

PRACTICAL APPLICATIONS OF GENAI – MIDTERM CHECKPOINT

Project: Su-Chef – Your Smart AI Cooking Assistant

Team: Yarden Liberman

& Rony Barzilay



Agenda

Introduction .1

Application Development .2

Methods .3

Results .4

Summing Up .5

Open Questions & Takeaways .6

Motivation

Why We Built Su-Chef: Making Cooking Smarter, Simpler, and More Enjoyable for Everyone

Su-Chef addresses two key challenges:

Recipe Accessibility

- Online recipes are typically generic and scattered across platforms
- They lack personalization based on the user's time, dietary needs, cooking skill, or available ingredients
- This makes it difficult and inefficient for users to find a recipe that truly fits their individual needs

Difficulty Executing Recipes

- It's difficult to follow written recipes while actively cooking: switching attention between the stove and screen, scrolling, and interpreting instructions breaks the flow and creates friction
- There's no dynamic support during the cooking process—if something is unclear, goes wrong, or the user has questions, there's no real-time guidance to explain or assist in adapting the recipe

Objectives

Vision: Create a voice-first, personalized cooking assistant powered by GenAI

Goals:

- Generate tailored recipe suggestions
- Guide users step-by-step with voice output
- Parse and structure custom user recipes
- Learn user preferences over time
- Use an agent to support real-time interaction and adaptability

Related Work

Academic Literature:

RecipeRadar (2023)

Focus: Filters recipes by nutritional goals to encourage healthier eating. •

Limitation vs Su-Chef: Lacks real-time personalization or adaptability during cooking. •

Kochen Helfer (2022) .1

Focus: Tracks cooking actions using vision-based systems to monitor progress. •

Limitation vs Su-Chef: Requires external camera hardware and doesn't support personalized, voice-driven interactions like Su-Chef. •

Su-Chef.

RecipeRec (2021) .1

Focus: Uses graph-based modeling to match users with relevant recipes through ingredient relationships. •

Limitation vs Su-Chef: Lacks interactive cooking support and operates offline. •

•

Existing Apps

Category	Examples	Limitation
Recipe Platforms	Yummly, AllRecipes	No interaction, no personalization
Voice Assistants	Alexa, Google Assistant	Generic, non-contextual
Meal Kit Services	HelloFresh, Blue Apron	Fixed meals, no adaptive experience
Smart Apps	SideChef, Whisk	Basic personalization, limited interactivity

Use Cases

1 Personalized Recipe Suggestions

- Tailored to ingredients, time, skill, dietary needs
- Powered by NLP + prompt design

