Note: Dear participants, please make a copy of this doc and delete these instructions. Do not request edit access on this doc iself.

# C4GT DMP - Proposal Template

| | |
|---|---|
| Name | Mohanraj A |
| Email ID | mohn08052006@gmail.com |
| Phone Number | 8248777476 |
| GitHub ID | mohan-bee |
| Discord ID | 1238122660345548882 |
| Current occupation<br>*(Working Professionals - add current organization & years of exp)* | Student |
| Education Details<br>*(College Name - Degree Name and branch of engineering or other course/specialization)* | Lovely Professional University - B.tech - CSE (Software Product Engineering) |
| Technical skills with level<br>*(Mention tech skills/languages known/UI-UX and level - Novice/Intermediate/Expert)* | Full Stack web developer familiar with javascript and typescript - Intermediate |

CODE 4
GovTech
Building Digital Public Goods

Title: **Poshan Tracker Augmentation: Child-Level Nutrition Monitoring and Alerts**

Summary

I am here to build a unified, child-centric nutrition tracking system that will upgrade the existing Poshan Tracker by connecting the data from Anganwadi centers (AWCs), Community Health Centers (CHCs), hospitals, and periodic surveys. My approach includes automated data extraction pipelines, a relational PostgreSQL database model to track the data, a visually appealing analytics-driven dashboards and at the end rule based alert messaging system through SMS and email. This solution will track the granular movements of growth patterns, periodic identification of malnutrition risks and efficient actions by supervisors and frontline workers.

Project Detail

## Project Overview

Understanding of the project

- Extract Poshan Tracker exports at regular intervals (daily/weekly) via SOPs and automation.
- Standardize and load data into a relational PostgreSQL schema centered on child records.
- Integrate multi-point data inputs: Anganwadi observations, CHC checkups, hospital referrals, and community surveys.
- Build analytics dashboards (Metabase/Superset) for growth trend visualization and intervention tracking.
- Implement SMS/email alerts via Twilio or Google Apps Script for critical malnutrition flags.

## Issues and Support Needed

- Authentication based exports: Cannot perform export operation in Poshan Track for unauthorized users and making it difficult to make the automated export pipeline.
- Data Quality: Exports may vary in schema so this requires coordination with Poshan Tracker officials for consistent CSV exports.

Probable Solutions

- Establish a versioned CSV schema document and pre-processing scripts to normalize inputs.
- Use PostgreSQL encryption extensions (pgcrypto) and strict IAM roles.

## Macro Implementation Details with Timelines

### **Milestone 1**: SOP design and automated extraction script; initial data load for 2 AWC clusters (Weeks 1-3)

#### Deliverables

- Versioned CSV Schema Document
    - Definition of required fields, types, formats, and version tags for Poshan Tracker exports.
- Extraction SOP
    - Step-by-step standard operating procedure (PDF/Markdown) covering authentication, polling cadence, retry logic, error handling.
- Data Extraction Scripts
    - Python scripts (with unit tests) to pull child-level records daily/weekly into a staging schema.
- Initial Load Report
    - Logs and data quality summary for two pilot AWC clusters (counts, missing fields, schema compliance).

## Outcomes

- Automated Pipeline in Place
    - Extraction runs on schedule with < 1% failure rate (monitored via logs).
- Validated Staging Data
    - 95% of records conform to schema, with data quality issues documented for remediation.
- Team Alignment
    - AWC supervisors trained on the SOP, and handover materials reviewed with Poshan officials.

## Code Sample:

1. Data Extraction SOP (Python):

```python
# Configuration
poshan_url = 'https://poshan.tracker/api/exports' # sample endpoint
local_csv = '/data/poshan_export_{}.csv'.format(datetime.today().strftime('%Y%m%d'))

# Step 1: Fetch export
response = requests.get(poshan_url, params={'type': 'child_records'})
response.raise_for_status()
with open(local_csv, 'w') as f:
    f.write(response.text)
```

# **Milestone 2**: Database schema implementation and multi source ingestion(CHC & survey data) (Weeks 4-6)

## Deliverables

- Relational PostgreSQL Schema
  - Child, guardian, growth measurements, CHC checkups, hospital referrals, survey tables with PK/FK constraints and pgcrypto encryption on PII fields.
- ETL Pipelines for CHC & Survey Data
  - Python/Twilio-backed ingestion scripts to normalize and load CHC checkup CSVs and community survey JSONs into the new schema.
- Data Validation Suite
  - Automated tests (pytest) for referential integrity, range checks on anthropometric values, and deduplication logic.

