



## **Artificial Intelligence**

**Assignment#03**

**CCP Report**

**Prepared by: Farhan-Ul-Haq (55853)**

**Submitted to: Mr. Usman**

**Class: Fall 2025 SE 5-1**

**RIPHAH INTERNATIONAL UNIVERSITY I-14 CAMPUS,  
ISLAMABAD**

## 1. Introduction

SmartTracker is a Flutter-based mobile application that allows users to track their live location, capture images, log activities, and synchronize them with a Node.js REST API backend. The system is designed to work online and offline using local storage.

## 2. Objectives

### No. Objective

- 1 Integrate GPS and Camera sensors in Flutter
- 2 Perform GET/POST through REST API
- 3 Store recent activities offline
- 4 Show real-time location on Google Maps
- 5 Display activity history using Dashboard
- 6 Test backend API using PowerShell / Postman

## 3. System Architecture (Updated)

SmartTracker uses a clean 4-layer modular architecture.

### 3.1 Architecture Layers Table

Layer	Description	Components
<b>Presentation Layer</b>	UI screens & widgets	MapScreen, AddActivityScreen, Dashboard

<b>State Management Layer</b>	Handles real-time data, API calls, camera/image state	ActivityProvider
<b>Data Layer</b>	API interaction + local storage	Repository + Hive/SharedPreferences
<b>Backend Layer</b>	Node.js REST API	server.js with GET/POST endpoints

### 3.2 Data Flow Diagram (Text Form)

User



Flutter UI



ActivityProvider



Repository



Node.js REST API (server.js)



Local Storage (Hive)



Dashboard Screen

## 4. API Design & Endpoints

### 4.1 API Endpoints Table

Endpoint	Method	Description
/activities	GET	Returns all activities
/activities	POST	Adds new activity

### 4.2 Example POST JSON

```
{  
    "latitude": 33.6844,  
    "longitude": 73.0479,  
    "imageBase64": "test123",  
    "timestamp": "2025-12-02T12:00:00",  
    "address": "Current Location"  
}
```

## 5. Sensor Handling

### 5.1 GPS Handling Table

Feature	Description
GPS Library	geolocator
Map Widget	GoogleMap

Permissions	Location permission management
Output	Latitude + Longitude + Marker updates

## 5.2 Camera Handling Table

Feature	Description
Camera Library	camera
Conversion	Image → Base64
Usage	Image shown in Add Activity screen & POST body

## 6. Offline Storage System

Storage Method	Purpose
Hive/ SharedPreferences	Cache last 5 activities
Offline Access	Dashboard shows cached activities
Auto-Load	Loads cached data when server is not reachable

## 7. App UI/UX Screens

### 7.1 App Screens Table

Screen Name	Description
Map Screen	Shows real-time GPS + Capture button

<b>Add Activity Screen</b>	Shows image + metadata + Save button
<b>Dashboard Screen</b>	Displays activity logs, search, refresh

## 8. API Testing (With Screenshot Section)

Below are the screenshot placeholders :

### 8.1 Screenshot: Server Running

Starting server.js...

SmartTracker API running on <http://localhost:3000>

```
PS C:\Sem-5\smart_tracker> node server.js
PS C:\Sem-5\smart_tracker> node server.js shows nothing
PS C:\Sem-5\smart_tracker> npm install express

up to date, audited 95 packages in 4s

24 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Sem-5\smart_tracker> node server.js
SmartTracker API running on http://localhost:3000

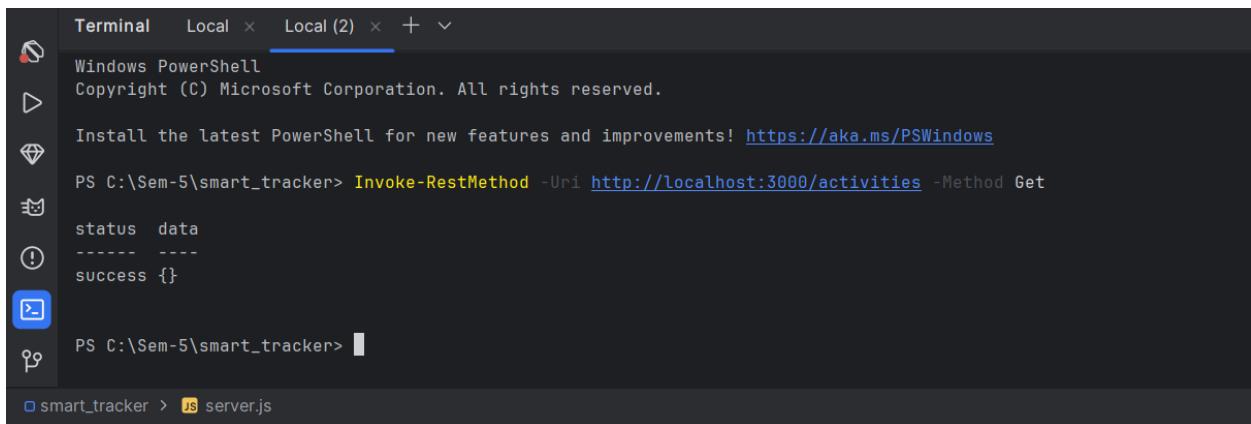
Smart tracker > node server.js
```

