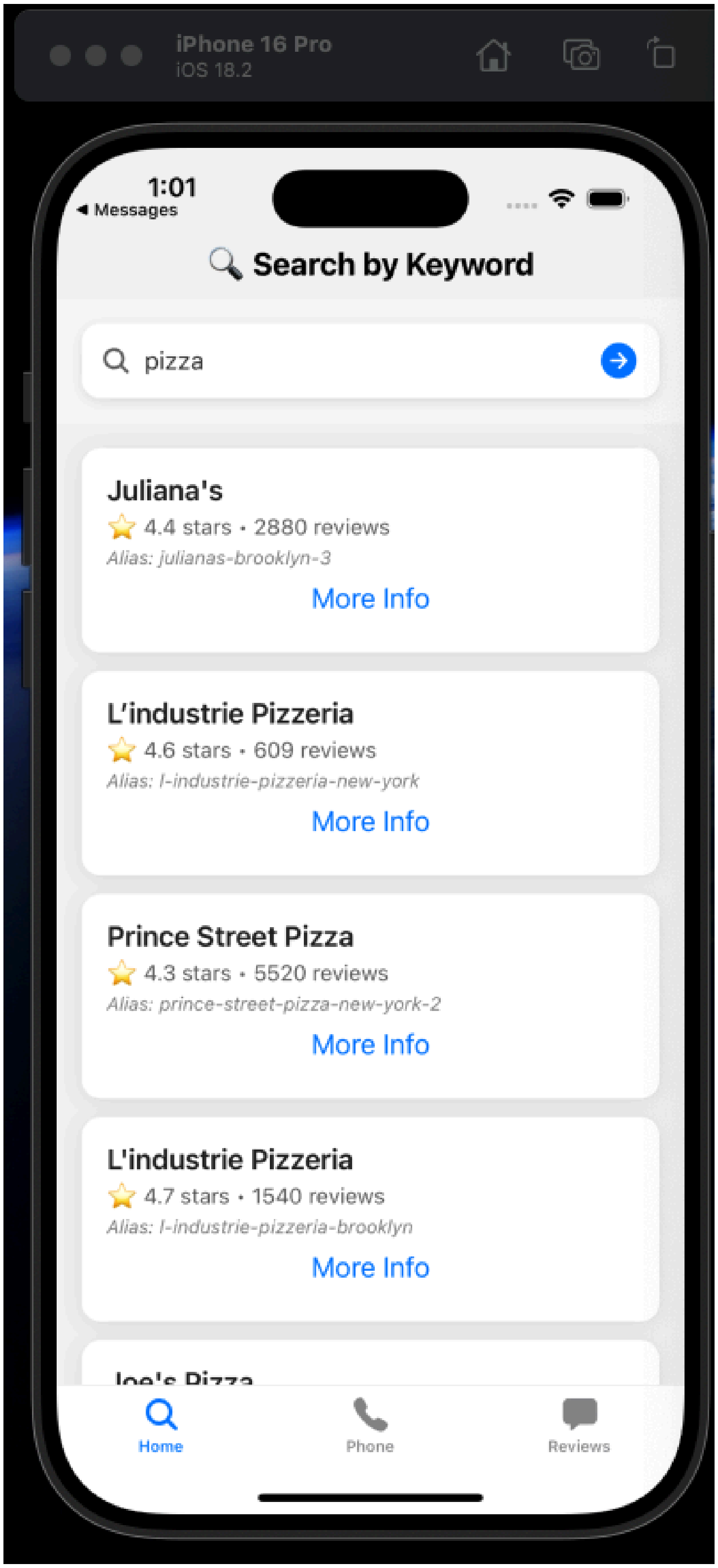
#Project 06



🔍 Search by Keyword

pizza

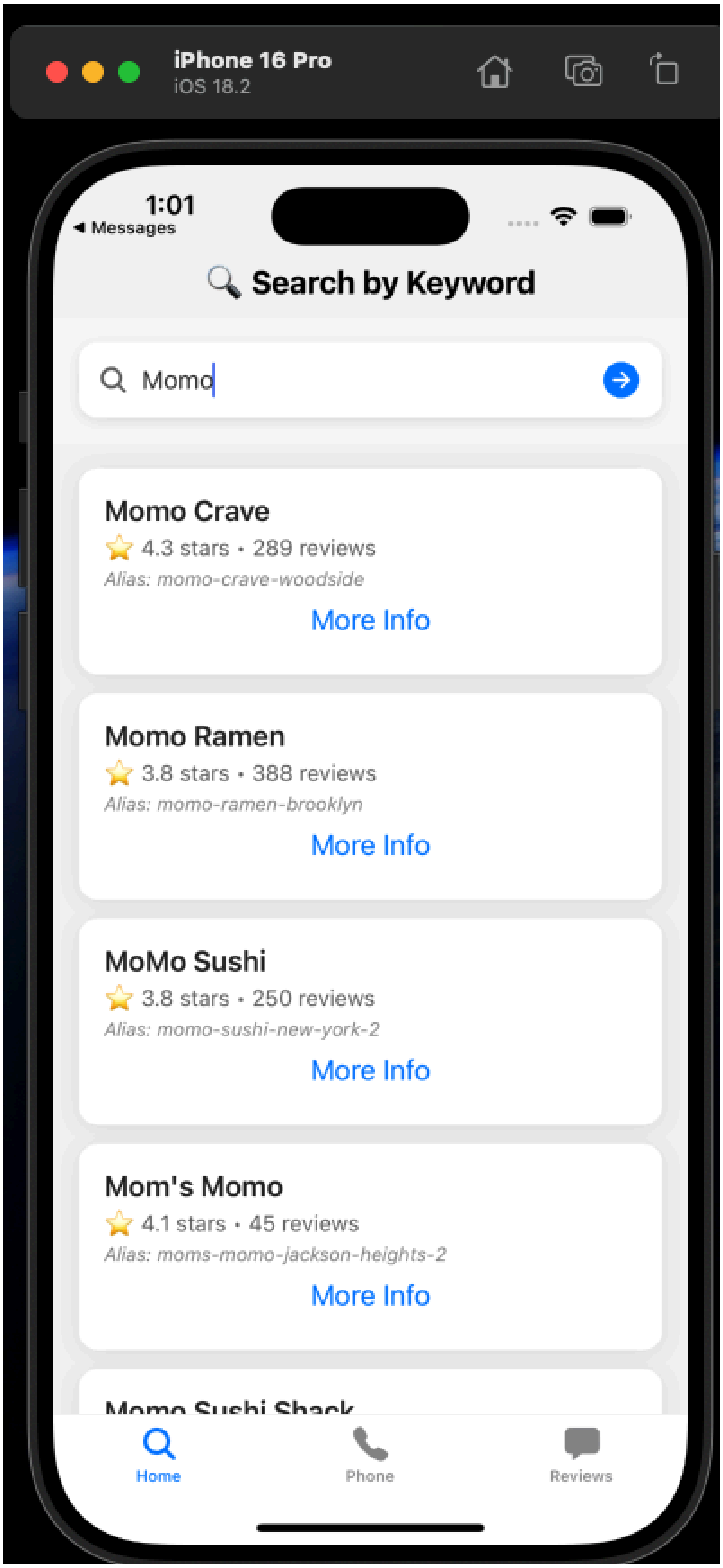
Juliana's
⭐ 4.4 stars • 2880 reviews
Alias: julianas-brooklyn-3

L'industrie Pizzeria
⭐ 4.6 stars • 609 reviews
Alias: l-industrie-pizzeria-new-york

Prince Street Pizza
⭐ 4.3 stars • 5520 reviews
Alias: prince-street-pizza-new-york-2

L'industrie Pizzeria
⭐ 4.7 stars • 1540 reviews
Alias: l-industrie-pizzeria-brooklyn

