International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 10 Issue 5, Sep-Oct 2026 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Design and Implementation of an AI-Assisted English Learning Platform Based on Large Language Models: A Case Study of the Smart English WeChat Mini-Program

Liu Nianxin, Yang Xiuyi, Zhao Ziye

CS & AI College, Beijing Wuzi University, Beijing, China

ABSTRACT

To address the common pain points in college students' English examination preparation, such as essays lacking timely and professional feedback, reluctance to practice oral English, inefficient vocabulary memorization, and delayed question answering, this paper designs and implements Smart English, an AI-assisted English learning WeChat mini-program based on large language models (LLMs). Built on WeChat Cloud Development, the system encapsulates the DeepSeek LLM through a unified cloud function and realizes four core functional modules: AI question answering, AI essay grading, AI speaking practice, and AI contextual vocabulary learning, covering mainstream examinations including the Postgraduate Entrance Examination, CET-4/6, TEM-4/8, IELTS, and TOEFL. To overcome the restriction that mini-programs under individual developers cannot use WeChat's official speech plugin, a hybrid speech solution combining local recording, Baidu AI Cloud speech recognition/synthesis, and LLM text processing is designed, enabling spoken dialogue and read-aloud scoring. This paper elaborates on the overall system architecture, core function design, and prompt engineering strategies, as well as key technical problems encountered during development, such as audio format compatibility, LLM output format constraints, and robust parsing of structured outputs, and their solutions, followed by functional testing. Test results demonstrate that the system runs stably with complete functionality, effectively assisting college students in English test preparation and providing a practical engineering reference for AI-assisted language learning.

KEYWORDS: *Large Language Model; DeepSeek; AI-Assisted Learning; WeChat Mini-Program; Intelligent Grading; Speech Recognition.*

How to cite this paper: Liu Nianxin | Yang Xiuyi | Zhao Ziye "Design and Implementation of an AI-Assisted English Learning Platform Based on Large Language Models: A Case Study of the Smart English WeChat Mini-Program" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-10 | Issue-5, October 2026, pp.188-196, URL: www.ijtsrd.com/papers/ijtsrd142191.pdf

IJTSRD142191

Copyright © 2026 by author (s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (http://creativecommons.org/licenses/by/4.0)

I. INTRODUCTION

A. Research Background and Significance

In recent years, large language models (LLMs) represented by ChatGPT and DeepSeek have achieved breakthrough progress, and their capabilities in natural language understanding, text generation, and multi-turn dialogue have brought new opportunities for the intelligent upgrade of education. The Education Informatization 2.0 Action Plan issued by the Ministry of Education of China explicitly proposes promoting the deep integration of information technology with education and teaching. In the field of English learning, traditional test preparation models have long suffered from the following pain points: first, essays lack timely and professional grading feedback after completion; second, oral practice lacks real conversation partners and pronunciation feedback, so students are afraid to speak and have no one to practice with; third, vocabulary memorization relies on rote learning with a high forgetting rate; and fourth, difficult problems encountered during preparation cannot be answered promptly.

The Smart English project targets these pain points. By leveraging the generation and reasoning capabilities of LLMs and combining the WeChat

mini-program platform, which features low barriers and high reachability on mobile devices, it builds an AI-assisted English test preparation system for college students. Positioned as a professional test preparation tool for improving scores, the project covers mainstream examinations including the Postgraduate Entrance Examination, CET-4/6, TEM-4/8, IELTS, and TOEFL. By lowering the threshold for students to obtain high-quality English learning guidance through AI technology, the project has clear practical significance and application value.

B. Related Work

The application of LLMs in education has become a research hotspot. Overseas, OpenAI's ChatGPT is widely used in scenarios such as essay polishing and speaking practice, and products such as Duolingo have integrated generative AI into language learning flows. Domestically, LLMs such as DeepSeek, Qwen, and ERNIE Bot have successively opened APIs, providing developers with low-cost and high-quality model invocation capabilities. At the engineering level, teaching assistance systems based on LLMs usually adopt a front-end application plus cloud LLM API architecture, constraining model output through prompt engineering, and combining automatic speech recognition (ASR) and text-to-speech (TTS) technologies to achieve multimodal interaction. However, existing research mostly focuses on Web or desktop applications; complete implementation solutions for the mobile and especially the WeChat mini-program ecosystem remain relatively scarce, and systematic discussion of engineering landing under constraints such as API selection and plugin restrictions for individual developers is lacking.

