PROJECT PROPOSAL

LLM Feedback Console

Sigbha Manzoor (Group lead)

Abdul Moiz Arsalan

Muzammil Khan

AUGUST 27, 2024

PROSENSIA SMC PVT LIMITED

Project Proposal: LLM Feedback Console

Overview

The objective of this project is to create an interactive feedback system that allows users to submit feedback on answers generated from their queries. This system utilizes DuckDuckGo for performing searches and generating responses and collects user feedback to improve future results. The feedback, including ratings and comments, is stored in a CSV file, and statistical analysis can be performed to gauge user satisfaction.

Project Components

1. Main Script (main.py)

The main script facilitates the entire user interaction process. It performs the following tasks:

- **Query Input**: Takes a query input from the user.
- **Search and Response**: Utilizes the DuckDuckGo API to search for information and generate an answer based on the user's query.
- **Feedback Collection**: Prompts the user to provide feedback on the provided answer, including a satisfaction rating and, if needed, additional comments or suggestions.
- **Feedback Storage**: Saves the feedback data to a CSV file located in a designated directory.

2. Feedback Module (functions/take_feedback.py)

This module contains the feedback() function responsible for:

- **Collecting User Feedback**: Asks users to rate their satisfaction from 1 to 5. If the rating is 3 or lower, the user is prompted to provide additional comments or suggestions.
- **Returning Feedback Data**: Returns a dictionary containing the user's rating and comments, which is then stored in a CSV file.

3. Feedback Statistics Script (feedback_stats.py)

This script is designed to analyze the feedback data collected in the CSV file. It performs the following:

- **Reads Feedback Data**: Loads the CSV file containing user feedback.
- **Calculates Statistics**: Computes the average satisfaction rating and lists all feedback entries with their corresponding comments.
- **Displays Results**: Outputs the calculated statistics and detailed feedback to the user.

4. Feedback Data Storage (Feedback data/Feedback.csv)

The feedback data is stored in a CSV file located in a directory named "Feedback data." The system automatically creates the directory and file if they do not exist. This file serves as the primary data source for calculating feedback statistics.

Key Features:

- **Automated Feedback Collection**: The system automatically prompts users for feedback after providing an answer, ensuring consistent data collection.
- **Dynamic Feedback Storage**: Feedback is stored securely in a CSV file, which is dynamically created and updated as needed.
- **Statistical Analysis**: The system provides an easy way to analyze feedback data, calculate average ratings, and review detailed comments, enabling continuous improvement.
- **Error Handling and Validation**: The system includes comprehensive error handling to ensure smooth operation, including input validation and file management.

Implementation Plan

1. **Setup Project Directory**: Create the required project directory and files.
2. **Develop Main Script**: Implement the main script to handle user input, perform searches, provide answers, and collect feedback.
3. **Create Feedback Module**: Develop the feedback module to handle user feedback collection and return structured data.
4. **Implement Feedback Storage**: Ensure feedback data is stored correctly in a CSV file, with automatic file creation if necessary.
5. **Develop Feedback Statistics Script**: Create the script to read feedback data and calculate statistics.

6. **Testing and Validation**: Conduct thorough testing to ensure all components work together seamlessly and handle edge cases appropriately.

Project Phases: Planning and Implementation:

Day 1: Project Setup and Initial Development

Tasks:

1. **Set Up Project Environment:**
    - Create the project directory structure.
    - Install necessary libraries (e.g., pandas, duckduckgo_search).
    - Initialize a Git repository for version control (optional but recommended).
2. **Develop the Main Script (main.py):**
    - Implement code to accept user input for queries.
    - Integrate DuckDuckGo API to perform searches and generate answers.
    - Display the generated answer to the user.
3. **Begin Feedback Collection Module (functions/take_feedback.py):**
    - Start developing the feedback() function to prompt the user for a satisfaction rating (1-5).

Day 2: Complete Feedback Collection and Storage

Tasks:

1. **Finalize Feedback Collection Module (functions/take_feedback.py):**
    - Complete the feedback() function to handle input validation and collect additional comments if the rating is 3 or lower.
    - Ensure feedback is returned in a structured format (dictionary).
2. **Implement Feedback Storage:**
    - Write code in main.py to check if the "Feedback data" directory and Feedback.csv file exist.

- Develop logic to create the directory and file if they do not exist.
- Implement functionality to append feedback data to the Feedback.csv file.

Day 3: Develop Feedback Statistics and Testing

Tasks:

1. **Develop Feedback Statistics Script (feedback_stats.py):**
   - Write a script to read feedback data from Feedback.csv.
   - Implement functions to calculate the average satisfaction rating.
   - Add functionality to display each feedback entry with its corresponding comment.
2. **Testing and Debugging:**
   - Conduct unit testing on the feedback() function and feedback storage.
   - Perform basic integration testing to ensure the main script works with the feedback module and storage.
   - Debug any issues identified during testing.

Day 4: Final Testing, Documentation, and Review

Tasks:

1. **Final Testing and Validation:**
   - Conduct end-to-end testing of the entire system to ensure all components work seamlessly together.
   - Validate all user inputs and outputs to ensure a smooth user experience.
2. **Documentation:**
   - Writing Project Final report explaining different parts of project and adding how to run project
   - Document all functions and modules with comments for maintainability.

3. **Project Review and Handover:**
    - Review the project with the internship supervisor or mentor.
    - Make any final adjustments based on feedback.
    - Prepare the project for submission or handover.

Timeline Overview:

Days	Tasks	Completion
Day 01	Project Initial Development of 'main.py' and take feedback.py	Basic Project Setup and initial scripts
Day 02	Complete the 'feedback.py' and feedback storage.	Completed feedback collection and storage mechanism
Day 03	Develop 'feeback_stats.py' testing and debugging	Functional statistics scripts and tested components
Day 04	Final testing, documentation and project review	Finalized Project with documentation and review summary

Conclusion

This project aims to enhance the quality of automated responses by integrating a robust feedback system. By collecting and analyzing user feedback, the system can provide more accurate answers and improve overall user satisfaction. The proposed system is simple, efficient, and designed to be easily maintainable, providing a scalable solution for continuous improvement in answer quality.