Smart Grocery

List Generator

PRESENTED BY

Shubham Sourava Sahoo

VSSUT, Burla

Challenges of Manual Grocery Lists

- Manual meal planning is chaotic and time-consuming.
- "I want to cook Italian this week" is a vague goal that doesn't translate easily into a shopping checklist.
- Vague planning leads to multiple store trips and food waste.

The Objective:

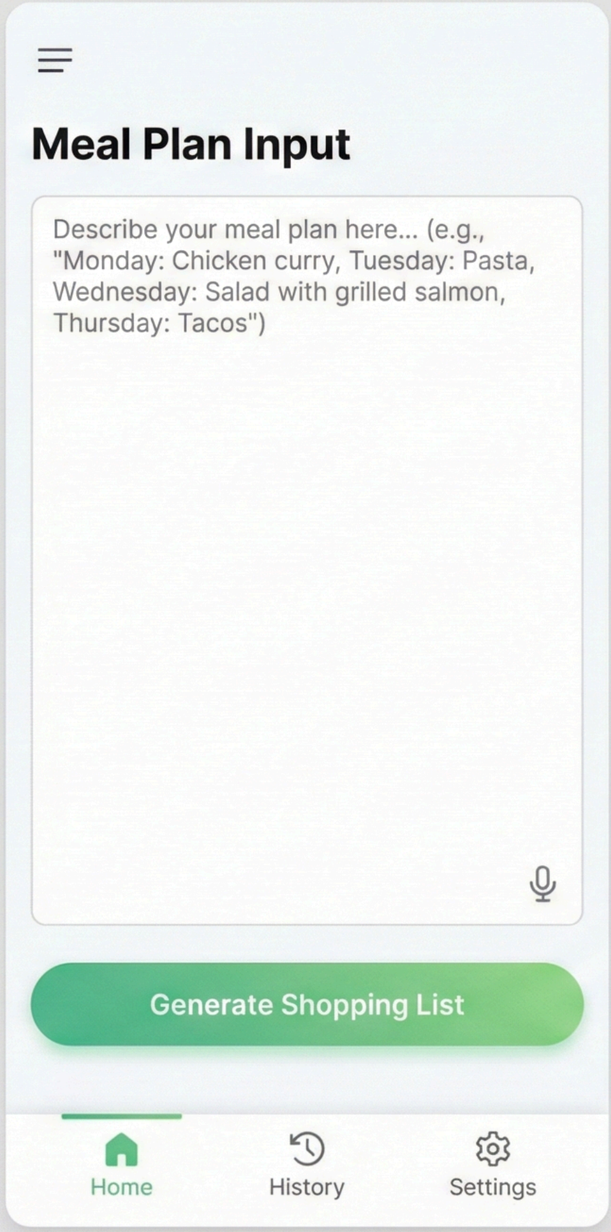
- Build a Smart Grocery Assistant that bridges the gap between meal ideas and shopping carts.
- Leverage Generative AI to understand context (e.g., serving sizes, cuisines) and output structured data.
- Demonstrate the capability of running Large Language Models (LLMs) on consumer hardware for privacy and zero cloud costs.

