مكتب وزير الدولة لشؤون
تكنولوجيا المعلومات
والذكاء الاصطناعي

CODE
FOR
LEBANON

HTech

Code4Lebanon x USJ Hackathon

Student Working Guide

How to Approach the Challenge Effectively (From Idea to Prototype)

What You Are Building (Read This First)

Your task is to build a **web prototype** for a **National Monitoring & Analytics Dashboard for NUMŪ**, using a **survey-first data architecture**. The prototype must follow a **two-tier architecture**: **backend + frontend**. Please refer to the **"Hackathon Challenge: National Program Monitoring & Analytics for NUMŪ"** document for the full project scope.

Recommended Team Workflow (How You Should Work)

1. Challenge Identification and Understanding (First 30–45 minutes)

Goal

Understand the problem before writing code.

What your team should do

Read the challenge and answer these questions together:

- **Who is the user of the dashboard?**
  (MITAI / program administrators / policy decision-makers)
- **What decisions should the dashboard help them make?**
  Examples:
  - Which channels are bringing the most learners?
  - Which regions are underrepresented?
  - Which training tracks are in demand?
  - What challenges are learners reporting?
- **What is non-negotiable?**
  Two-tier architecture (backend + frontend).

Output (deliverable for this step)

Create a **1-page problem understanding note** (bullet points), including:

- Problem statement (in your own words)
- Primary user(s)
- Key decisions supported
- Minimum features you will build today

Common mistake to avoid

❌ Jumping into UI design or coding before understanding what insights the Ministry needs.

2. Architecture of the Solution (Most Important Technical Step) (45–60 minutes)

Goal

Design the system before implementation so your team can split work in parallel.

The brief requires a **Two-Tier System**:

- **Backend**: fetch survey data, merge provider status, aggregate metrics
- **Frontend**: consume backend processed data and visualize it

A. Recommended architecture (hackathon-friendly)

1) Data Source

- Survey Mock API
  => API Documentation: https://numu-survey.codeforlebanon.com/api-docs
  API KEY: `cfl_7f3a9b2c8d1e4f6a0b5c3d9e2f4a1b3c`

2) Backend Layer (API / service layer)

Your backend should:

- Fetch raw survey data
- Normalize learner profiles
- Merge provider status by student/learner ID
- Aggregate analytics for frontend use

The brief explicitly asks for normalization + aggregation for filters by:

- Dissemination channel
- Track and learner profile
- Region/geography

3) Frontend Layer (dashboard)

Your frontend should display:

- Charts / tables / filters
- Drill-down views (especially for dissemination channels)
- Unified learner profile page/card

B. Suggested API endpoints (example)

Even if simple, structure your backend clearly:

- `GET /api/summary`
- `GET /api/dissemination`
- `GET /api/interests`
- `GET /api/geography`
- `GET /api/learners`
- `GET /api/learner/:id`

This makes your system look professional and scalable.

C. Suggested team split (5 students)

Since most teams are ~5 members, a practical split is:

1. **Team Lead / Product + Pitch**
    - scope, decisions, final storytelling
2. **Backend Dev (1 to 2)**
    - data ingestion, normalization, aggregation
3. **Frontend Dev (1 to 2)**
    - dashboard layout + charts + filters + learner profile + integration
4. **UX / QA / Integration**
    - UI polish, testing, demo flow, slides, submission prep

(If no designer: make person #5 handle QA + charts + presentation)

Common mistake to avoid

❌ One person coding everything while others wait.
✅ Split work after architecture so people build in parallel.

3. Implementation of the Solution (Main Build Phase) (Majority of the Day)

Goal

Build a working prototype that is **coherent, demonstrable, and aligned to the brief**.

Recommended implementation sequence (very important)

Phase 1: Set up the skeleton (30–45 min)

- Create frontend app + backend app
- Define data schema / types
- Add mock data or Survey API connection
- Confirm local run works for everyone

Phase 2: Backend data pipeline first (60–90 min)

Build the logic for:

- Fetching survey data
- Normalizing records
- Aggregating by:
    - channel
    - track/profile
    - region

Then expose simple endpoints.

✅ If your backend is good, frontend becomes much easier.

Phase 3: Frontend dashboard core screens (90–120 min)

Build the 4 required sections in simple form:

- 1–2 charts per section is enough if done well
- Prioritize clarity over visual complexity

Phase 4: Integration + filters + learner profile (60–90 min)

- Connect frontend to backend endpoints

- Add filters (channel, region, track)
- Add one Unified Learner Profile detail page/card
- Add provider status badge

Phase 5: Polish + testing + demo prep (45–60 min)

- Fix bugs
- Ensure data loads
- Simplify UI
- Prepare pitch and demo route
- Create backup screenshots/video in case of live demo issue

How to Prioritize (If You Run Out of Time)

Build in this order (highest value first)

1. **Backend aggregation working**
2. **Dissemination Performance section**
3. **Geographic Insights**
4. **Unified Learner Profile**
5. **Interest & Strategy Insights**
6. UI polish

Why: judges will reward **correct logic + useful policy insights**, not just pretty screens.

Team Presentation Strategy (Very Important)

Since pitch time is likely short, your presentation must be focused.

Suggested pitch structure (simple)

1) Problem (20–30 sec)

What NUMŪ needs to monitor and why it matters nationally.

2) Solution (30–45 sec)

What your dashboard does and how it supports program decisions.

3) Architecture (30–45 sec)

Show two-tier system:

- Backend ingestion + aggregation
- Frontend visualization

4) Demo (60–90 sec)

Walk through:

- Channel performance
- Geography
- Learner profile
- Provider status

5) Why your approach is strong (20–30 sec)

Mention:

- scalability
- clarity
- policy relevance
- implementation readiness

Minimum Deliverables Checklist (What Your Team Should Have Before Presenting)

Required

- ✅ Working prototype (frontend + backend)
- ✅ Clear architecture explanation
- ✅ At least core required sections implemented
- ✅ One demo flow prepared
- ✅ Team member ready to present

Strongly recommended

- ✅ Architecture diagram
- ✅ GitHub repo (if applicable)

Common Mistakes Teams Should Avoid

1. Overbuilding UI, underbuilding logic

A beautiful UI with weak analytics will score poorly.

2. Ignoring the “survey-first” requirement

The brief is explicit: survey is the primary source of truth.

3. No backend / fake-only frontend

The brief requires a two-tier architecture to qualify.

4. No clear story in the pitch

Judges need to understand:

- what problem you solve,

- what you built,
- why it matters.

5. Trying to finish every feature perfectly

A focused prototype with correct logic beats a half-finished “everything dashboard.”

Final Advice to Teams

Think like a **public-sector product team**, not just hackathon participants.

Your goal is not only to “show charts.”
Your goal is to demonstrate how MITAI could use this dashboard to make **better national decisions** about outreach, inclusion, and training impact.

Build something **credible, structured, and useful**.