

Smart AI-Based Interview Platform

Detailed Technical Documentation

1. Project Overview

The Smart AI-Based Interview Platform is designed to automate and enhance the technical interview process using Artificial Intelligence. The system enables candidates to participate in interviews where responses can be submitted as text or audio. The platform processes responses using AI models to generate structured evaluation feedback, scoring metrics, and multilingual support through translation and transcription services.

2. System Architecture

The platform follows a layered architecture integrating Frontend, Azure API Management (APIM), FastAPI backend services, and AI processing components. The high-level flow is as follows:

- Frontend Application sends interview requests and candidate responses.
- Azure API Management (APIM) acts as a secure gateway handling routing, policies, and authentication.
- FastAPI backend processes business logic and integrates AI services.
- Speech-to-Text service converts audio responses into text.
- Translation service enables multilingual support.
- LangChain pipelines process responses and generate structured AI evaluation output.

3. Backend API Development (FastAPI)

RESTful APIs were developed using FastAPI to manage interview sessions, candidate responses, evaluation processing, and system integration. Pydantic models were used for strict request and response validation ensuring structured data exchange.

- Interview session creation and management APIs.
- Candidate response submission endpoints (text and audio).
- Validation using Pydantic schemas.
- Exception handling and standardized error responses.
- Swagger/OpenAPI documentation for API testing and visibility.

4. AI Transcription and Translation Integration

To support multilingual interviews and audio-based responses, transcription and translation services were integrated into the evaluation pipeline. Audio inputs are first converted to text using Speech-to-Text services, after which translation is applied if required.

- Audio-to-text conversion using Speech-to-Text service.
- Language detection and multilingual translation support.
- Integration of transcription output into evaluation workflow.
- Error handling for unsupported languages and invalid audio inputs.

5. LangChain Integration for AI Evaluation

LangChain was used to structure and optimize interactions with Large Language Models (LLMs). Structured Output, Output Parsers, Chains, and Runnables were implemented to create reliable evaluation pipelines.

- Structured output using defined schemas for consistent evaluation responses.
- Output parsers for converting model responses into JSON format.
- Chain-based workflows for multi-step processing.
- Runnable pipelines for modular and scalable AI evaluation logic.
- Prompt optimization for improved scoring accuracy.

6. Azure API Management (APIM) Configuration

Azure API Management was configured to expose backend APIs securely. Policies were applied to enforce authentication, rate limiting, CORS handling, and request validation.

- Importing backend APIs into APIM.
- Inbound and outbound policy configuration.
- CORS configuration for frontend integration.
- Subscription key validation and security policies.
- Rate limiting and traffic control.

7. Testing and Validation

Comprehensive testing was performed across the entire pipeline to ensure system reliability, performance, and response accuracy.

- Unit testing of APIs using Swagger and Postman.
- End-to-end integration testing (Frontend → APIM → Backend → AI Services).
- Validation of structured AI responses.
- Performance testing and response time analysis.
- Bug fixes and optimization of prompts and parsers.