## Outcomes

- Unified Child Records
  - All sources (AWC, CHC, surveys) linked by unique child_id in the database.
- Secure & Compliant Storage
  - PII encrypted at rest; IAM roles ensure least-privilege access.
- Baseline Analytics Ready
  - Data completeness ≥ 90% across fields required for growth-curve analysis.

## Code Sample

```python
# Load into PostgreSQL staging
conn = psycopg2.connect(dbname='poshan', user='user',
password='pw')
cur = conn.cursor()
```

```
cur.execute("TRUNCATE staging.child_records;")
with open(local_csv, 'r') as f:
    reader = csv.reader(f)
    next(reader)  # skip header
    for row in reader:
        cur.execute(
            "INSERT INTO staging.child_records VALUES
(%s,%s,%s,%s,%s)",
            row
        )
conn.commit()
cur.close()
conn.close()
```

## **Milestone 3**: Analytics dashboard and alerts engine with SMS/email integration; end user testing (Weeks 7-10)

### Deliverables

- Analytics Dashboards
    - Metabase (or Fully typed Next.js Application) dashboards showing growth trajectories, cluster-level malnutrition rates, intervention tracking, exportable charts.
- Rule-Based Alerts Engine
    - Node.js service that queries analytics.risk_flags and sends SMS/email alerts via Twilio or Google Apps Script.
- End-User Testing & Feedback Report
    - Scripts for UAT sessions with supervisors; consolidated user feedback and bug-fix log.

### Outcomes

- Interactive Insights
    - Supervisors can drill down from district → AWC → child; date filters and export options available.
- Timely Alerts Delivered
    - ≥ 95% of “high severity” flags generate alerts within 1 hour of flag creation.
- Improved Response Times
    - Measurable reduction in follow-up lag (baseline vs. post-launch) tracked over next quarter.

**Code Sample**:

JavaScript

```
async function sendAlerts() {
  await pg.connect();
  const res = await pg.query(
    `SELECT child_id, guardian_phone FROM analytics.risk_flags
WHERE severity='high';`
  );
  for (const row of res.rows) {
    await client.messages.create({
      body: `Alert: Child ${row.child_id} flagged for
malnutrition. Please follow up.`,
      from: process.env.TWILIO_FROM,
      to: row.guardian_phone,
    });
  }
  await pg.end();
}

sendAlerts().catch(console.error);
```

Availability

| | |
|---|---|
| Number of hours available to dedicate to this project per week | 42 hours/week |
| Do you have any other engagements that will require your time? (projects/internships) | No |

Share any other details about your availability clearly here- NA

Personal Information

## About Me

I'm Mohanraj, currently in my first year, an aspiring full-stack engineer with a strong foundation in Python, JavaScript, and TypeScript. I thrive on tackling new challenges—learning quickly, solving problems creatively, and staying disciplined in my work. Whether it's diving into complex codebases or refining user experiences, I bring dedication and enthusiasm to every project.

## Motivation

This project speaks to me on a personal level: my mother is an Anganwadi worker, and through our conversations she also witnessed firsthand the obstacles she and her colleagues face. By building a child-centric nutrition tracking system, I hope not only to make a real difference for frontline workers and the families they serve, but also to make my mother proud of the positive impact her work—and my efforts—can have in our community.

Please mention if you have solved any issues/tickets for this or other C4GT projects: (Optional)

| Link to to Issue | Resolution description in short | Link to pull request |
|---|---|---|
| #4627 | Improved test coverage | Link |

Previous experience/open source projects (Optional):

In this section you can mention your relevant work experience/projects (not just limited to open-source). You should mention experiences in this section if any with the relevant tech stack of the project (for product usability & design projects, the design software you used; like Figma; can be mentioned)

| Project Name | Project Description | Links (if any) |
|---|---|---|
| tscircuit.com | Added settings feature | PR Link |
| tscircuit.com | Readme code syntax highlight | PR Link |