2 Hands-Free Cooking Guidance

- Real-time spoken instructions
- Powered by Whisper + TTS

3 Adaptive Feedback Loop

- Learns user preferences via feedback
- Powers future recipe refinement

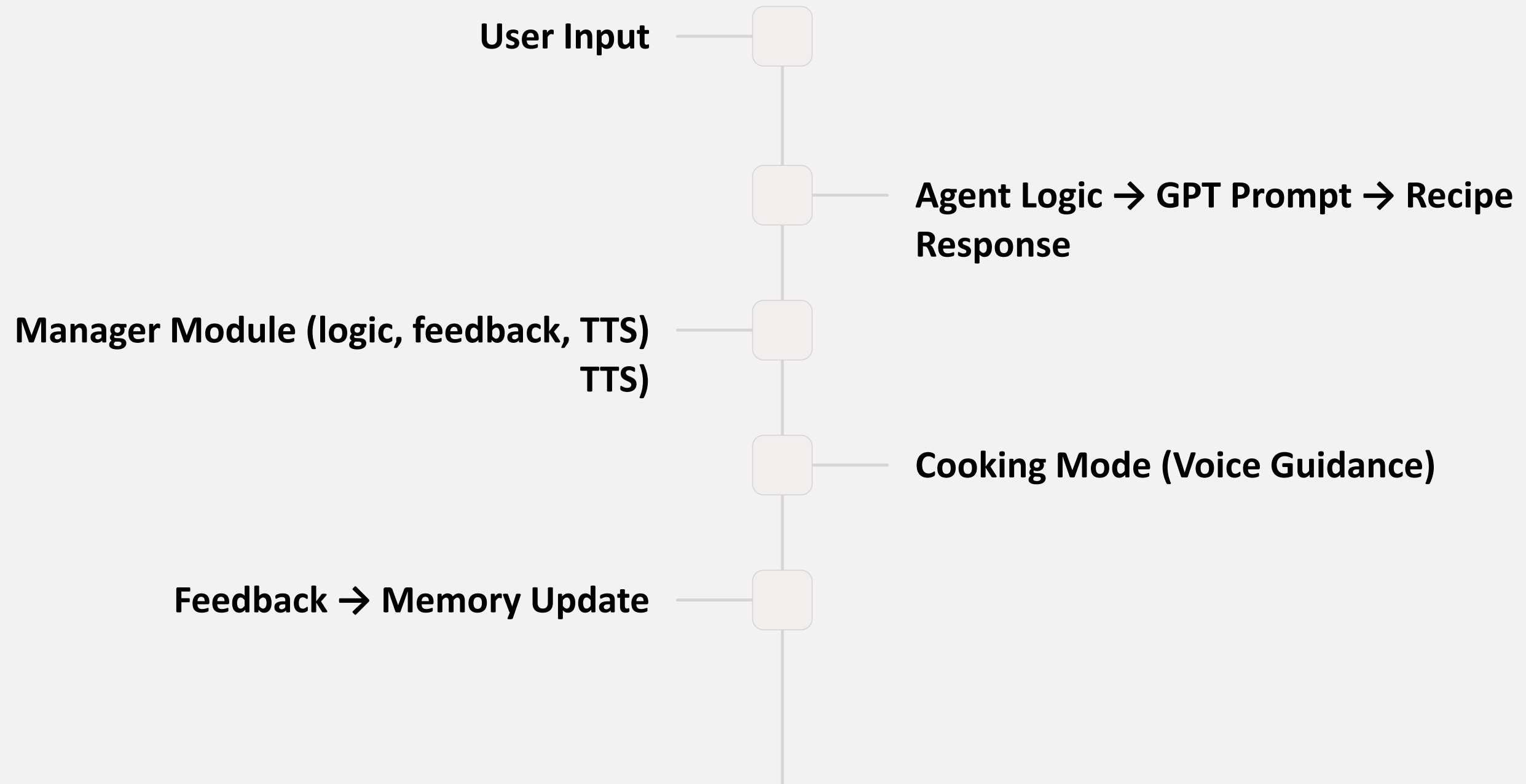
4 Recipe Structuring

- Parses user-submitted recipes
- Segmented steps, formatted output

5 Agent-Driven Cooking Guide

- Provides real-time spoken instructions and dynamically answers user questions during recipe preparation

Application Flow



GenAI Tech Requirements

Functionality	Requirement	Tools Used
Recipe Generation	NLP + prompt generation	Azure GPT-4
Voice Interaction	Speech-to-text & text-to-speech	Azure Speech SDK (STT + TTS)
Agent Interaction	Multi-step reasoning	Azure GPT with function calling
UI	Interactive interface	Streamlit
Prompt Engineering	Structured logic	Role prompts + tools JSON
Secure Config	Credential handling	Python-dotenv / Streamlit secrets

UI Interaction Flow

1

Recipe Setup Phase

- Generate a new recipe by entering cooking preferences such as meal type, cooking level, available time, dietary restrictions, and optionally a list of available ingredients
- Or choose an existing recipe from the user's personal history of saved dishes

