Developing a Tech-Based Solution to Reduce Food Loss in Bangladesh

Background

Bangladesh loses a significant portion of its food particularly grains and staple crops because of inadequate storage systems, poor handling, and inefficient transportation. These losses contribute to food insecurity, economic waste, and environmental impact. Reducing food loss is also directly linked to SDG 12: Responsible Consumption and Production, especially Target 12.3, which focuses on cutting food loss along supply chains.

Problem Statement

Bangladesh experiences substantial food loss throughout its supply chain, undermining both economic growth and food security. According to the Financial Express, the country loses about **4.5 million metric tonnes of food grains every year**, resulting from inadequate storage facilities, poor handling practices, and inefficient transportation. These losses represent an annual economic cost of around **US $1.5 billion**, as estimated by the Bangladesh Institute of Development Studies (BIDS) (Financial Express, 2025).

Moreover, food inefficiencies extend beyond grains. Approximately **12–32% of staple foods** including rice, pulses, vegetables, meat, and dairy are lost annually during production and distribution (The Financial Express, 2025). Such high loss rates not only translate to economic losses but also exacerbate food insecurity, particularly among vulnerable communities.

These challenges are directly related to **Sustainable Development Goal (SDG) 12: Responsible Consumption and Production**, especially **Target 12.3**, which calls for reducing food losses along production and supply chains (United Nations, 2025; UN Bangladesh, 2025).

Target 12.3

By 2030, halve per capita global food waste at the retail and consumer levels and reduce food losses along production and supply chains, including post-harvest losses

By failing to address post-harvest and distribution losses, Bangladesh hinders progress toward achieving SDG 12.3, missing an opportunity to improve food availability, stabilize farmer incomes, and build a more resilient food system.

PART A – Pre-Hackathon (48 hours before onsite)

Weight: 50%

Submit GitHub repo + deployed link + 2-minute video demo by deadline.

Must be completed before coming onsite. AI assistance (V0, Lovable, Gemini, Claude, etc.) is fully allowed and encouraged.

Core Tasks (you are free to add more, but at least two of the following must work perfectly)

A1. UI/UX Challenge: Storytelling Landing Page (Problem & Solution Pitch)

- **Storytelling Focus:**
  Create a visually engaging landing/onboarding screen that highlights the seriousness of Bangladesh's food loss problem and clearly presents HarvestGuard as the solution.

- **Problem Narrative:**
  Present a concise, easy-to-understand problem statement explaining the scale and impact of food loss in Bangladesh. Must support both **English and Bangla**.

- **Visual Solution Metaphor:**
  Use strong visual storytelling—CSS animations, SVG graphics, or motion elements—to demonstrate the core app workflow:
  Data → Warning → Action → Saved Food.

- **Mobile-First Engagement:**
  Ensure fast performance, minimal load time, and large, intuitive UI elements designed for low-literacy users, guiding them toward quick registration/onboarding.

Bonus: Use 3D models to enhance the narrative and visual impact of the landing experience.

A2. Farmer and Crop Management (Identity, Inventory, & Engagement)

- **Farmer Registration and Profiles:** Users register with email, a securely hashed password, and a phone number. Profiles must store the user's name, contact information, and preferred language (Bangla/English toggle).
- **Crop Batch Registration:** Verified farmers must be able to register a new harvested batch. Required fields include: **Crop Type** (initially only Paddy/Rice), **Estimated Weight (kg)**, **Harvest Date**, **Storage Location** (Division/District, using a static predefined list), and **Storage Type** (e.g., Jute Bag Stack, Silo, Open Area).
- **Profile Page & Gamification:** Display a list of the farmer's active and completed crop batches, historical loss events, and intervention success rates.
  - **Achievement Badges:** Implement a simple system where farmers earn badges (e.g., "First Harvest Logged," "Risk Mitigated Expert") for completing interventions successfully.
  - Work offline (save to LocalStorage) and sync when online

Must export data as CSV/JSON.

A3: Hyper-Local Weather Integration

- **Requirement:** Fetch live weather data based on the farmer's selected Upazila.
- **Key Features:**
  - Use a free API (OpenWeatherMap/WeatherAPI).
  - Display: Temperature, Humidity, and *Chance of Rain* for the next 5 days.
  - **Constraint:** The UI must be in **Bangla**.
- Show simple Bangla advisories based on weather + farmer's crop data Example: "আগামী ৩ দিন বৃষ্টি ৮৫% → আজই ধান কাটুন অথবা ঢেকে রাখুন" Example: "তাপমাত্রা ৩৬°C উঠবে → বিকেলের দিকে সেচ দিন" Farmer must understand the advice even if he reads slowly.