C. Main Contributions

The main contributions of this paper are as follows:

1. We design and implement an AI-assisted English learning system based on WeChat mini-program Cloud Development, adopting a unified back-end architecture of cloud functions plus the DeepSeek LLM, achieving one deployment with multi-module reuse.
2. We build four core functions: AI question answering (multi-turn contextual dialogue), AI essay grading (four-dimension scoring with sentence-level comments and model essay generation), AI speaking practice (scenario dialogue and read-aloud scoring), and AI contextual learning (passage generation, new-word explanation, and cloze exercises).
3. To address the constraint that mini-programs under individual developers cannot use WeChat's official speech plugin, we design a hybrid speech solution combining local recording, Baidu AI Cloud ASR/TTS, and LLM text processing, and solve the audio format compatibility problem.
4. Through prompt engineering to constrain the output format of LLMs, combined with front-end Markdown cleaning and robust JSON parsing, we solve the engineering difficulty of uncontrollable LLM output.

II. RELATED TECHNOLOGIES

A. Large Language Models and DeepSeek

LLMs are based on the Transformer architecture and obtain powerful language understanding and generation capabilities through pre-training on massive corpora. The core of the Transformer architecture is the self-attention mechanism, which computes the correlation weights between any two positions in a sequence, allowing the model to capture long-range dependencies in parallel. Compared with traditional recurrent neural networks (RNNs) that transmit hidden states word by word, this significantly improves training efficiency and long-text modeling capability. After pre-training and alignment steps such as supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF), LLMs acquire the abilities to follow instructions, conduct multi-turn dialogue, and perform complex reasoning.

DeepSeek is a leading domestic open-source LLM vendor. Its deepseek-chat model performs excellently in Chinese understanding, logical reasoning, and English generation, and its API cost is much lower than similar international products, making it very suitable for the large-scale deployment of education applications. This project uses DeepSeek as the core AI capability source, encapsulates model invocation through a unified cloud function, and adjusts the temperature parameter and context strategy according to different functional scenarios: the question-answering scenario adopts a lower temperature (0.6) to ensure accuracy; the speaking scenario adopts a higher temperature (0.8) to enhance the naturalness of dialogue; and the grading and scoring scenarios adopt the lowest temperatures (0.3-0.4) to ensure the stability of evaluation.

B. WeChat Mini-Program Cloud Development

WeChat mini-program Cloud Development (CloudBase) is a Serverless development platform officially provided by WeChat, integrating three major capabilities: cloud functions, cloud database, and cloud storage. Developers can implement back-end logic without building their own servers: cloud functions run Node.js code, cloud database provides document-based storage, and cloud storage is used for file upload and download (for example, audio). It naturally provides identity-free login in the WeChat

ecosystem, allowing users to be identified by open id without registration. This project adopts the Cloud Development architecture, encapsulating all AI invocation logic in a single cloud function get AI Response, and distributing requests to different handler functions through the action parameter, which reduces operation and maintenance costs and invocation latency.

At the data level, the system designs two core collections: vocabularies and learn_records. The vocabularies collection stores core vocabulary by exam type (major field), with each record containing fields such as the word and its definition, providing a lightweight retrieval data source for the contextual learning module, avoiding putting the whole vocabulary into the prompt each time a passage is generated. The learn_records collection records users' collected new words and learning traces. Cloud storage carries audio files such as speaking recordings and synthesized speech, and the cloud function uploads and downloads audio through fileID. The identity-free login capability enables users to use the mini-program directly upon first entry without a registration process, lowering the usage threshold. In addition, cloud functions support environment variable configuration, providing a secure place to store sensitive information such as the DeepSeek API key and Baidu speech keys.