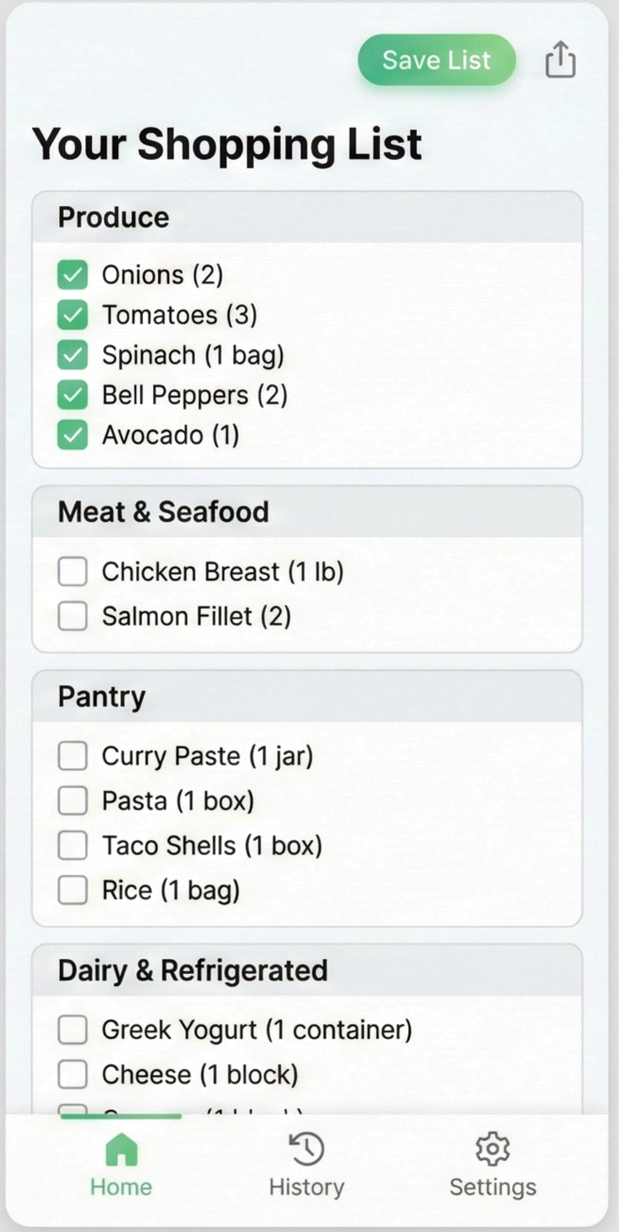
Proposed Solution

A desktop-based application where users simply describe their meal plan in natural language, and the system instantly generates a categorized shopping list.

How it Works:

- Input: User types: "Pasta Alfredo for 2 and a Caesar Salad".
- AI Processing: The model identifies the recipes, extracts ingredients, and estimates quantities.
- Output: A structured table (Item | Category | Quantity) ready for export.

