } from '@react-google-maps/api';

const containerStyle = {

width: '100%',

height: '400px',

};

const center = {

lat: 37.7749, // Default to San Francisco

lng: -122.4194,

};

const MapComponent = ({ address }) => {

const { isLoaded } = useJsApiLoader({

googleMapsApiKey: 'YOUR\_GOOGLE\_MAPS\_API\_KEY', // Replace with your API key

});

const [location, setLocation] = useState(center);

// Function to geocode address to coordinates (simplified for demo)

const geocodeAddress = (address) => {

// Simulate a geocode service with some predefined coordinates

if (address === '1600 Amphitheatre Parkway, Mountain View, CA') {

setLocation({ lat: 37.423021, lng: -122.083739 });

} else if (address === '1 Infinite Loop, Cupertino, CA') {

setLocation({ lat: 37.33182, lng: -122.03118 });

}

};

const handleLoad = useCallback(() => {

geocodeAddress(address);

}, [address]);

if (!isLoaded) return <div>Loading...</div>;

return (

<GoogleMap mapContainerStyle={containerStyle} center={location} zoom={10} onLoad={handleLoad}>

<Marker position={location} />

</GoogleMap>

);

};

export default React.memo(MapComponent);

5. Main Application Component

// src/App.js

import React, { useState } from 'react';

import { getProfiles } from './services/profileService';

import ProfileCard from './components/ProfileCard';

import MapComponent from './components/MapComponent';

const App = () => {

const [profiles] = useState(getProfiles());

const [selectedAddress, setSelectedAddress] = useState(null);

const handleSummaryClick = (address) => {

setSelectedAddress(address);

};

return (

<div className="App">

<h1>Profile List with Map Integration</h1>

<div className="profile-list">

{profiles.map((profile) => (

<ProfileCard key={profile.id} profile={profile} onSummaryClick={handleSummaryClick} />

))}

</div>

{selectedAddress && (

<div className="map-container">

<h2>Map for: {selectedAddress}</h2>

<MapComponent address={selectedAddress} />

</div>

)}

</div>

);

};

export default App;

6. Basic Styling

/\* src/App.css \*/

.App {

font-family: Arial, sans-serif;

padding: 20px;

}

.profile-list {

display: flex;

gap: 20px;

}

.profile-card {

border: 1px solid #ccc;

padding: 20px;

width: 200px;

text-align: center;

}

.profile-card img {

width: 100px;

height: 100px;

border-radius: 50%;

}

.map-container {

margin-top: 20px;

width: 100%;

height: 400px;

}