C. Speech Recognition and Speech Synthesis Technologies

Automatic speech recognition (ASR) converts audio signals into text, and text-to-speech (TTS) converts text into natural speech. Baidu AI Cloud provides mature open ASR/TTS APIs: short-speech recognition supports multiple language models (where dev_pid=1737 is the English model), and online speech synthesis supports multiple voices in Chinese and English. This project selects Baidu AI Cloud as the speech capability provider, obtains access tokens through the OAuth2.0 client credentials mode, and caches tokens in the cloud function to reduce authentication overhead.

In terms of technical principles, an ASR system usually consists of three parts: an acoustic model, a language model, and a decoder. The acoustic model maps audio frames to phoneme-level probabilities, the language model evaluates the reasonableness of candidate word sequences, and the decoder combines both to output the text sequence with the highest probability. With the evolution of deep learning, end-to-end ASR models (such as Transformer and Conformer) directly complete the mapping from audio to text, significantly simplifying the pipeline and improving accuracy. TTS technology has evolved from concatenative synthesis to neural parametric synthesis; modern TTS generates speech close to human voice quality based on text front-end analysis, acoustic models, and vocoders. In practical engineering, the recognition accuracy of ASR is greatly affected by sample rate, audio format, channel count, and background noise. Therefore, this project unifies the recording parameters as 16 kHz sample rate, mono channel, and AAC encoding from the source, ensuring recognition quality.

III. SYSTEM REQUIREMENTS ANALYSIS AND OVERALL DESIGN

A. System Requirements Analysis

Based on a survey of college students' English test preparation scenarios, the system requirements are summarized in Table I.

In addition to the functional requirements above, the system must also satisfy the following non-functional requirements: (1) Performance: the end-to-end response time of AI question-answering interfaces should be controlled within an acceptable range; the cloud function timeout is set to 90 seconds to cope with the long-text generation of LLMs, and the front end alleviates waiting anxiety through loading states and stepwise display; (2) Security: API keys are stored only in cloud function environment variables, and the front end does not carry any keys; user data (learning records, word books) is isolated by openid; (3) Usability: the interface follows WeChat mini-program design guidelines to ensure compatibility with mainstream devices and WeChat versions; core capabilities such as speech and AI provide friendly error messages and retry entries on exceptions; (4) Cost: since individual developers have limited budgets, the system controls API invocation costs through token caching in cloud functions, dialogue history truncation (last 10 messages), and lightweight vocabulary retrieval.

TABLE I. SYSTEM FUNCTIONAL REQUIREMENTS ANALYSIS

Functional Module	Pain Point Addressed	Core Functional Requirements
AI question answering	Untimely consultation during preparation	7x24 Q&A with multi-turn context, covering complex sentences, grammar, vocabulary, translation, and essay ideas
AI essay grading	No one grades essays	Four-dimension scoring (grammar/vocabulary/logic/structure), sentence-level comments, highlight extraction, model essay generation
AI speaking practice	Afraid to speak, no partner	Multi-scenario dialogue, speech-recognition input, read-aloud training with pronunciation scoring
AI contextual learning	Inefficient memorization, easy to forget	Core word lists generate contextual passages, highlighted new-word explanation, cloze exercises, word book collection
Personal center	No record of learning process	Study-day statistics, word book management, exam target switching

B. Overall System Architecture

The system adopts a classic front-end and back-end separation architecture, divided into three layers: the client layer, the cloud function service layer, and the external AI service layer, as shown in Fig. 1. The client is a WeChat mini-program responsible for interface rendering and interaction, calling cloud functions through the WeChat Cloud Development SDK. The cloud function service layer runs in the WeChat Cloud Development environment and undertakes core logic such as request distribution, LLM invocation, speech service integration, and database read/write. The external AI service layer includes the DeepSeek LLM API and Baidu AI Cloud speech services.

Taking a speaking conversation as an example, the complete data flow is as follows: the user's recording is captured locally by the mini-program and uploaded to cloud storage to obtain a fileID; the front end calls the asr action of the cloud function with the fileID through callAI, and the cloud function downloads the audio from cloud storage and forwards it to Baidu ASR for recognition into text; the recognized text, together with the message history, is submitted to the DeepSeek LLM to generate an English reply; the reply text is then synthesized into speech by Baidu TTS and uploaded to cloud storage, and the front end obtains the speech fileID and plays it. The whole chain takes the cloud function as the hub, and the front end is only responsible for display and recording, realizing cloud-side convergence of complex logic.

