

# **Revised Project Proposal: Development of a Web-Based Management System for Husma Foundation**

## **1. Executive Summary**

**Project Title:** Digital Transformation of Husma Foundation: A Python-Centric Web Platform for Nutritional Aid Distribution and Global Donations

**Proposed By:** [Your University Team Name/Group ID], Department of Information and Communication Technology, [University Name]

**Date:** September 23, 2025

**Duration:** 1 Month (October 2025) – Accelerated for University Review

**Estimated Cost:** LKR 100,000 (reduced for MVP focus; covers hosting, API keys, and domain; leverages university resources for development)

Husma Foundation, a Sri Lankan charity supporting children with cancer through milk powder and nutritional aid, faces inefficiencies from manual record-keeping. This revised ICT project proposes a Python-driven (using Django framework) web application to automate distributions for up to 100 children monthly, enable secure donations from countless global users (prioritizing all countries), and integrate real-time multilingual support. Key enhancements include anonymous donations with email confirmations, a multi-language AI chatbot for donor queries, live Facebook community integration, SMS notifications, downloadable PDF reports, and transparency features like impact statements.

Inspired by the American Cancer Society's donation flows and Indira Cancer Trust's support tools, the system will emphasize mobile-first design for donors and guardians, including in-person donation options at the Homagama center (278/1, Katuwana Road, Homagama, Sri Lanka). Monetization via Google AdSense (as KMPlayer lacks a confirmed affiliate program) aims to generate supplementary funds.

This 1-month timeline aligns with university interim deadlines, delivering an MVP that scales for future expansions like volunteer management. Success metrics: 80% user satisfaction, 95% uptime, and initial donation processing for 50+ users.

## **2. Introduction**

### **2.1 Background**

Husma Foundation provides essential milk powders (e.g., Pediasure Age 2+, Ensure) and extends to medicines (e.g., Paracetamol Syrup) and other items (e.g., vitamin syrups) to

children with cancer, distributed to guardians at clinics (Wednesdays, Thursdays, Fridays, 9:00 AM–12:00 PM). With 100 children supported monthly and "countless" potential donors worldwide, manual processes hinder scalability.

The project incorporates:

- **Indira Cancer Trust:** Adapting helpline concepts into a multilingual chatbot for equitable access.
- **American Cancer Society:** Fund-specific, anonymous donations with global reach.
- **Husma Facebook Group** (700+ members): Live posts for community stories, highlighting events like urgent milk drives (e.g., recent 2025 posts on nutritional needs).

All development prioritizes Python for backend logic, ensuring alignment with university requirements.

## 2.2 Problem Statement

- Inefficient manual tracking for 100+ monthly distributions.
- Limited global donation access, lacking anonymity, multilingual support, and in-person options.
- No real-time query handling or transparency tools.
- Absence of scalable notifications and reports.

## 2.3 Rationale

A Python/Django-based platform will digitize operations, supporting all countries via Stripe/PayPal (test accounts to be created). The 1-month timeline focuses on MVP, with scalability for expansions.

## 3. Objectives

### 3.1 General Objective

Develop a Python-powered web system to streamline Husma's aid distribution and boost global donations through user-centric features.

### 3.2 Specific Objectives

- Automate non-sensitive distribution records (e.g., child name, guardian contact, quantity) for 100 children/month.

- Enable anonymous/registered donations with email confirmations, multi-item support, and in-person pickup at Homagama center.
- Integrate a real-time AI chatbot prioritizing donor queries in English, Sinhala, and Tamil.
- Add live Facebook group feeds, SMS alerts (via Twilio), and PDF reports for all key analytics (stock, trends, distributions).
- Ensure transparency with public impact statements and ethical monetization (AdSense).
- Design mobile-first UI for rural guardians/donors, scalable for future features.

## 4. Scope

### 4.1 In-Scope

- **Core Modules** (Python/Django):
  - Admin Dashboard: Inventory CRUD, PDF reports (using ReportLab library).
  - Guardian Portal: Simple registration (non-sensitive fields: name, age, guardian details), history views.
  - Donor Portal: Cart for items/funds, anonymous toggle (email optional for confirmation), global payments (all countries; LKR/USD/EUR auto-conversion). In-person donation form for Homagama visits.
  - Chatbot: Tidio widget with multilingual flows (English/Sinhala/Tamil) for queries like "How to donate milk powder?" or "Clinic schedule?"
- **Features:** Live FB group posts (via FB SDK), SMS notifications (Twilio for low stock/donations), transparency dashboard (e.g., "Rs. 10,000 funded 5 children").
- **Monetization:** Google AdSense banners (target Rs. 300/month; KMPlayer link integrated as optional download promo, but no earnings confirmed).
- **Branding:** Incorporate Husma logo from FB page (blue/orange scheme; logo details to be fetched via manual review if tool access limited).
- **Deployment:** Heroku with PostgreSQL for 100+ users.

### 4.2 Out-of-Scope

- Sensitive child data (e.g., cancer type, uploads).

- Advanced expansions (e.g., volunteer module) – prepare hooks only.
- Custom affiliate setups beyond AdSense.

