Innovation

Below are descriptions of the implementation of extra features within the news , country and weather app.

1. News Article Saving with Ionic Storage

The app allows users to save their favourite news articles for later reading. The saved articles can be accessed directly from the homepage, and users can also remove them when desired. Additionally, each article acts as a link that redirects the user to the actual article page when clicked.

Implementation:

- **Ionic Storage:** I used Ionic Storage to store news articles locally. Each article is saved and stored in the local storage of the app. This feature allows articles from all countries to be stored in one storage space. The key passed to the dataService is 'news' which is the same for all news objects. The stored articles are retrieved when the user navigates to the saved articles page from the home page, and they are displayed in a list.

- **Save and Remove Functionality:** When a user saves an article, it is added to the storage, and the list on the homepage is updated. The user can remove an article by clicking a "Remove" button. Upon removal, the article is deleted from storage, and the list is re-rendered.
- **Link to Original Article:** Each saved article displays a clickable link that redirects the user to the actual article using its URL.

2. Use of alternate API urls to display weather data.

The weather page always displays the weather of the capital city of the country, retrieved through a custom API link using the country name.

Below is the option I used from OpenWeather API with geocoding to convert name into and country code into weather data. This means that all user input also is attached to a country code which means locations not in a particular country are not displayed and an error message is displayed.

```
https://api.openweathermap.org/data/2.5/weather?
q={city name},{country code}&appid={API key}
```

Implementation:

- **API Integration:** I used a weather API that accepts the country code and location name and returns weather data for the capital city. The weather information, such as temperature, description, and weather icon, is dynamically fetched and displayed.

3. User Location Input for Weather Search

Concept: Users can search for the weather of a location by inputting the name of the location.

Implementation:

- **Location Input:** A user inputs the name of a location (city or country) in a search field the location is passed to the API. The returned data is pushed to the weather locations array. Once this array is updated the *ngFor in the html page displays this new item.
- **Storage Handling:** When a location is added by the user , it is automatically stored in Ionic Storage with a key that combines the country code (**'weather' + {{countryCode}})**. This ensures that each country has a unique key, and weather data is stored and retrieved separately for each country.
- These locations will be rendered on the page each time the user navigates to that countries weather page with the latest weather data retrieved from the API.
- For the storage I chose to save the location via its unique id. The id is then passed to the API via the url below to display the latest weather for the location.

https://api.openweathermap.org/data/2.5/weather?id={city id}&appid={API key}

- Allows the user to save favourite locations.
- **Remove Location:** Users can remove a location from the list of displayed locations. However, the capital city's weather data cannot be removed, ensuring it remains on the weather page The removeItem method uses the dataService to delete the item from the storage array which clears the weatherLocations array and the repopulates with the saved weather items in storage.

- **Displaying Stored Locations:** When the user navigates back to the weather page, ionViewWillEnter is called to reload and display all stored weather locations, updating the UI with the latest data.

4. API Service and Environment Configuration

Concept: An API service is used to handle API requests, which abstracts the logic of fetching data from the weather and news APIs. API keys are securely stored in environment (env) files to prevent hardcoding sensitive data in the application.

Implementation:

- **API Service:** A dedicated service was created to handle API calls. This service interacts with external APIs, processes the data, and returns it to the component for rendering.
- **Environment Files:** API keys and other sensitive data are stored in environment files (environment.ts), ensuring that no hardcoded keys are present in the source code.

5. Custom Styles for Pages

Concept: I enhanced the user interface by incorporating basic styles for each page,

Custom Styles: CSS was used to apply custom styling to the pages. This includes adjusting margins, padding, fonts, and layout elements to ensure the app is visually appealing and intuitive.

Conclusion: These innovations go beyond the basic requirements by introducing advanced features like dynamic weather location management, API service abstraction, and data persistence using Ionic Storage.