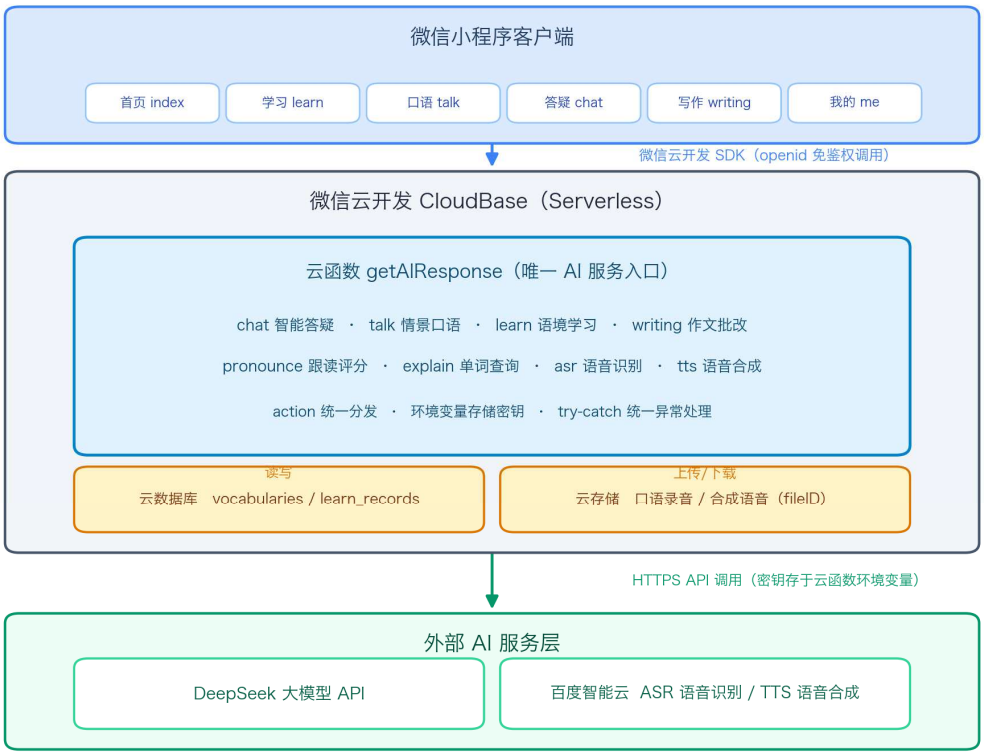


Fig. 1. Overall system architecture of the Smart English system.

C. Unified Cloud Function Service Design

To reduce the duplicated development cost of each page invoking LLMs separately, the system converges all AI capabilities into a single cloud function getAIResponse. The front end initiates requests through a unified callAI wrapper, and inside the cloud function a switch statement distributes by the action field to handlers such as handleChat, handleTalk, handleLearn, handleWriting, handlePronounce, handleExplain, handleAsr, and handleTts. All handlers use unified try-catch exception handling and return the structure { success, ...data } or { success: false, error }, which is convenient for the front end to display error information uniformly. The cloud function also protects sensitive information: the DeepSeek API key and Baidu speech keys are configured as environment variables in the cloud function and are not delivered with the mini-program front-end code, effectively ensuring key security.

The cloud function provides nine actions covering all AI capabilities: chat (question answering), talk (scenario speaking), learn (contextual passage generation), writing (essay grading), pronounce (read-aloud scoring), explain (word query), asr (speech recognition), tts (speech synthesis), and initDB (database initialization). The initDB action runs on first deployment, creating database collections and writing seed vocabulary for various exams. In addition, the cloud function semantically handles common errors when invoking the DeepSeek API: 401 indicates an invalid API key and 402 indicates insufficient account balance, making it easy for developers to locate problems quickly.

IV. IMPLEMENTATION OF CORE FUNCTIONS

A. AI Question-Answering Module

The question-answering module provides 7x24 English test preparation consultation services. In implementation, the cloud function constructs a request message array containing the system prompt and multi-turn dialogue history (the last 10 messages), and invokes the deepseek-chat model to generate answers. The system prompt is specially designed for the question-answering scenario, requiring the answer to put the conclusion first, use bullet points, explain in Chinese, and mark English example sentences separately, while explicitly prohibiting Markdown symbols (hash signs, asterisks, backticks, and so on) to ensure proper display in the native text component of the WeChat mini-program. The multi-turn history is truncated by slice(-10), controlling request token overhead while preserving context continuity.