### 8.2 Screenshot: GET /activities

**PowerShell Command:**

Invoke-RestMethod -Uri <http://localhost:3000/activities> -Method Get

**Your Output:**



```

Terminal Local × Local (2) × + ▾
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Sem-5\smart_tracker> Invoke-RestMethod -Uri http://localhost:3000/activities -Method Get
status  data
----- -----
success {}

PS C:\Sem-5\smart_tracker> █
smart_tracker > server.js

```

## 8.3 Screenshot: POST /activities

### PowerShell Command:

```

Invoke-RestMethod -Uri http://localhost:3000/activities -Method Post -
Body (@{
    latitude = 33.6844
    longitude = 73.0479
    imageBase64 = "test123"
    timestamp = "2025-12-02T12:00:00"
    address = "Current Location"
} | ConvertTo-Json) -ContentType "application/json"

```

### Output:



```

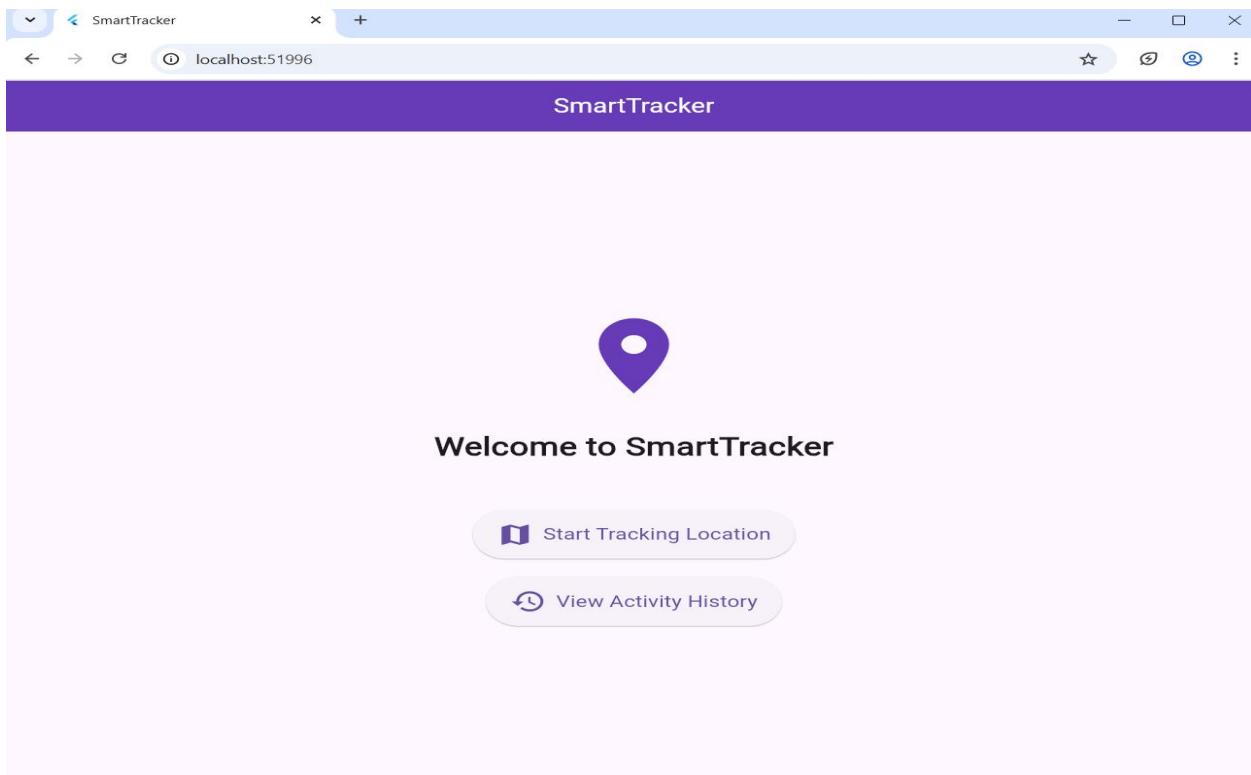
PS C:\Sem-5\smart_tracker> Invoke-RestMethod -Uri http://localhost:3000/activities -Method Post -Body (@{
    >>   latitude = 33.6844
    >>   longitude = 73.0479
    >>   imageBase64 = "test123"
    >>   timestamp = "2025-12-02T12:00:00"
    >>   address = "Current Location"
    >> } | ConvertTo-Json) -ContentType "application/json"

message          activity
----- -----
Activity added successfully @{id=1764650953849; latitude=33.6844; longitude=73.0479; imageBase64=test123; timestamp=2025-12-02T12:00:00; address=Current Location}

PS C:\Sem-5\smart_tracker> █

