Coding Challenge: Simple Countries Info App via a RESTful API 🌍

Objective:
Develop a Flutter app that interacts with the Rest Countries API to retrieve and display a list of countries. Implement functionality for searching countries by name and filtering by region. Ensure the app is well-organized with regular commits and pull requests, showcasing effective use of GitHub workflows.

Requirements:

1. **Country List Page**:
    - Fetch and display a list of all countries from the Rest Countries API.
    - Each country card in the list should display:
        - Country name
        - Flag (image)
        - Region
2. **Search Functionality**:
    - Add a search bar to allow users to search for a country by name.
3. **Country Details Page**:
    - When a user taps on a country from the list, navigate to a new page to display detailed information, including:
        - Capital
        - Population
        - Currency
        - Languages
        - Borders (other neighboring countries)
4. **Error Handling**:
    - Gracefully handle API errors, such as when a country cannot be found or if there's a network issue.
5. **Bonus Features (Optional)**:
    - **Filter**: Allow users to filter countries by region (e.g., Africa, Europe, Asia, etc.).
    - **Favorite Countries**: Let users mark countries as "favorite" and store them locally (using SQLite or Hive).

Technical Requirements:

- **State Management**: Use Provider, getX, Riverpod, or any preferred state management solution.
- **API Integration**: Use the `http` or `dio` package to make API calls.

- **Responsive UI**: Ensure the app looks good on both Ui and Ux aspects.
- **Code Quality**: Write clean, modular code with meaningful comments where necessary.

Final Submission

- Share the GitHub repository with the completed project.
- Ensure the `main` branch contains the final merged code with all features implemented.
- Submit a **build APK** in the repository or via a link (e.g., in the `Releases` section or directly in the repository).

Evaluation Criteria:

- **Functionality**: Does the app meet the core requirements (list, search, and details)?
- **Code Quality**: Is the code well-structured and easy to read?
- **Performance**: Is the app responsive and optimized, especially when handling large lists of countries?
- **UI/UX Design**: Is the app user-friendly with good design practices?
- **Error Handling**: Does the app handle API failures or edge cases gracefully?

API Documentation:

- **`Api Documentation site:`** https://restcountries.com/
- **Base URL**: https://restcountries.com/v3.1
- **Endpoints**:
    - `GET /all` - Get a list of all countries.

      `Example:`
      https://restcountries.com/v3.1/all

    - `GET /name/{name}` - Search for a country by its name.

      `Example:`
      https://restcountries.com/v3.1/name/ethiopia

- **Search by Full Name:** Search for a country by its full name, either common or official.
    - `GET /name/{name} ? fullText=true`

      `Example:`

https://restcountries.com/v3.1/name/ethiopia?fullText=true

- **Search by Currency:** Search for countries by currency code or name.
  - GET /currency/{currency}

    Example:
    https://restcountries.com/v3.1/currency/cop

For more information, visit the API documentation at https://restcountries.com/