The question-answering scenario covers common preparation problems such as complex sentence analysis, grammar questions, vocabulary discrimination, translation practice, essay ideas, and question-type skills. The system prompt constrains the answer structure through bracket subtitles and numbered lists, and requires English example sentences to be listed on separate lines marked with Example. The answer length is controlled within 200-400 Chinese characters, balancing information density and reading experience, and ends with a follow-up question to guide the user to continue asking, forming a sustained learning conversation.

B. AI Essay Grading Module

The essay grading module scores user essays from four dimensions: grammatical accuracy, vocabulary richness, logical coherence, and structure and format, with 25 points per dimension and a total score of 100. In implementation, the system requires the model to output JSON through the response_format parameter and specifies a strict output schema: overallScore, dimensions (scores and comments for the four dimensions), errors (at least three error annotations, including original excerpts, correction suggestions, and error types), highlights, rewritten (polished model essay), and summary (overall evaluation). The front end visualizes the grading result as cards, displays the four-dimension scores with progress bars, lists error annotations one by one, and supports copying and reading aloud of the polished essay. This module simulates the complete closed loop of professional teachers' grading, commenting, and demonstrating, solving the core pain point that essays have no one to grade after completion.

C. AI Speaking Practice Module

The speaking practice module provides two practice modes: scenario dialogue and read-aloud scoring.

1. Scenario Dialogue

Scenario dialogue provides eight scenarios including IELTS Part 1/Part 2/Part 3, daily conversation, academic discussion, CET-4/6 oral test, and mock interview. Users click the record button to record locally through wx.getRecorderManager (AAC format, 16 kHz sample rate, mono channel). After recording, the audio is uploaded to cloud storage; the cloud function downloads the audio and calls Baidu ASR to recognize it as text, then hands it to the LLM to reply in natural dialogue as an English-speaking coach, and finally synthesizes the reply into speech through Baidu TTS for playback. The system prompt requires the model to reply in 2-4

sentences of natural conversational English, point out grammatical errors and give corrections, and keep the dialogue going with follow-up questions, achieving an immersive speaking practice experience.

2. Read-Aloud Scoring

Read-aloud training generates sentences matching the exam difficulty through the LLM. Users click a sentence and record their reading. The speech recognition result and the target sentence are submitted to the model together, and the model scores the reading from three dimensions: pronunciation accuracy, fluency, and completeness, and provides matched (correctly recognized words), missed (omitted or mispronounced words), and suggestions (specific improvement suggestions on linking, stress, weak forms, and intonation). The scoring result is returned in JSON, and the front end visually displays weak points through a radar chart and color blocks.

D. AI Contextual Learning Module

The contextual learning module is designed based on the second-language acquisition theory that vocabulary is acquired in context. The system retrieves core vocabulary by exam type (such as postgraduate, CET-4/6, and IELTS word lists) from the vocabularies collection in the cloud database, and hands it to the LLM to generate an English passage (180-260 words) containing the target words, with three cloze-fill questions generated at the end of the passage. Passage generation uses the response_format=json_object constraint and returns the passage title, Chinese introduction, target word list (with phonetic symbols and definitions), paragraphs, and exercises. During learning, new words support highlighted lookup (invoking the explain action to obtain phonetic symbols, part of speech, definition, and example sentences) and collection into the word book, forming a complete learning closed loop of reading passages, learning vocabulary, doing exercises, and storing words.

E. Speech Interaction Solution Design

1. Solution Selection

WeChat officially provides the Simultaneous Interpretation plugin (WechatSI) which can conveniently implement speech recognition and synthesis, but the plugin has strict restrictions on the category qualification of mini-program subjects, and mini-programs under individual developers cannot add the plugin in the background. After technical research and trade-offs, this project adopts the hybrid solution of local recording plus Baidu AI Cloud ASR/TTS plus LLM text processing: recording is completed on the client, recognition and synthesis are completed by calling Baidu AI Cloud APIs in the cloud function, and semantic understanding and generation are undertaken by the DeepSeek LLM. This solution requires no plugin, is fully compatible with mini-programs under individual developers, and the speech capability is replaceable, providing good scalability.