Joe's Pizza



🔍 Search by Keyword

Momo

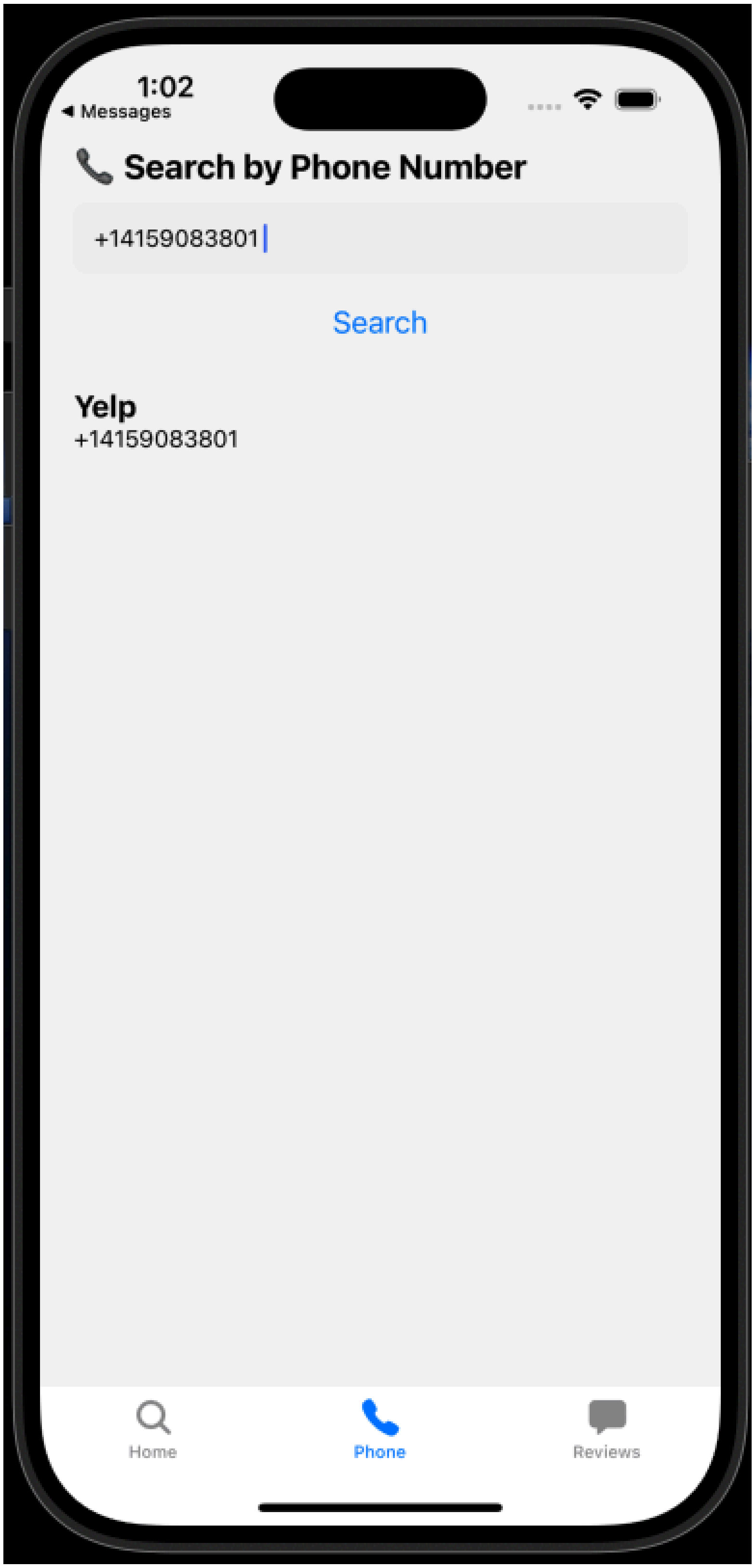
Momo Crave
⭐ 4.3 stars • 289 reviews
Alias: momo-crave-woodside
More Info