A4: Prediction, Weather Integration, and Risk Forecasting

- **Prediction Engine Logic:** Implement a simple logic model to calculate the **Estimated Time to Critical Loss (ETCL)** based on continuous high moisture and high temperature readings.
- **Weather Data Integration (Mock):** Integrate a mock 7-day weather forecast (Temperature, Rainfall Probability, Humidity) based on the Batch's registered Division/District. This data must be used to refine the ETCL calculation (e.g., if rain is predicted, drying is riskier).
- **Output Format:** A human-readable risk summary (e.g., **"High Risk of Aflatoxin Mold (ETCL: 72 hours). Weather forecast suggests high humidity, requiring indoor aeration."**)

A5: Basic Crop Health Scanner (AI Wrapper)

- **Requirement:** A feature to upload a photo of a crop.
- **Key Features:**
  - Use a pre-trained API (like HuggingFace or Teachable Machine) to detect if the crop looks "Fresh" or "Rotten."
  - **Note:** You do not need to train a complex model. Focus on the *integration* - how fast does it load on a mobile browser?

Pre-Onsite Evaluation Criteria

Criteria	Weight	Description
Core Functionality (A1–A5)	**25%**	At least two core modules must work flawlessly. Smooth workflows for farmer profiles, crop batch registration, offline mode, weather integration, prediction logic, or pest identification. Deployed link must be stable.
Bangla-First User Experience	**10%**	Clear and simple Bangla UI. Easy to understand for real Bangladeshi farmers. Weather advisories and insights must be in Bangla.
UI/UX & Device Compatibility	**7%**	Clean layout, readable text, big buttons. Offline storage behavior earns credit.
Technical Implementation	**5%**	Quality of integrations (weather API, prediction logic, offline sync, file export). Efficient and structured use of technologies.
Code Quality & GitHub Submission	**3%**	Proper Git usage, meaningful commits, clean project structure, documented setup.
Pre-Onsite Video Demo	**5%**	A clear 2-minute walkthrough showing working features, navigation, and purpose. Professional and easy to follow.

PART B – Onsite (7 hours)

Weight: 50%

AI coding assistants are still allowed, but the tasks are deliberately designed so that copy-paste from AI will not give high scores, you need to integrate, debug together, make smart decisions, and make it feel alive.

Required Tasks (all teams must attempt as many challenges as possible)

B1: Visualize the local risk landscape to foster community awareness.

- **Description:** This utility establishes a Local Risk Map to promote community awareness of spoilage threats. It integrates a responsive map component centered on the farmer's district and populates it with anonymous, color-coded markers representing the current risk status (Low, Medium, or High) of surrounding farms. The goal is to provide visual context about regional threats without compromising individual privacy.
- **Features:**
  - **Map Setup:** Integrate a simple, responsive map component (e.g., **Leaflet.js**) that auto-centers and zooms to the farmer's registered **Division/District**.
  - **Mock Neighbor Data:** Generate **10-15 mock data points** (coordinates, Risk Level: Low/Medium/High) within the chosen District boundaries, maintained client-side.
  - **Pin Visualization:**
    - Plot the farmer's own (mocked) location with a **distinct blue pin**.
    - Plot mock neighbor locations using clear, color-coded markers (Green for Low Risk, Red for High Risk) that reflect their mock risk level.
  - **Privacy Constraint:** All neighbor data displayed **MUST be completely anonymous** (no names or personal identifiers).
  - **Bangla Pop-up Interaction:** Tapping a neighbor pin must trigger a small pop-up (tooltip) displaying the **General Crop Type**, **Current Risk Level**, and **Last Update Time** (mocked) entirely in **Bangla** (e.g., "ঝুঁকি: উচ্চ").
  - **Interactivity:** The map must support standard, touch-friendly panning and zooming.

B2. The Smart Alert System (Decision Engine)

- **The Task:** Generate Specific Actionable Advice in Bangla.
- **Complexity:** It must combine data from Crop Type + Weather + Risk.
  - *Bad Alert:* "Weather is bad."
  - *Good Alert:* "আগামীকাল বৃষ্টি হবে এবং আপনার আলুর গুদামে আর্দ্রতা বেশি। এখনই ফ্যান চালু করুন।" (Tomorrow it will rain and humidity is high in potato storage. Turn on fans now.) .

  - **Requirement:** Simulate an SMS notification in the browser console when risk hits "Critical."