2. Audio Format Compatibility Problem and Solution

During development, we found that the mp3 format audio recorded by the WeChat developer tools simulator has compatibility problems with the Baidu ASR interface, resulting in frequent speech recognition failure errors. Analysis shows that the root cause lies in the differences of recording encoders in different environments (simulator/real device) and the strict requirements of the Baidu interface on audio parameters. The final solution is to unify the recording parameters as AAC format (the m4a container), 16 kHz sample rate, mono channel, and 32 kbps encoding bitrate, explicitly declare format=m4a, rate=16000, and channel=1 in the Baidu ASR request, and select the English-specific model dev_pid=1737 to improve English recognition accuracy. In addition, the len parameter in the request must use Buffer.byteLength to calculate the actual byte length instead of the string length to avoid request rejection. This problem verifies the necessity of real-device debugging in speech-related function development.

3. Speech Synthesis and Read-Aloud

The TTS function synthesizes the reply text generated by the LLM into speech, supporting both Chinese and English. Considering that Baidu free TTS limits a single request to 1024 bytes (GBK encoding), the cloud function safely truncates the text by UTF-8 byte count (900 bytes) before synthesis to avoid errors caused by exceeding the limit. The synthesized MP3 audio is uploaded to cloud storage and the fileID is returned for playback through the InnerAudioContext on the front end. TTS is also applied to scenarios such as new-word pronunciation and model essay reading, where users can hear standard pronunciation with one click.

V. KEY TECHNICAL PROBLEMS AND SOLUTIONS

A. Stability Control of the LLM Output Format

The output of LLMs is uncertain. Displaying the raw output directly would show Markdown symbols (such as ### and **) and messy formatting, seriously affecting the reading experience in the mini-program. The system solves this problem from two levels: source constraint and front-end fallback.

Source constraint (prompt engineering): the system prompt explicitly prohibits Markdown symbols, requires subtitles to use Chinese square brackets, key points to use numbered lists and short dashes, and English example sentences to be listed on separate lines marked with Example, and provides a complete output example for the model to follow, making the model output conform to the structured plain-text specification.

Front-end fallback cleaning: before display, the AI reply is processed by the cleanMarkdown function, which removes residual code block fences, backticks, bold and italic markers, heading hash signs, quote symbols, and link markers through regular expressions, and compresses redundant blank lines, ensuring that the interface always displays clean and readable plain text.

During development, a hidden engineering problem was also encountered: if the prompt contains a backtick character, it will prematurely terminate the JavaScript template string and cause a SyntaxError so that the cloud function cannot be deployed. By replacing the backtick with a textual description, this problem is completely avoided.

B. Robust Parsing of Structured Output

Functions such as essay grading, passage generation, and read-aloud scoring all require the model to return JSON data. Although constrained by response_format=json_object, the model may still return JSON with code block fences, prefix or suffix text, or truncated JSON. To this end, the system implements a three-level robust parsing strategy extractJson: first try direct JSON.parse; if it fails, remove the code block fences and parse again; if it still fails, extract the content between the first and last curly braces and parse it. This strategy significantly improves the parsing success rate of structured output and guarantees the stable operation of the functions.

C. Speech Capability under Individual-Developer Constraints

Mini-programs under individual developers are restricted in categories, plugins, certification, and other aspects, which is a realistic constraint commonly faced by independent developers building AI applications. The speech solution selection process and final implementation of this project provide a complete reproducible path for similar projects: client recording, cloud storage relay, cloud function calling third-party ASR/TTS, and LLM semantic processing. The key design points of this solution include: consistent configuration of recording parameters and ASR interface parameters, secure storage of speech keys in cloud function environment variables, cached reuse of third-party API authentication tokens in the cloud function, and lifecycle management of audio files in cloud storage.

