**Voice-Driven Form-Filling Solution for the Banking Sector**

**1. Project Overview**

This project aims to streamline and modernize the process of form-filling by introducing a voice-driven solution powered by Azure OpenAI and Azure Speech Services. This application reduces the time required for form completion, particularly for frontline workers in banking, and helps overcome language barriers, making it accessible to a diverse user base. The solution’s flexibility allows it to be adapted for various workflows beyond the banking sector, such as job applications and digital identity verification.

**2. Objectives**

* Develop a user-friendly, voice-driven form-filling solution for the banking sector.
* Leverage Azure Speech and Azure OpenAI to capture, interpret, and validate spoken input for banking forms.
* Improve workflow efficiency by reducing form-filling time and overcoming language barriers through multilingual support.
* Extend the solution for use in other sectors or tasks that involve form completion, including job applications, digital identity creation, and account opening.

**3. Methodology**

**3.1 Technology Stack**

* **Azure Speech Service**: For real-time voice-to-text transcription and multilingual capabilities.
* **Azure OpenAI**: Using GPT models to interpret, validate, and refine input data for specific form fields.
* **Python**: As the primary programming language for integrating APIs, managing workflow, and structuring form-filling logic.
* **JSON**: Used for form templates to enable flexibility and easy customization.

**3.2 Project Structure**

The project is organized as follows:

* main.py: The main script to run the application.
* config.py: Contains configuration for Azure API keys and settings.
* form\_fill.py: Logic for form processing, interacting with GPT, and validating data.
* voice\_input.py: Captures voice input, processes it with Azure Speech, and handles multilingual translation.
* templates/: Contains JSON form templates for specific workflows (e.g., bank account form).

**3.3 Workflow**

1. **Voice Input**: Users respond to form prompts using voice commands. The application captures and transcribes this input in real-time.
2. **Data Interpretation**: Transcribed text is processed by GPT to interpret and autofill form fields according to the question asked.
3. **Form Completion and Verification**: Completed form data is displayed to the user for verification, ensuring accuracy and completeness.
4. **Output**: The final form can be saved, stored, or transmitted as needed by the bank’s systems.

**4. Implementation**

**4.1 Initial Setup**

1. **Azure Services Configuration**: Set up Azure Speech Service and Azure OpenAI. Obtain API keys and configure endpoints in config.py.
2. **Python Environment**: Install the required libraries using requirements.txt.

**4.2 Key Modules**

1. **Voice Input (voice\_input.py)**:

# Recognize and capture speech

result = recognizer.recognize\_once()

1. **Form Filling (form\_fill.py)**:

Interacts with GPT to interpret and complete form fields based on voice input.

# Generate form field completion with GPT

response = openai.Completion.create(...)

1. **Main Application (main.py)**: Guides the user through the form, integrates voice input, and stores the completed form.

# Loop through form fields and prompt for input

user\_response = recognize\_speech()

**4.3 JSON Form Templates**

Templates are stored in the templates/ directory as JSON files to standardize and organize form fields, enabling flexibility across different types of forms.

Example template (bank\_account.json):

{

"first\_name": {"question": "What is your first name?"},

"last\_name": {"question": "What is your last name?"},

"address": {"question": "Please provide your current address."},

"dob": {"question": "What is your date of birth?"}

}

**4.4 Running the Application**

1. Activate the virtual environment and install dependencies.
2. Run main.py to initiate the form-filling workflow, following prompts to fill each form field using voice input.

**5. Outcomes and Benefits**

* **Reduced Time for Form Filling**: Voice input speeds up the process significantly for users.
* **Enhanced Accessibility**: Multilingual support enables a wider audience to complete forms in their preferred language.
* **Improved Data Accuracy**: Real-time feedback and validation through GPT improve data quality and reduce entry errors.
* **Scalability**: The solution’s modular structure makes it adaptable for other workflows and sectors.

**6. Future Enhancements**

* **User Interface**: Develop a GUI or mobile application to make the solution more accessible.
* **Data Privacy and Security**: Implement secure data transmission and storage protocols.
* **Offline Capability**: Enable offline data collection and syncing for use in areas with limited connectivity.
* **Integration with CRM**: Link with existing bank customer relationship management (CRM) systems to streamline data flow.

**7. Conclusion**

The Voice-Driven Form-Filling Solution offers a transformative approach to data collection in the banking sector, enhancing productivity, accessibility, and data accuracy. Through Azure Speech and OpenAI, the solution allows banks to cater to diverse user needs while improving operational efficiency. With future improvements, this solution could be widely adopted in various sectors, streamlining workflows across industries.

**My Project Link**

https://github.com/Sahil4000/BankingProject.git