B3: Pest Identification and Action Plan (Visual RAG)

- **Description:** This utility enables farmers to upload a picture of a pest or crop damage. The system uses the **Gemini API's Visual RAG capabilities** (Image Understanding + Google Search Grounding) to instantly identify the threat, assess the risk, and generate a hyper-local, grounded, and specific treatment plan entirely in Bangla.
- **Features:**
  - **Image Upload Interface:**
    - Dedicated, mobile-friendly button for uploading an image (JPEG/PNG) of the pest/damage.
    - Display a preview of the uploaded image before submission.
  - **Gemini Visual RAG Flow (MANDATORY):**
    - Send the uploaded image and a text prompt to the **Gemini API**.
    - **Tool Use:** The API call includes the Google Search grounding tool.
  - **AI-Generated Output:** The Gemini response must identify the pest and provide a categorized **Risk Level** (High/Medium/Low).
  - **Grounded Action Plan (Bangla Output):**
    - The generated treatment plan must be practical, focus on local methods, and be presented entirely in **Bangla**.

Bonus: Instead of utilizing the Google Search Grounding Tool, employ the Custom RAG pipeline.

B4. Bangla Voice/Touchless Interface

Farmers should be able to **ask questions** in spoken Bangla and get spoken answers. Minimum workable version (will get full marks):

- Use **Web Speech API** (recognition + synthesis) with language 'bn-BD'
- Support at least 4–5 common questions (আজকের আবহাওয়া? আমার ধানের অবস্থা? গুদামে কী করব? কবে কাটব? etc.)

- If Web Speech API Bangla is not good in your test → fall back to text chat with Bangla keyboard

Bonus: The system is capable of converting rural-accented speech into accurate text

Onsite Evaluation Criteria

Criteria	Weight	Description
B-Feature Functionality & Stability	**20%**	At least two of the Part B features (B1–B4) must work smoothly. The features should load quickly, function reliably, and integrate properly with the existing Part A system.
Bangla Experience (UI + Voice/Text)	**10%**	All onsite-created features must work fully in Bangla. Voice queries, alerts, and texts should be simple, clear, and understandable for farmers.
Real-Time Integration & Decision Logic	**10%**	On-site features should combine data from multiple modules (crop info, weather, scanner results, predictions) to produce meaningful, context-specific outputs.
Farmer-Centric UI/UX	**7%**	Interfaces must remain simple, readable, and optimized for low-cost Android devices. Large buttons, quick interactions, and offline-aware behavior are valued.
Presentation & Practical Impact	**3%**	During the final demo, the solution should clearly show real usefulness, smooth flow, and strong potential value for Bangladeshi farmers.

Bonus (up to +20%)

- Working voice in real Bangla accent (+5)
- Community feature (farmers in same upazila see each other's risk alerts anonymously) (+5)
- Anything that makes judges go "wow this could actually helps farmers

Allowed Tools & APIs (Free tier only)

- **Weather:** OpenWeatherMap, WeatherAPI.com, WeatherBit, AccuWeather free
- **ML:** TensorFlow.js, HuggingFace, Kaggle models, Roboflow
- **3D:** Three.js, Babylon.js, React Three Fiber
- **Backend:** Railway, Render, Vercel, HuggingFace Spaces
- **Storage:** Supabase free, Firebase free, PocketBase
- **SMS:** Twilio free trial or console simulation
- **Note:** No restriction on using any tools apart from the premium

Submission

Part A (pre): GitHub link + deployed URL + 2-min video (by deadline)
Final (onsite): Updated repo + live demo (8 mins + 4 mins Q&A)

References

Financial Express. (2025, February 10). *Strengthening food storage facilities. The Financial Express*. Retrieved from https://thefinancialexpress.com.bd/views/columns/strengthening-food-storage-facilities (The Financial Express)

TBS News. (2025, September 29). *Bangladesh wastes 34% of food annually: World Bank study. The Business Standard*. Retrieved from https://www.tbsnews.net/bangladesh/bangladesh-wastes-34-food-annually-experts-1249056 (tbsnews.net)

United Nations. (2025). *Responsible Consumption and Production – SDG 12*. United Nations Sustainable Development. Retrieved from https://bangladesh.un.org/en/sdgs/12 (bangladesh.un.org)

বাংলাদেশ কৃষি আবহাওয়া তথ্য সেবা (বাংলাদেশ কৃষি আবহাওয়া তথ্য সেবা (বামিস)

Mock Batch Dataset for moisture (mock_batch_state_data.xlsx)

Plant Disease Dataset (Plant Disease)

এই হ্যাকাথনে যারা সত্যিকারের **farmer**-এর কথা ভেবে, সুন্দর করে ইন্টেগ্রেট করবে, তারাই জিতবে — শুধু **AI** দিয়ে কোড জেনারেট করলে হবে না।

শুভ কামনা! এবার দেখি কে সত্যি সত্যি গ্রামের কৃষকের জন্য কিছু বানাতে পারে !