Momo Ramen
⭐ 3.8 stars • 388 reviews
Alias: momo-ramen-brooklyn
More Info

MoMo Sushi
⭐ 3.8 stars • 250 reviews
Alias: momo-sushi-new-york-2
More Info

Mom's Momo
⭐ 4.1 stars • 45 reviews
Alias: moms-momo-jackson-heights-2
More Info

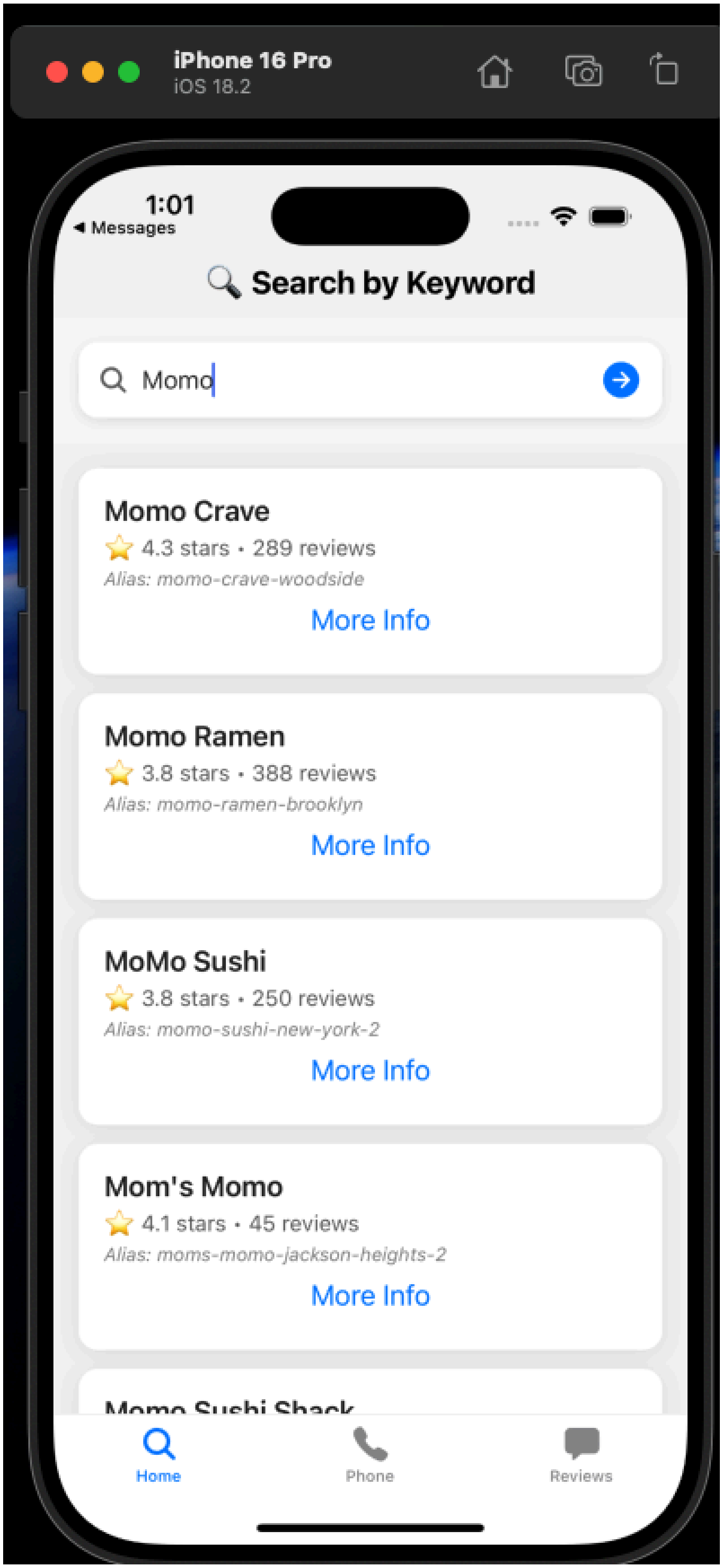
Momo Sushi Shack



📞 Search by Phone Number

+14159083801

Yelp
+14159083801



🔍 Search by Keyword

Momo

Momo Crave

⭐ 4.3 stars • 289 reviews

Alias: momo-crave-woodside

More Info

Momo Ramen

⭐ 3.8 stars • 388 reviews

Alias: momo-ramen-brooklyn

