



## Mobile App Developer Internship – Practical Assessment

### **Congratulations!**

You've advanced to the next stage of our Mobile App Developer Internship process. This task is designed to assess your mobile app development skills, attention to detail, and ability to follow UI specifications.

### **Deadline**

You have 4 Days from the date of receipt to complete the task.

Estimated Time: 6–8 hours

### **Objective**

- Build a mobile application using Flutter that allows users to browse countries from a public API, view details, add destinations to a “Bucket List”, and save personal notes locally.
- This task is designed to assess your practical Flutter development skills, including:
  - API integration
  - Local storage
  - State management
  - UI/UX design
  - Code architecture
  - Debugging & problem-solving
  - Clean code and reusability

## API Details

Base URL:

<https://api.sampleapis.com/countries/countries>

Example Response:

```
{
  "name": "Sri Lanka",
  "capital": "Colombo",
  "region": "Asia",
  "subregion": "Southern Asia",
  "population": 22100000,
  "flag": "https...",
  "coatOfArms": "https...",
  "timezones": ["UTC+05:30"],
  "borders": ["IND"],
  "languages": ["Sinhala", "Tamil"]
}
```

## Requirements

1. Splash Screen
  - a. The app shows the logo or app name for 2 seconds.
  - b. Automatically navigates to the Home Screen.
2. Home Screen – Country Explorer

Display the country list fetched from the API.

  - a. Each item must show:
    - i. Country flag image
    - ii. Country name
    - iii. Capital

b. Required Features:

- ✓ Search bar (search by country name)
- ✓ Filter by region/continent (Asia, Europe, Africa, etc.)
- ✓ Sorting: A–Z, Z–A
- ✓ Shimmer/Pulse loading effect
- ✓ Error state UI
- ✓ Pull-to-refresh

3. Country Details Screen

Show the full information of the selected country:

- a. Flag (large image)
- b. Country name
- c. Capital
- d. Region
- e. Population
- f. Timezones
- g. Languages
- h. Coat of arms (if available)

Extra Required Feature:

- ✓ Add a personal note (text input)
- ✓ Save notes locally (attached to that country)

**Notes must be editable and deletable.**

#### 4. Bucket List Screen

Users can save destinations to a personal bucket list.

Requirements:

- a. Save items in local storage (Hive / SharedPreferences)
- b. Display a list of all saved countries
- c. Remove button (remove from bucket list)
- d. Sorting options
  - i. Alphabetical
  - ii. by region

Important:

**The bucket list must be accessible offline.**

#### 5. State Management

Choose one:

- a. Riverpod
- b. Bloc
- c. Provider
- d. GetX

#### 6. Folder Structure

lib/

|              |                       |
|--------------|-----------------------|
| api/         | # API services        |
| db/          | # Local storage       |
| models/      | # Data models         |
| screens/     | # UI pages            |
| controllers/ | # State management    |
| widgets/     | # Reusable components |
| utils/       | # Constants, helpers  |

## 7. UI/UX Requirements

- ✓ Material 3 design
- ✓ Consistent padding and spacing
- ✓ Reusable widgets
- ✓ Scrollable content
- ✓ Visual hierarchy
- ✓ Clean typography
- ✓ Responsive layout for small screens

## 8. Optional (Bonus Points)

- a. Dark mode toggle
- b. Animations (fade, slide, hero)
- c. Image caching
- d. Network error retry button
- e. Offline-first design
- f. Pagination

## Deliverables

Submit the following:

1. GitHub repository link
2. README file containing:
  - i. Project description
  - ii. Instructions to run
  - iii. State management explanation
  - iv. Architecture overview
  - v. Screenshots of the app
3. APK file (Share via Google Drive; ensure the link is accessible.)

## Good Luck!

Focus on clean code and meaningful commit history.