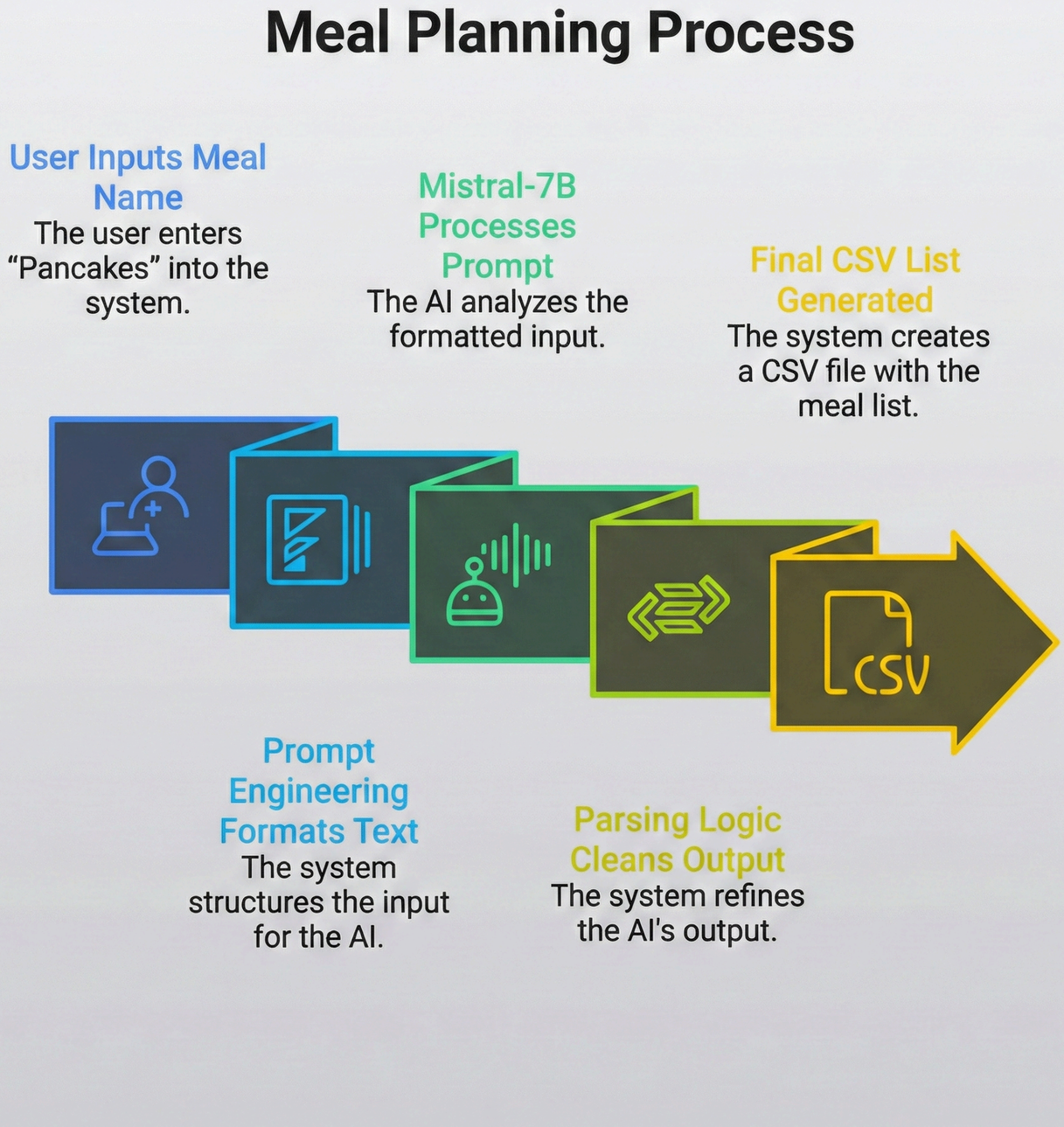




System Architecture

The Workflow:

1. User Input: Captures meal name & servings.
2. Prompt Engineering: Formats text with specific instructions to ensure strict data output.
3. Local Inference: Mistral-7B (Quantized) processes the request locally.
4. Parsing Logic: Python cleans and structures the raw AI text.
5. Final Output: Generates CSV file.



Technology Stack

- Core Language: Python 3.10
- Interface: Streamlit
- AI Model: Mistral-7B-Instruct-v0.2 (Quantized GGUF)
- Inference: Llama.cpp
- Data: Pandas

Implementation Logic

1. Few-Shot Prompting: We give the AI examples to force strict formatting.
2. Robust Parsing: Python logic cleans the AI output to prevent errors.
3. Quantization: Using 4-bit models allows a 7-Billion parameter brain to run on a standard laptop.
4.Note on Trade-offs: We used 4-bit Quantization (Q4_K_M) to compress the model. While this allows local execution, a production version would utilize a REST API architecture. This would offload the computational load to a remote server, significantly improving start-up time and battery efficiency for the end user.