2

Interactive Cooking Experience

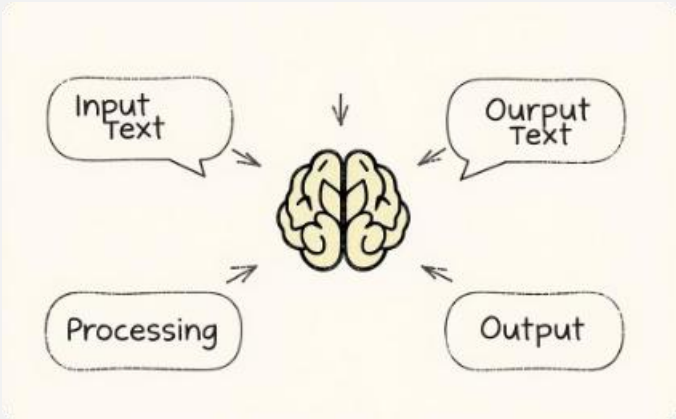
- Recipe steps are read aloud using voice output
- Real-time cooking guidance through fully voice-based interaction
 - System answers user questions and guides through the process dynamically

3

Feedback and Learning Phase

- User can mark recipe as “Liked” or “Cooked”
 - Submit feedback on recipe quality or suggestions
- Data is saved to local database for future personalization

GenAI Methods



Recipe NLP

Prompt design via Azure GPT-4 for natural natural language processing.



Agent Logic

Custom Python logic for handling voice commands effectively.



Voice Output+Input

Utilizing Azure Speech Synthesizer (TTS) and Azure Speech Recognizer (STT).



Recipe Parsing

Prompt-based segmentation for structured structured recipe output.



Feedback Memory

Local JSON logs for feedback, with Firebase integration planned.

Progress So Far

Completed

- Azure GPT-4 integration
- Streamlit UI prototype
- GPT-powered recipe generation
- Initial wireframe finalized
- Agent with substitution, repetition, and skipping capabilities
- Local feedback logging implemented

In Progress

- TTS module integration finalization
- Optional: add support for parsing free-text recipe requests (e.g., "I want a vegan pasta in 20 minutes") using GPT or NLP techniques to extract intent and constraints
- Development of the full user-facing UI (visual layout, branding, responsiveness) – not yet implemented in Streamlit

Demo Preview



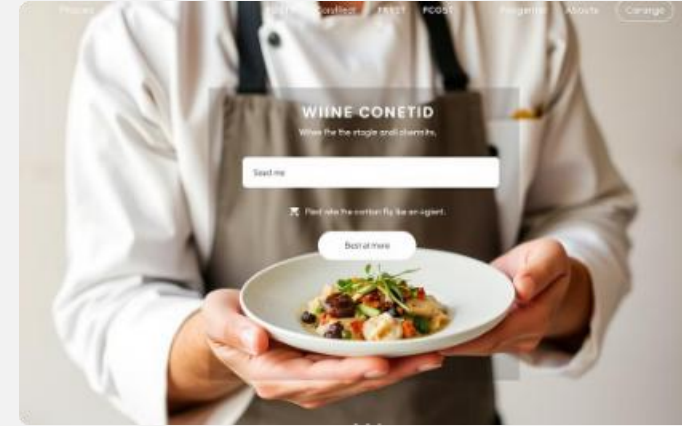
Enter Cooking Preferences

User enters cooking preferences (e.g., meal type, time, dietary needs) or selects available ingredients.



AI Generates Recipe

The agent constructs a structured prompt and sends it to the GPT model to create a personalized personalized recipe.



Recipe Presentation

A personalized recipe is generated and presented to the user.



Interactive Cooking Guidance

The interactive cooking phase begins: the system reads steps aloud and guides the user in real time, responding to voice questions contextually.



Feedback and Saving

User provides feedback at the end of the session, and the recipe is optionally saved to their personal history.

Summary

Su-Chef demonstrates how GenAI can be practically applied to enhance everyday tasks like cooking. Our project integrates personalized recipe generation, voice-based interaction, and user feedback learning into a cohesive experience. With much of the core functionality in place, the next steps involve refining the interface and evaluating user experience in real-world usage.