## 5. System Design

### 5.1 High-Level Architecture

- **Frontend:** Mobile-first HTML/CSS/JS (Bootstrap), with Python-templated Django views.
- **Backend:** Python 3.12/Django 5.x for all logic (e.g., forms, APIs).
- **Database:** PostgreSQL via Django ORM; models avoid sensitive fields.
- **Integrations:** Stripe/PayPal (Python SDKs; setup guidance provided), Twilio (Python lib for SMS), Tidio (JS for chatbot), FB Graph API (Python requests for live posts), ReportLab (PDF generation).
- **Security:** Python-based validation, HTTPS, anonymous sessions via UUIDs.

### 5.2 Entity-Relationship Diagram (Conceptual)

Entity	Attributes (Non-Sensitive)	Relationships
<b>MilkPowderType</b>	id, name, price, stock_qty, desc	One-to-Many: Distributions
<b>ChildrenRecord</b>	id, child_name, age, guardian_name, contact, date, quantity	Many-to-One: MilkPowderType
<b>Donation</b>	id, donor_email (opt), amount, items (JSON), anonymous (bool), timestamp	Many-to-One: DonorProfile
<b>InventoryLog</b>	id, type_id, change_qty, reason	One-to-One: Donation
<b>DonorProfile</b>	id, name (opt), email (opt), phone (opt)	One-to-Many: Donations

### 5.3 User Flows

- **Donor:** Multilingual chatbot query → Cart → Anonymous/email confirmation → Payment/SMS receipt or Homagama visit scheduling.
- **Guardian:** Register/view history (mobile-optimized).
- **Admin:** Generate PDF reports, monitor live FB feed.

## 6. Technologies and Tools (Python-Focused)

Category	Tools/Technologies	Justification
Backend	Python 3.12, Django 5.x, ReportLab (PDFs), Twilio SDK (SMS)	Core Python emphasis; handles reports/notifications.
Frontend	Django Templates, Bootstrap 5, JS (for cart/chat)	Mobile-first; Python-integrated.
Database	PostgreSQL (Django ORM)	Scalable for 100 children + donors.
Payments	Stripe/PayPal (Python SDKs)	Global; test accounts setup guide included.
Chatbot	Tidio (multilingual JS; Python webhook for escalations)	Prioritizes donor queries (e.g., donation process, items).
Monetization	Google AdSense (JS; Python tracking)	Ethical; KMPlayer link as promo (no affiliate found).
Integrations	Requests (Python for FB API), Celery (async tasks)	Live posts, scalability.
Deployment	Heroku	Quick Python deployment.
Testing	Django TestCase (Python)	Coverage for core flows.

## 7. Methodology

Agile with daily stand-ups for 1-month crunch; Python code reviews emphasized.

### 7.1 Phases

1. **Planning (Days 1-3):** Finalize models, setup test APIs.
2. **Development (Days 4-20):** Backend first (Python models/views), then frontend integrations.
3. **Testing (Days 21-25):** Python unit tests, UAT with mock 100 users.
4. **Deployment (Days 26-30):** Launch MVP, university review handover.

## 8. Timeline

Phase	Duration	Key Milestones	Responsible
Planning	Oct 1-3	Python setup, API keys created	All Team
Development	Oct 4-20	Core Python modules, multilingual chatbot	Developers
Testing	Oct 21-25	Bug fixes, PDF/SMS tests	Tester/All
Deployment & Review	Oct 26-31	Live MVP, university demo	DevOps/Team

## 9. Budget

Item	Estimated Cost (LKR)	Notes
Domain & Hosting (Heroku)	10,000	1-month trial.
API Keys (Twilio/Stripe/Tidio)	20,000	Test accounts; multilingual support.
Development Tools	5,000	Python libs (free).
Misc (Domain Email)	5,000	SMS credits.
Contingency	60,000	Accelerated timeline buffer.
<b>Total</b>	<b>100,000</b>	Grant-eligible.

## 10. Risks and Mitigation

Risk	Probability	Impact	Mitigation Strategy
API Setup Delays	Medium	High	Provide Python setup scripts; parallel testing.
Multilingual Challenges	Low	Medium	Use Tidio's built-in translations.
User Scale (100+ Children)	Low	Medium	PostgreSQL auto-scaling on Heroku.
Timeline Crunch	High	High	Daily Python sprints; MVP prioritization.

## 11. Team Roles and Responsibilities

Role	Members	Responsibilities
<b>Project Lead</b>	[Student A]	Timeline oversight, Python reviews.
<b>Backend Dev</b>	[Student B, C]	Django models, integrations (Twilio, etc.).
<b>Frontend Dev</b>	[Student D]	Mobile UI, chatbot/JS.
<b>Tester/QA</b>	[Student E]	Python tests, multilingual UAT.
<b>Designer</b>	[Student F]	Branding (FB logo integration).

## 12. Conclusion and Expected Outcomes

This Python-centric MVP will transform Husma's operations within 1 month, supporting 100 children and global donors with multilingual, transparent tools. Outcomes: Automated distributions, Rs. 500K+ potential donations, 50% engagement via chatbot/live feeds. Scalable for expansions, it positions Husma as a digital leader in Sri Lankan cancer support. Approval requested for October start.

### Appendices:

- Enhanced HTML Prototype (with anonymous toggle, chatbot).
- Python Setup Guide (e.g., Stripe/Twilio integration snippets).
- References: Django Docs, Twilio Python Guide.

**Approval Signature:** \_\_\_\_\_ (Project Advisor)