More Info

MoMo Sushi

⭐ 3.8 stars • 250 reviews

Alias: momo-sushi-new-york-2

More Info

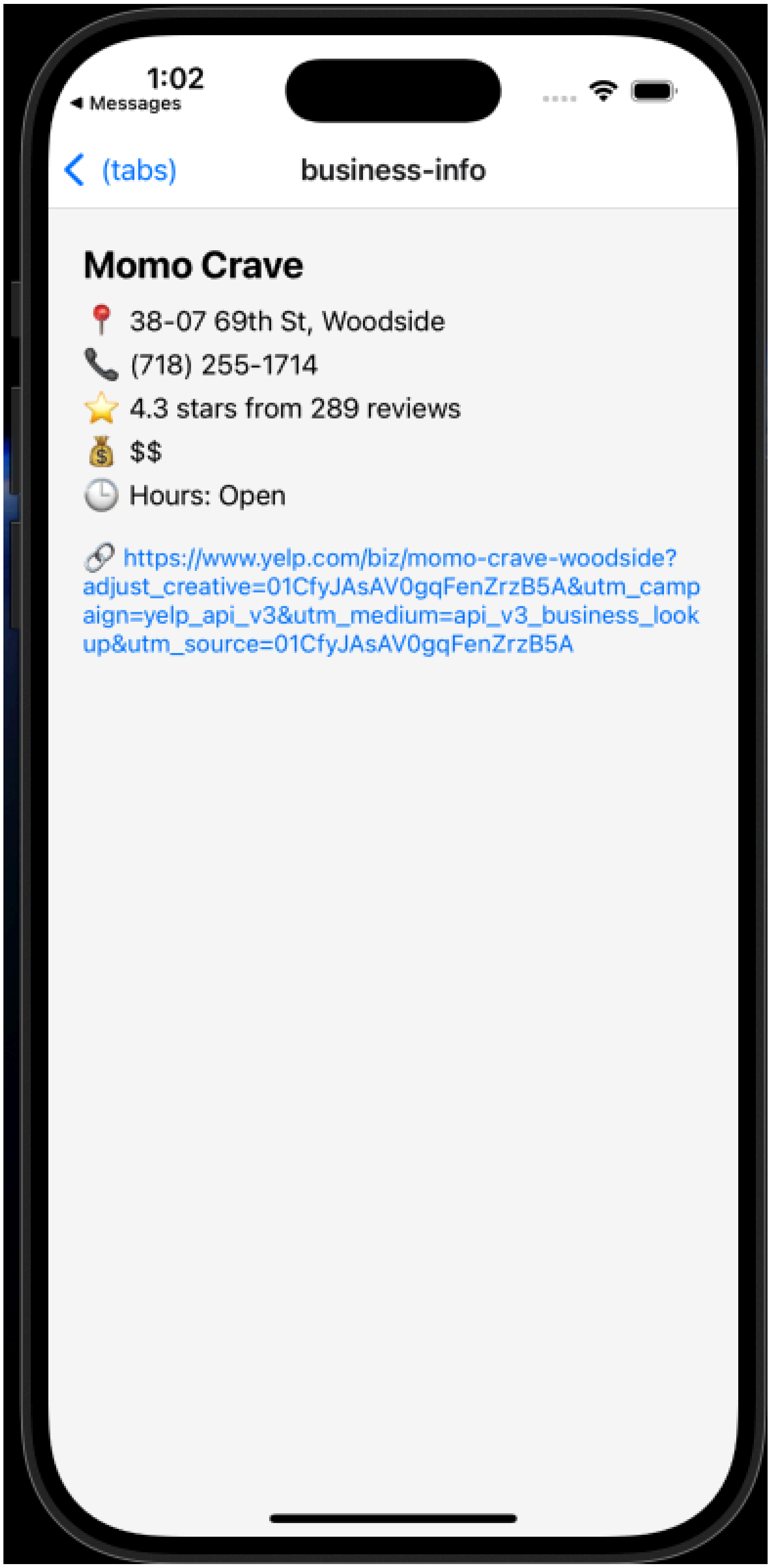
Mom's Momo

⭐ 4.1 stars • 45 reviews

Alias: moms-momo-jackson-heights-2

More Info

Momo Sushi Shack



Momo Crave

📍 38-07 69th St, Woodside

📞 (718) 255-1714

⭐ 4.3 stars from 289 reviews

💰 $$

🕒 Hours: Open

🔗 https://www.yelp.com/biz/momo-crave-woodside?adjust_creative=01CfyJAsAV0gqFenZrzB5A&utm_campaign=yelp_api_v3&utm_medium=api_v3_business_lookup&utm_source=01CfyJAsAV0gqFenZrzB5A

