**UNIVERSITY OF MORATUWA**

**Faculty of Information Technology**

**B.Sc. (Hons) IT&M**

**Level 3 Semester 5**

**IN 3510 Wireless Communication & Mobile Networks**

---

# Final Individual Assignment: Personalized Weather Dashboard (Flutter + REST)

**Platform:** Android | **No backend required**

## What you will build

A small Flutter app that:

1. Reads your **student index** (e.g., 194174B)
2. Derives **latitude/longitude** from it
3. Calls **Open-Meteo** (no API key) and shows the **current weather**
4. Displays **index, coords, request URL, last update time**, and handles errors
5. Caches the last successful result locally (so it shows something offline)

### Derive coordinates from index:

```
firstTwo  = int(index[0..1])      // e.g., 19
nextTwo   = int(index[2..3])      // e.g., 41
lat = 5  + (firstTwo / 10.0)      // 5.0 .. 15.9
lon = 79 + (nextTwo  / 10.0)      // 79.0 .. 89.9
```

### Weather API (use this)

**Open-Meteo (no key):**

https://api.open-meteo.com/v1/forecast?latitude=LAT&longitude=LON&current_weather=true

*(If Open-Meteo is blocked on your network, you may choose a different public weather API and note it in your report.)*

## Functional requirements (must have)

- Text input for **index** (pre-filled allowed).
- Show **computed lat/lon** (2 decimals).
- Button **Fetch Weather** → calls API and shows:
    - Temperature (°C), Wind speed, Weather code (raw number is fine)
    - **Last updated time** (from device clock)
- Display the **exact request URL** on screen (in tiny text) for verification.
- **Loading indicator** while fetching; **friendly error** if it fails.
- **Cache** the last successful result using shared_preferences and show "(cached)" tag if offline.

## Deliverables

1. **project_<index>.zip** – full Flutter project
2. **report_<index>.pdf** (≤2 pages):
    - Index, formula, coordinates
    - Screenshot(s) with request URL visible
    - 3–5 sentence reflection (what you learned, issues faced)
3. **video_<index>.mp4** (≤60s):
    - Type/confirm index → Fetch → show live result
    - Toggle airplane mode and show error + cached state

## Marking

| Area | Points |
|---|---|
| Correct index→coords & shown in UI | 15 |
| Working API call & JSON parsing | 25 |
| Loading, error handling, and offline cache | 25 |
| Clean UI (labels, layout, readability) | 15 |
| Report (clear + screenshots + URL) | 10 |
| Video demo (≤60s, shows online + cached) | 10 |

## Student checklist (what to show in video/report)

- The app displays **the index**, computed **latitude/longitude**, and **request URL**.
- Tap **Fetch Weather** → Temperature, wind/code appear, with an **updated time**.
- Turn on **Airplane Mode** → tap Fetch → shows **friendly error or cached** data.
- Report includes **screenshot(s)** with URL visible and 3–5 lines of reflection.

> Note: Students may be randomly selected to explain their implementation during a brief viva session, if required.