
Work Proposal: Cooking Assistant Chatbot with Persona

Prepared by: [Ahmed Elsayed]

GitHub link: [Personalized Chatbot](#)

HuggingFace link: [finetuned-gpt2](#)

1. Project Overview

The proposed project aims to design and develop an intelligent **Cooking Assistant Chatbot** that provides recipe guidance, cooking tips, ingredient substitutions, and meal planning support. Unlike a generic chatbot, this system will feature a **persona-driven conversational style** (e.g., a friendly chef or cooking mentor) to make interactions more engaging and relatable.

The chatbot will leverage **transformer-based AI models** (GPT, T5, BERT), **attention mechanisms** for handling multi-turn conversations, and a **continuous learning pipeline** to adapt based on user feedback. MLOps practices will ensure scalable deployment, version control, and continuous monitoring.

2. Objectives

- Build a chatbot specialized in cooking and recipes with a persona.
- Enable dynamic, context-aware, multi-turn conversations.
- Provide recipe suggestions, ingredient substitutions, and personalized cooking advice.
- Integrate continuous learning and reinforcement from user feedback.
- Deploy on cloud platforms with CI/CD pipelines for automation and scalability.
- Deliver a final presentation and live demo showcasing real-time interactions.

3. Work Breakdown Structure

Data Collection and Preprocessing

- Collect cooking-related conversational data and recipe datasets.
- Clean and preprocess text (tokenization, normalization, lemmatization).
- Deliverables: Preprocessed dataset and documentation.

Chatbot Development and Training

- Fine-tune transformer-based models (GPT, T5, BERT) on cooking dataset.
- Add persona-driven response generation.
- Evaluate model performance with BLEU, ROUGE, and Perplexity.
- Deliverables: Trained chatbot and performance evaluation report.

Advanced Techniques and Integration

- Enhance multi-turn dialogue handling with attention mechanisms.
- Integrate a continuous learning pipeline for real-time adaptation.
- Deliverables: Enhanced chatbot with feedback integration pipeline.

MLOps and Deployment

- Track model versions and experiments with MLflow.
- Deploy chatbot on cloud (AWS, Azure, or GCP).
- Expose as REST API via Flask/FastAPI.
- Implement CI/CD pipeline with GitHub Actions or Jenkins.
- Deliverables: Deployed chatbot with monitoring and CI/CD pipeline.

Final Report, Presentation, and Demonstration

- Prepare a comprehensive report covering project lifecycle.
- Present findings and demonstrate chatbot capabilities in real time.
- Deliverables: Final report, presentation slides, live demo.

4. Responsibilities Allocation

Description	Leader	Team Members
Data Collection and Preprocessing	Ahmed Mohamed Abdo	Ihab Hamed Abdelfattah, Ahmed Mohamed Mahmoud, Mostafa Omar Saeed, Ebrahim AbdElrahman Hashem
Chatbot Development and Training	Ihab Hamed Abdelfattah	Ahmed Mohamed Mahmoud, Mostafa Omar Saeed, Ahmed Mohamed Abdo, Ebrahim AbdElrahman Hashem
Advanced Techniques and Integration	Mostafa Omar Saeed	Ihab Hamed Abdelfattah, Ahmed Mohamed Mahmoud, Ahmed Mohamed Abdo, Ebrahim AbdElrahman Hashem
MLOps and Model Management	Ebrahim AbdElrahman Hashem	Ihab Hamed Abdelfattah, Ahmed Mohamed Mahmoud, Mostafa Omar Saeed, Ahmed Mohamed Abdo
Final Report, Presentation, and Demonstration	Ahmed Mohamed Mahmoud	Ihab Hamed Abdelfattah, Mostafa Omar Saeed, Ahmed Mohamed Abdo, Ebrahim AbdElrahman Hashem

5. Workflow (Code Cycle Style)



The development of the **AI Model Chatbot** will follow an **iterative, six-cycle approach**. Each cycle spans approximately **two weeks**, with clearly defined objectives and deliverables. This structure ensures that the chatbot evolves gradually, allowing for continuous improvement, regular evaluation, and incorporation of feedback at every stage.

6. 6 Code Cycles Development Timeline

Cycle	Duration	Focus Areas
Cycle 1	Weeks 1–2	Collect initial dataset, build baseline chatbot, deploy minimal version
Cycle 2	Weeks 3–4	Improve preprocessing, fine-tune with cooking dataset, evaluate
Cycle 3	Weeks 5–6	Integrate continuous learning pipeline, reinforce with feedback
Cycle 4	Weeks 7–8	Add attention mechanisms for multi-turn cooking conversations
Cycle 5	Weeks 9–10	Deploy improved model with persona dialogues, add monitoring, CI/CD automation
Cycle 6	Weeks 11–12	Final refinements, prepare report, presentation, live demo

7. Tools & Technologies

- **NLP & Modeling:** Hugging Face Transformers, PyTorch/TensorFlow
- **Data Processing:** Python, NLTK, SpaCy
- **MLOps & Deployment:** MLflow, Docker, GitHub Actions, Flask/FastAPI, AWS/Azure/GCP
- **Pipeline Orchestration:** Apache Airflow, Kubeflow

8. Expected Outcomes

- A **Cooking Assistant Chatbot with Persona** capable of personalized, dynamic cooking support.
- Ability to handle complex cooking queries and suggest creative recipe ideas.
- Continuous improvement through feedback-driven learning.
- Cloud deployment with CI/CD ensuring scalability, monitoring, and reliability.
- A final report, presentation, and live demo validating the project success.