_layout.tsx

```
import { Slot } from 'expo-router';
import { SafeAreaProvider } from 'react-native-safe-area-context';
import BottomTabs from '@/navigation/BottomTabs';

export default function Layout() {
  return (
    <SafeAreaProvider>
      <BottomTabs />
    </SafeAreaProvider>
  );
}
```

Index.tsx

```
// app/index.tsx
import React from 'react';
import { SafeAreaView, Text, StyleSheet } from 'react-native';
import Lookup from '@/components/Lookup';
import Result from '@/components/Result';
import useAPIresponse from '@/hooks/useAPIresponse';

export default function HomeScreen() {
  const [results, searchAPI, term, setTerm] = useAPIresponse('pizza');

  return (
    <SafeAreaView style={styles.container}>
      <Text style={styles.title}>🍕 Yelp Search App</Text>
      <Lookup term={term} onTermChange={setTerm} onSearch={() => searchAPI(term)} />
      <Result results={results} />
    </SafeAreaView>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingTop: 50,
  },
  title: {
    fontSize: 20,
    fontWeight: 'bold',
    textAlign: 'center',
    marginBottom: 10,
  },
});
```

Lookup.tsx

```
// components/Lookup.tsx
import React from 'react';
import { View, TextInput, StyleSheet, Pressable } from 'react-native';
import { Ionicons } from '@expo/vector-icons';

interface Props {
  term: string;
  onTermChange: (term: string) => void;
  onSearch: () => void;
}

const Lookup: React.FC<Props> = ({ term, onTermChange, onSearch }) => {
  return (
    <View style={styles.container}>
      <View style={styles.searchBox}>
        <Ionicons name="search" size={20} color="#666" style={styles.icon} />
        <TextInput
          style={styles.input}
          placeholder="Search businesses..."
          value={term}
          onChangeText={onTermChange}
          onSubmitEditing={onSearch}
          returnKeyType="search"
        />
        <Pressable onPress={onSearch}>
          <Ionicons name="arrow-forward-circle" size={28} color="#007AFF" />
        </Pressable>
      </View>
    </View>
  );
};

export default Lookup;

const styles = StyleSheet.create({
  container: {
    padding: 16,
    backgroundColor: '#f8f8f8',
  },
  searchBox: {
    flexDirection: 'row',
    alignItems: 'center',
    backgroundColor: '#fff',
    paddingHorizontal: 12,
    borderRadius: 12,
    shadowColor: '#000',
    shadowOpacity: 0.1,
    shadowRadius: 4,
    shadowOffset: { width: 0, height: 2 },
    elevation: 2,
  },
  icon: {
    marginRight: 8,
  },
  input: {
    flex: 1,
    height: 48,
    fontSize: 16,
    color: '#333',
  },
});
```

screens/
HomeScreen.tsx

```
// screens/HomeScreen.tsx
import React from 'react';
import { SafeAreaView, Text, StyleSheet } from 'react-native';
import Lookup from '../components/Lookup';
import Result from '../components/Result';
import useAPIresponse from '../hooks/useAPIresponse';

const HomeScreen = () => {
  const [results, searchAPI, term, setTerm] = useAPIresponse('pizza');

  return (
    <SafeAreaView style={styles.container}>
      <Text style={styles.title}>🔍 Search by Keyword</Text>
      <Lookup term={term} onTermChange={setTerm} onSearch={() => searchAPI(term)} />
      <Result results={results} />
    </SafeAreaView>
  );
};
```

```
export default HomeScreen;

const styles = StyleSheet.create({
  container: { flex: 1, paddingTop: 50 },
  title: { fontSize: 20, fontWeight: 'bold', textAlign: 'center', marginBottom: 10 },
});
```

