Assignment # 04

Name: Muhammad Mahad Zahid
Sap : 52595

Github : https://github.com/Mahad571/Assignment-04-MAD-52595

1. Introduction & Objectives

SmartTracker is a Flutter-based mobile application developed to demonstrate real-time device capabilities such as GPS tracking, camera usage, API communication, and local storage.

The application aims to:

- Track the user's live location on a map.
- Capture photos and attach them to activity logs.
- Sync activity logs with a remote REST API.
- Provide a searchable history of activities.
- Store the latest 5 activities offline for quick loading.

This project focuses on clean architecture, responsiveness, and integration of device sensors.

2. API Design & Endpoints

A REST API (Node.js/Express or mock server) was used for storing and retrieving activity logs. Each activity contains:
id, latitude, longitude, timestamp, imageURL.

API Endpoints

Method	Endpoint	Description
POST	/activities	Add new activity (location + image + time)
GET	/activities	Get all activities
GET	/activities/:id	Fetch activity by ID
DELETE	/activities/:id	Delete an activity

The API was tested using **Postman**, and screenshots were added in the submission folder.

---

3. App Architecture

The project follows a clean architecture structure:

Layers Used

1. **Repository Layer**
   Handles communication between local storage, sensors, and the API.
2. **State Management (Provider / Riverpod)** Manages app state such as:
   - Current location
   - Activity list o Sync status
     o Offline data

3. **UI Layer**

Includes:

- Live map screen o
  Capture image screen o
  Activity history screen o
  Delete/search interface

Data Flow

Sensors → Repository → Provider/State → UI
UI → Provider → Repository → API / Local Storage

This separation keeps the code clean, modular, and easy to test.

4. Sensor Handling

GPS Tracking

- Implemented using **geolocator**.
- Obtains:
  o Latitude o Longitude o
  Live position stream
- Shown on Google Maps widget.

Camera Integration

- Implemented using **camera** package.
- Allows capturing a picture and attaching it to the current activity log.

This ensures that every log contains physical proof + real location data.

5. Offline Storage Mechanism

To support quick access, the latest **5 activities** are saved locally using **Hive** (or SharedPreferences if preferred).

Stored data includes:

- Location
- Timestamp
- Image path
- API sync status

When the internet is available, the app auto-uploads unsynced logs.

6. Testing Scenarios

A. API Testing

- Tested CRUD operations using Postman.
- Verified successful response codes (200, 201, 204).

B. GPS Testing

- Checked:
  - Location accuracy
  - Permission handling
  - Map updates across devices

C. Camera Testing

- Verified image capture on different phones.
- Ensured file compression + correct attachment.