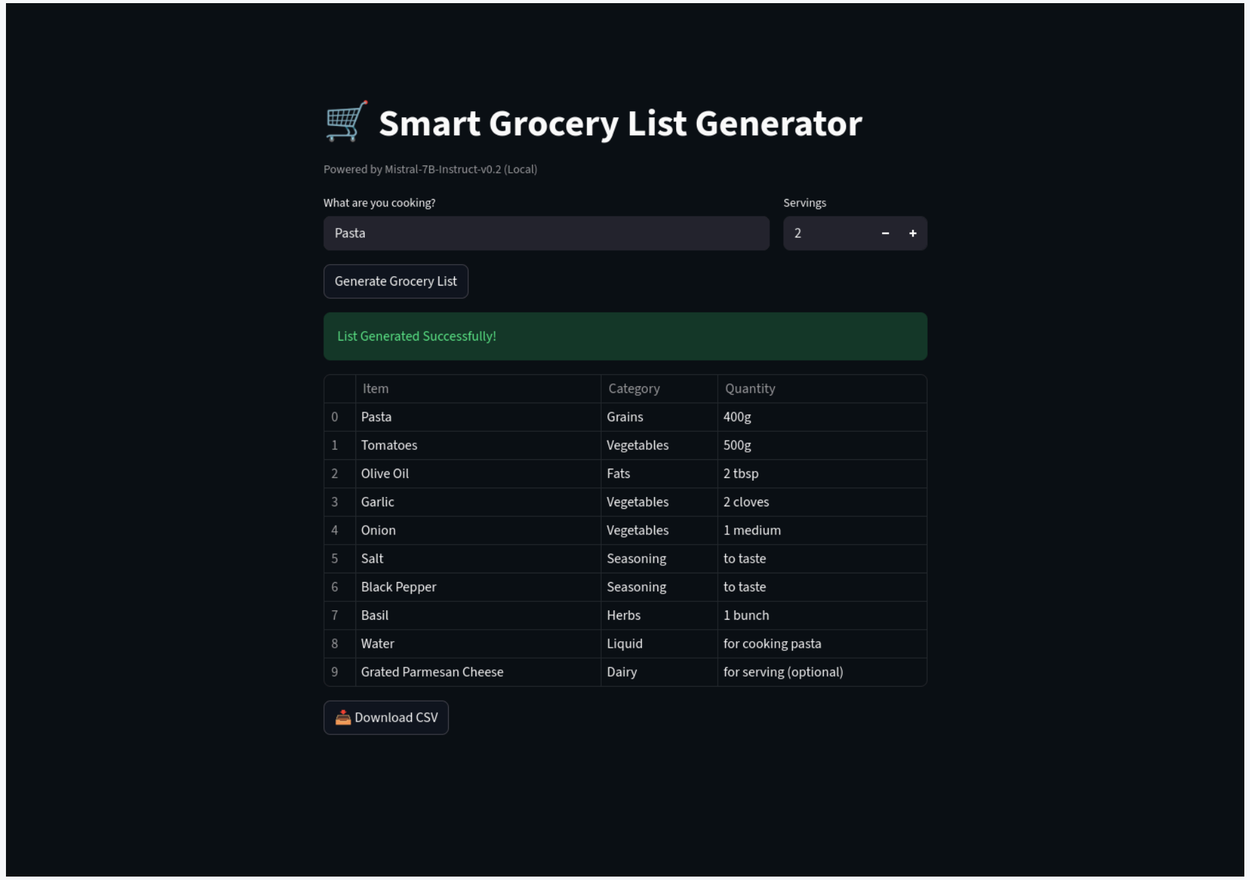
Key Features of Our Solution

- Privacy-First Prototype: Demonstrates secure, local-only processing where no user data leaves the device.
- Zero-Cost Development: Utilizes open-source quantized models to avoid paid API subscriptions during the prototyping phase.
- Structured Output: Enforces strict CSV formatting (Item | Category | Quantity) from unstructured natural language.