PhoneSearchScreen.tsx

```
// screens/PhoneSearchScreen.tsx
import React, { useState } from 'react';
import { View, Text, TextInput, Button, StyleSheet } from 'react-native';
import yelp from '../api/APIInfo';

const PhoneSearchScreen = () => {
  const [phone, setPhone] = useState('');
  const [result, setResult] = useState<any>(null);

  const searchByPhone = async () => {
    try {
      const response = await yelp.get('/businesses/search/phone', {
        params: { phone },
      });
      setResult(response.data.businesses[0]);
    } catch (err) {
      console.error(err);
    }
  };

  return (
    <View style={styles.container}>
      <Text style={styles.title}>📞 Search by Phone Number</Text>
      <TextInput
        style={styles.input}
        placeholder="+1XXXXXXXXXX"
        keyboardType="phone-pad"
        value={phone}
        onChangeText={setPhone}
      />
      <Button title="Search" onPress={searchByPhone} />
      {result && (
        <View style={styles.result}>
          <Text style={styles.name}>{result.name}</Text>
          <Text>{result.phone}</Text>
        </View>
      )}
    </View>
  );
};

export default PhoneSearchScreen;

const styles = StyleSheet.create({
  container: { flex: 1, padding: 20, paddingTop: 60 },
  title: { fontSize: 20, fontWeight: 'bold', marginBottom: 10 },
  input: { backgroundColor: '#eee', padding: 12, borderRadius: 8, marginBottom: 10 },
  result: { marginTop: 20 },
  name: { fontSize: 18, fontWeight: 'bold' },
});
```

ReviewsScreen.tsx(although because of authorization it didnt happen)

```
import React, { useState } from 'react';
import { View, Text, TextInput, Button, StyleSheet, FlatList } from 'react-native';
import yelp from '../api/APIInfo';

const ReviewsScreen = () => {
  const [reviews, setReviews] = useState<any[]>([]);
  const [businessId, setBusinessId] = useState('julianas-brooklyn-3'); // TEMP: hardcoded alias for testing

  const fetchReviews = async () => {
    if (!businessId) return;
    try {
      const response = await yelp.get(`/businesses/${businessId}/reviews`, {
        params: { locale: 'en_US' },
      });
      console.log('Reviews:', response.data.reviews);
      setReviews(response.data.reviews);
    } catch (err: any) {
      console.error('Review fetch error:', JSON.stringify(err.response?.data || err.message));
    }
  };


  return (
    <View style={styles.container}>
      <Text style={styles.title}>📝 Fetch Reviews</Text>
      <TextInput
        style={styles.input}
        placeholder="Enter business alias"
        value={businessId}
        onChangeText={(text) => setBusinessId(text.trim())}
      />
      <Button title="Get Reviews" onPress={fetchReviews} />
      <FlatList
        data={reviews}
        keyExtractor={(item) => item.id}
        renderItem={({ item }) => (
          <View style={styles.review}>
            <Text style={styles.user}>{item.user.name}</Text>
            <Text>{item.text}</Text>
          </View>
        )}
      />
    </View>
  );
};

export default ReviewsScreen;

const styles = StyleSheet.create({
  container: { flex: 1, padding: 20, paddingTop: 60 },
  title: { fontSize: 20, fontWeight: 'bold', marginBottom: 10 },
  input: { backgroundColor: '#eee', padding: 12, borderRadius: 8, marginBottom: 10 },
  review: { marginTop: 10, backgroundColor: '#fff', padding: 10, borderRadius: 8 },
  user: { fontWeight: 'bold' },
});
```

// hooks/useAPIresponse.ts

```
import { useEffect, useState } from 'react';
import yelp from '../api/APIInfo';

export interface Business {
  id: string;
  name: string;
  rating: number;
  review_count: number;
  alias: string;
}
```

```
const useAPIresponse = (initialTerm: string) => {
  const [results, setResults] = useState<Business[]>([]);
  const [term, setTerm] = useState<string>(initialTerm);

  const searchAPI = async (searchTerm: string) => {
    try {
      const response = await yelp.get('/businesses/search', {
        params: {
          term: searchTerm,
          location: 'New York',
          limit: 5,
        },
      });
      setResults(response.data.businesses);
      console.log('BUSINESS ID:', response.data.businesses[0].id);

    } catch (err) {
      console.error(err);
    }
  };

  useEffect(() => {
    searchAPI(term);
  }, []);

  return [results, searchAPI, term, setTerm] as const;
};

export default useAPIresponse;
```