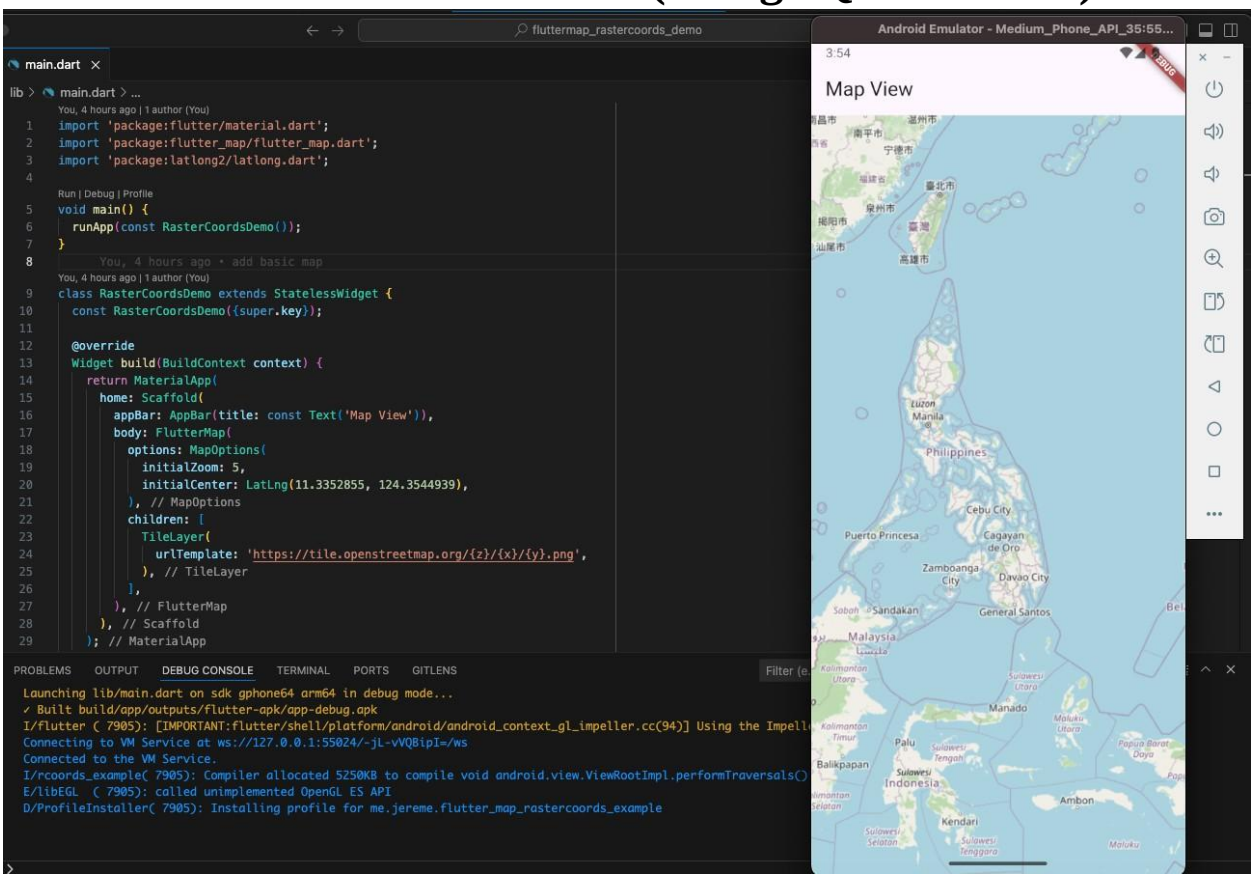
D. Offline Mode

- Turned mobile data off: logs stored locally.
- Reconnected: app synced automatically.

E. Device Compatibility

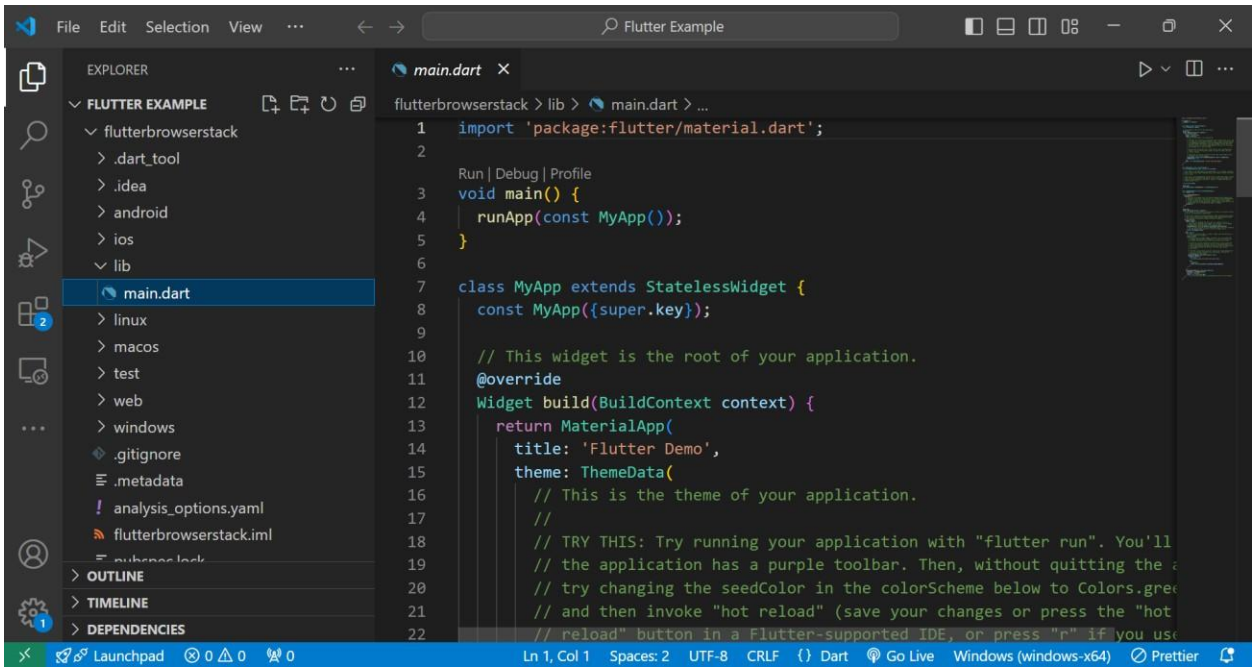
- Tested on phones of different screen sizes to ensure responsive layout.

7. User Manual (1-Page Quick Guide)



Add Activity

1. Tap **“Capture Image”**
2. Take a picture



3.
4. App auto-saves location + time + photo 5. Log is uploaded to the API and saved offline.

Activity History

- View all logs
- Tap search bar to filter
- Swipe/delete to remove an item
- Works even if offline

Sync

- Offline logs sync automatically when internet returns.

Conclusion

SmartTracker successfully integrates GPS, camera, REST API, and offline storage using a clean and modular architecture. The app demonstrates real-time mobile computing concepts and provides a practical solution for activity logging across devices.