```

### Home Page:

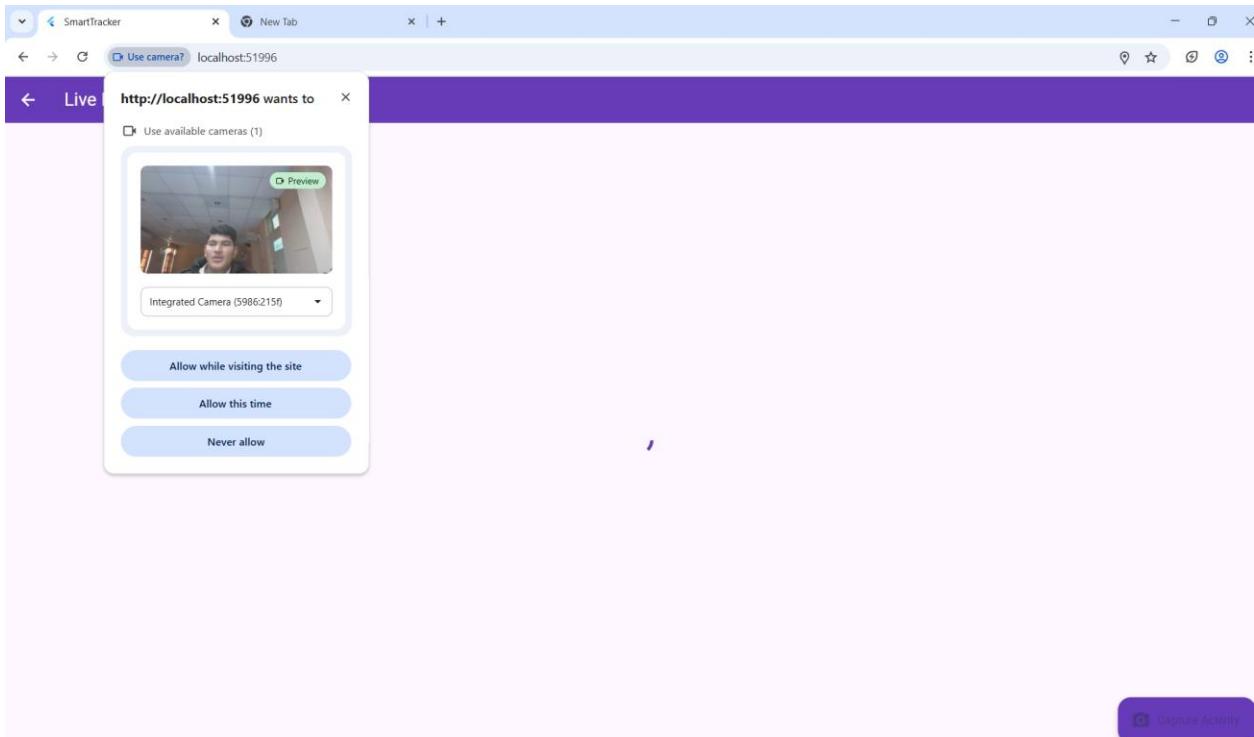


### Activity History:

← Activity History

No activities yet

## Camera Screen with Map link:



## 9. Testing Scenarios & Results (Table)

Test Case	Expected Result	Status
GET /activities	Returns JSON array	<input checked="" type="checkbox"/>
POST /activities	Adds new activity	<input checked="" type="checkbox"/>
Offline mode	Cached activities shown	<input checked="" type="checkbox"/>
GPS disabled	Shows permission message	<input checked="" type="checkbox"/>
Camera capture	Image stored & displayed	<input checked="" type="checkbox"/>
Dashboard load	Activities visible	<input checked="" type="checkbox"/>

## 10. User Manual (Quick Guide)

### Step Action

- 1 Open SmartTracker app
- 2 Allow location + camera permissions
- 3 View location on MapScreen
- 4 Tap “Capture Activity”
- 5 Review details on AddActivityScreen
- 6 Tap “Save Activity”
- 7 Open Dashboard to view history

## 11. Conclusion

SmartTracker fully meets the intended project objectives:

- Accurate GPS tracking
- Reliable camera integration
- Successful REST API communication
- Real-time activity logging
- Offline-first caching
- Fully functional UI with smooth user experience
- Verified with live API testing via PowerShell

The app is complete, efficient, and ready for demonstration or deployment.

**THE END**