Result.tsx

```
import React from 'react';
import { View, Text, FlatList, StyleSheet, Button } from 'react-native';
import { useRouter } from 'expo-router';
import { Business } from '../hooks/useAPIresponse';

interface Props {
  results: Business[];
}

const Result: React.FC<Props> = ({ results }) => {
  const router = useRouter();
  return (
    <FlatList
      contentContainerStyle={{ padding: 16 }}
      data={results}
      keyExtractor={(item) => item.id}
      renderItem={({ item }) => (
        <View style={styles.card}>
          <Text style={styles.name}>{item.name}</Text>
          <Text style={styles.details}>
            ⭐ {item.rating} stars • {item.review_count} reviews
          </Text>
          <Text style={styles.alias}>Alias: {item.alias}</Text>
          <Button
          title="More Info"
          onPress={() => {
         router.push({
         pathname: '/business-info',
      params: { alias: item.alias }, // ✅ this is how we "use" the screen
    });
  }}
/>


        </View>
      )}
    />
  );
};

export default Result;

const styles = StyleSheet.create({
  card: {
    backgroundColor: '#fff',
    padding: 16,
    marginBottom: 12,
    borderRadius: 12,
    shadowColor: '#000',
    shadowOpacity: 0.08,
    shadowRadius: 6,
    shadowOffset: { width: 0, height: 2 },
    elevation: 3,
  },
  name: {
    fontSize: 18,
    fontWeight: '600',
    color: '#222',
  },
  details: {
    marginTop: 4,
    fontSize: 14,
    color: '#666',
  },
  alias: {
    marginTop: 4,
    fontSize: 12,
    color: '#888',
    fontStyle: 'italic',
  },
});
```

app/(tabs)
Business-info.tsx

```
import React, { useEffect, useState } from 'react';
import { View, Text, StyleSheet, ActivityIndicator, ScrollView } from 'react-native';
import { useLocalSearchParams } from 'expo-router';
import yelp from '../api/APIInfo';

const BusinessInfoScreen = () => {
  const { alias } = useLocalSearchParams();
  const [info, setInfo] = useState<any>(null);
  const [loading, setLoading] = useState(true);

  const fetchBusinessInfo = async () => {
    try {
      const response = await yelp.get(`/businesses/${alias}`);
      setInfo(response.data);
    } catch (err: any) {
      console.error('Business info fetch error:', err.response?.data || err.message);
    } finally {
      setLoading(false);
    }
  };

  useEffect(() => {
    console.log('📦 Loading business info for alias:', alias);
    fetchBusinessInfo();
  }, []);

  if (loading) {
    return <ActivityIndicator style={{ marginTop: 60 }} />;
  }

  if (!info) {
    return (
      <View style={styles.container}>
        <Text style={styles.name}>Business info not found.</Text>
      </View>
    );
  }

  const isOpen = info?.hours?.[0]?.is_open_now;
  const address = info?.location?.address1 || 'N/A';
  const city = info?.location?.city || 'N/A';

  return (
    <ScrollView style={styles.container}>
      <Text style={styles.name}>{info.name || 'Unknown Business'}</Text>
      <Text style={styles.detail}>📍 {address}, {city}</Text>
      <Text style={styles.detail}>📞 {info.display_phone || 'N/A'}</Text>
      <Text style={styles.detail}>⭐ {info.rating ?? 'N/A'} stars from {info.review_count ?? 0}
reviews</Text>
      <Text style={styles.detail}>💰 {info.price || 'N/A'}</Text>
      <Text style={styles.detail}>🕒 Hours: {isOpen === true ? 'Open' : isOpen === false ?
'Closed' : 'N/A'}</Text>
      <Text style={styles.url}>🔗 {info.url || 'No URL'}</Text>
    </ScrollView>
  );
};

export default BusinessInfoScreen;

const styles = StyleSheet.create({
  container: { padding: 20, backgroundColor: '#f8f8f8' },
  name: { fontSize: 22, fontWeight: 'bold', marginBottom: 10 },
  detail: { fontSize: 16, marginBottom: 6 },
  url: { fontSize: 14, color: '#007AFF', marginTop: 12 },
});
```