App.jsx

```
import React from 'react';
import { NavigationContainer } from
'@react-navigation/native';
import { createBottomTabNavigator } from
'@react-navigation/bottom-tabs';
import HomeScreen from './screens/
HomeScreen';
import SettingsScreen from './screens/
SettingsScreen';
import { AppProvider } from './context/
AppContext';

const Tab = createBottomTabNavigator();

export default function App() {
  return (
    <AppProvider>
      <NavigationContainer>
        <Tab.Navigator>
```

```
          <Tab.Screen name="Home"
component={HomeScreen} />
          <Tab.Screen name="Settings"
component={SettingsScreen} />
        </Tab.Navigator>
      </NavigationContainer>
    </AppProvider>
  );
}
```

AppContext. Tsx

```
import React, { createContext, useState,
useContext } from 'react';

interface ContextProps {
  unit: 'metric' | 'imperial';
  setUnit: (val: 'metric' | 'imperial') => void;
  category: string;
  setCategory: (val: string) => void;
}
```

```
const AppContext =
createContext<ContextProps |
undefined>(undefined);

export const AppProvider: React.FC<{
children: React.ReactNode }> = ({ children
}) => {
  const [unit, setUnit] = useState<'metric' |
'imperial'>('metric');
  const [category, setCategory] =
useState('general');

  return (
    <AppContext.Provider value={{ unit,
setUnit, category, setCategory }}>
      {children}
    </AppContext.Provider>
  );
};

export const useAppContext = () =>
```

```
useContext(AppContext)!;
```

Api.tsx

```
import axios from 'axios';
import { WEATHER_API_KEY,
NEWS_API_KEY } from '../constants';

export const fetchWeather = async (lat:
number, lon: number, unit: string) => {
  const url = `https://
api.openweathermap.org/data/2.5/
onecall?lat=${lat}&lon=${lon}&units=${unit}
&exclude=minutely,hourly,alerts&appid=$
{WEATHER_API_KEY}`;
  const res = await axios.get(url);
  return res.data;
};

export const fetchNews = async (category:
string) => {
  const url = `https://newsapi.org/v2/top-
headlines?country=in&category=$
{category}&apiKey=${NEWS_API_KEY}`;
  const res = await axios.get(url);
  return res.data.articles;
};
```

Helpers.tsx

```
export const filterNewsByWeather =
(weather: string, news: any[]) => {
  if (weather.includes('cold')) {
    return news.filter(n =>
n.title.toLowerCase().includes('death') ||
n.title.toLowerCase().includes('crisis'));
  } else if (weather.includes('hot')) {
    return news.filter(n =>
n.title.toLowerCase().includes('fear') ||
n.title.toLowerCase().includes('threat'));
  } else {
    return news.filter(n =>
n.title.toLowerCase().includes('win') ||
n.title.toLowerCase().includes('happy'));
```

```
  }
};
```

HomeScreen. Tsx

```
import React, { useEffect, useState } from
'react';
import { View, Text, ScrollView,
ActivityIndicator } from 'react-native';
import * as Location from 'expo-location';
import { fetchWeather, fetchNews } from '../
utils/api';
import { useAppContext } from '../context/
AppContext';
import { filterNewsByWeather } from '../
utils/helpers';

export default function HomeScreen() {
  const { unit, category } = useAppContext();
  const [loading, setLoading] =
useState(true);
  const [weather, setWeather] =
useState<any>(null);
  const [news, setNews] =
useState<any[]>([]);

  useEffect(() => {
    (async () => {
      setLoading(true);
      const { status } = await
Location.requestForegroundPermissionsAs
ync();
      if (status !== 'granted') return;

      const loc = await
Location.getCurrentPositionAsync({});
      const w = await
fetchWeather(loc.coords.latitude,
loc.coords.longitude, unit);
      const n = await fetchNews(category);

      const filteredNews =
filterNewsByWeather(w.current.weather[0].
main.toLowerCase(), n);
      setWeather(w);
      setNews(filteredNews);
      setLoading(false);
    })();
  }, [unit, category]);

  if (loading) return <ActivityIndicator
size="large" />;

  return (
    <ScrollView style={{ padding: 16 }}>
      <Text style={{ fontSize: 22, fontWeight:
'bold' }}>
        Current: {weather.current.temp}°,
{weather.current.weather[0].main}
      </Text>
      <Text style={{ fontSize: 18, marginTop:
10 }}>5-Day Forecast:</Text>
      {weather.daily.slice(0, 5).map((day: any,
i: number) => (
        <Text key={i}>Day {i + 1}:
{day.temp.day}° {day.weather[0].main}</
Text>
      ))}
      <Text style={{ fontSize: 18, marginTop:
20 }}>News:</Text>
      {news.map((item, i) => (
        <View key={i} style={{ marginVertical:
8 }}>
          <Text style={{ fontWeight: 'bold' }}
>{item.title}</Text>
          <Text>{item.description}</Text>
        </View>
      ))}
    </ScrollView>
  );
}
```

SettingsScreen.tsx

```
import React from 'react';
import { View, Text, Picker } from 'react-
```

```
native';
import { useAppContext } from '../context/
AppContext';

export default function SettingsScreen() {
  const { unit, setUnit, category,
setCategory } = useAppContext();

  return (
    <View style={{ padding: 20 }}>
      <Text style={{ fontSize: 18 }}
>Temperature Unit:</Text>
      <Picker selectedValue={unit}
onValueChange={val => setUnit(val)}>
        <Picker.Item label="Celsius"
value="metric" />
        <Picker.Item label="Fahrenheit"
value="imperial" />
      </Picker>

      <Text style={{ fontSize: 18, marginTop:
```

```
20 }}>News Category:</Text>
      <Picker selectedValue={category}
onValueChange={val => setCategory(val)}>
        <Picker.Item label="General"
value="general" />
        <Picker.Item label="Sports"
value="sports" />
        <Picker.Item label="Technology"
value="technology" />
      </Picker>
    </View>
  );
}
```