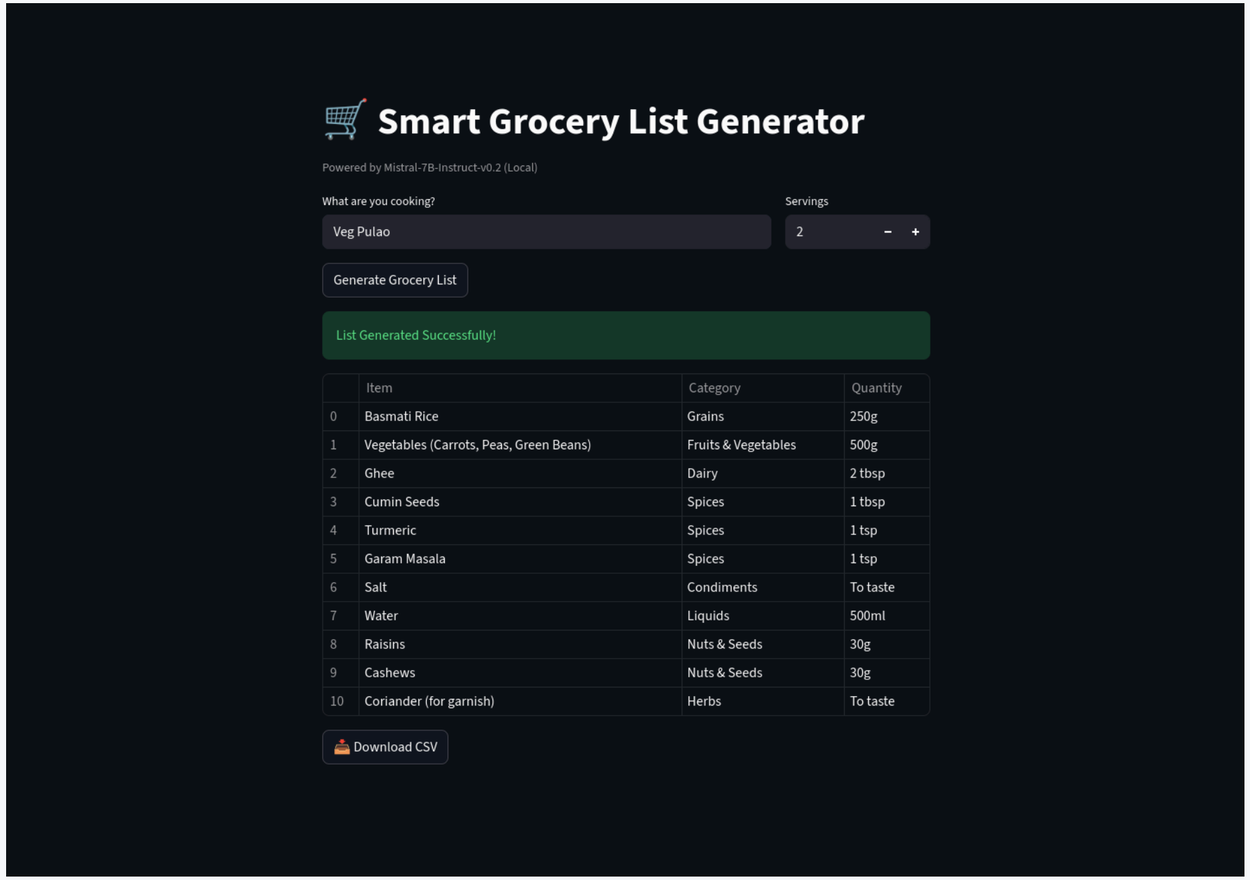
Project Results

Performance:

- Generates lists in <30 seconds.
- Accurately scales quantities (e.g., 2 servings vs. 10 servings).



	Item	Category	Quantity
0	Pasta	Grains	400g
1	Tomatoes	Vegetables	500g
2	Olive Oil	Fats	2 tbsp
3	Garlic	Vegetables	2 cloves
4	Onion	Vegetables	1 medium
5	Salt	Seasoning	to taste
6	Black Pepper	Seasoning	to taste
7	Basil	Herbs	1 bunch
8	Water	Liquid	for cooking pasta
9	Grated Parmesan Cheese	Dairy	for serving (optional)



	Item	Category	Quantity
0	Basmati Rice	Grains	250g
1	Vegetables (Carrots, Peas, Green Beans)	Fruits & Vegetables	500g
2	Ghee	Dairy	2 tbsp
3	Cumin Seeds	Spices	1 tbsp
4	Turmeric	Spices	1 tsp
5	Garam Masala	Spices	1 tsp
6	Salt	Condiments	To taste
7	Water	Liquids	500ml
8	Raisins	Nuts & Seeds	30g
9	Cashews	Nuts & Seeds	30g
10	Coriander (for garnish)	Herbs	To taste

Future Scope

- Pantry Tracking: Check inventory before adding items.
- Price Estimation: Estimate bill using local store APIs.
- Voice Input: Speech-to-text for hands-free use.
- Cloud Deployment for Efficiency: Move the heavy AI model to a cloud server (via Hugging Face API or AWS). This would reduce the user's app size from 4GB to <50MB and allow it to run on any low-end device or mobile phone instantly.

Conclusion

- We successfully bridged the gap between Generative AI and Daily Utility.
- Successfully prototyped a Generative AI application using local hardware, proving that privacy-centric AI is possible.
- Established a clear migration path from a secure local prototype to a scalable cloud-based product for mass adoption.

Github Repo

- github.com/Sonix-7732/smart-grocery-ai

References:

- Mistral AI Research Paper
- Llama.cpp Documentation
- Streamlit Docs
- Hugging Face Model Hub