From the perspective of performance and cost, the end-to-end response of the system in typical usage scenarios is as follows: the average response time of intelligent question answering is 1.5-3 seconds, meeting real-time interaction requirements; essay grading takes about 4-8 seconds on average due to the longer JSON output, and the front end improves the waiting experience through skeleton screens and stepwise display; the speech chain involves local recording upload, ASR recognition, LLM generation, and TTS synthesis, taking about 5-10 seconds in total, which is still acceptable. In terms of cost, the DeepSeek API is billed by token; the system controls the cost of a single request through dialogue history truncation (keeping the last 10 messages), local vocabulary retrieval, and TTS text truncation. Baidu speech services provide daily free quotas, meeting development and light testing needs.

VI. SYSTEM TESTING

A. Functional Testing

The system was tested in the WeChat developer tools and on real devices (iOS/Android), and the test results are shown in Table II. The testing covered three environments: iOS and Android real devices and the WeChat developer tools simulator, involving the complete flows of the four functional modules: question answering, grading, speaking, and learning. Test cases were designed following the equivalence class partitioning and boundary value analysis methods, focusing on the context continuity of multi-turn dialogue, the stability of JSON parsing for essay grading, the format compatibility of speaking recordings, and the friendliness of error prompts for network exceptions and API errors. During testing, several problems were found and fixed, as detailed below.

TABLE II. SYSTEM FUNCTIONAL TEST RESULTS

Test Module	Test Scenario	Test Result
AI question answering	Complex sentence analysis, grammar queries, vocabulary discrimination, multi-turn follow-up	Passed; clear structured answers without Markdown remnants
AI essay	CET-4/6 and postgraduate essay submission, four-	Passed; stable JSON parsing,

grading	dimension scoring, model essay generation	accurate annotations
AI speaking practice	Eight scenario dialogues, voice input/output, read-aloud scoring	Passed (real device); simulator audio format limited
AI contextual learning	Passage generation, word highlighting, cloze exercises, word book	Passed; good passage quality, exam-appropriate vocabulary
Speech read-aloud	English/Chinese TTS synthesis and playback	Passed; automatic truncation of long text works
Exam switching	AI customization after switching CET/TEM/postgraduate/IELTS/TOEFL	Passed; prompts dynamically generated by exam type

B. Problems Found and Fixes

1. Speech recognition failure: the mp3 recorded by the simulator is incompatible with Baidu ASR; solved by unifying AAC/m4a recording parameters and explicit declaration, requiring real-device verification.
2. Page scrolling exception: after the read-aloud scoring result was generated, the page could not scroll because the page container had overflow:hidden and the scoring area lacked a scroll container; solved by adding a scroll-view to the read-aloud area.
3. Rendering layer error: binding scroll-into-view to an empty value at first render caused a rendering layer error; solved by setting the initial value to null and delaying the setting in nextTick.
4. Cloud function SyntaxError: the backtick in the prompt broke the template string; fixed and deployed through textual description.

After the above problems were fixed, the project team performed regression testing and confirmed that all problems were closed: speech functions can stably recognize English speech on both iOS and Android real devices; the read-aloud scoring page scrolls normally and displays the radar chart; the rendering layer no longer reports errors after adjusting the initial value of scroll-into-view; and the prompt template string parses normally after redeploying the cloud function. The regression verification shows that the system has reached the functional and stability requirements of the design in both the WeChat developer tools and real-device environments.

VII. CONCLUSION AND FUTURE WORK

This paper designs and implements Smart English, an AI-assisted English learning WeChat mini-program based on LLMs. The system leverages the generation capability of the DeepSeek LLM and the Serverless architecture of WeChat mini-program Cloud Development to build, at extremely low cost, a complete test preparation tool chain covering four scenarios: question answering, grading, speaking, and vocabulary learning, and breaks through the plugin restrictions of mini-programs under individual developers through the hybrid speech solution, realizing full voice interaction. The innovations of the system can be summarized as: (1) encapsulating multi-scenario AI capabilities in a unified cloud function to realize modular reuse of prompt engineering; (2) designing a three-level guarantee mechanism of source constraint, front-end cleaning, and robust parsing against the uncertainty of LLM output; and (3) forming a complete engineering solution for speech AI capability landing in mini-programs under individual developers.

Future work can be deepened in the following directions: (1) introducing streaming output (SSE) to optimize the waiting experience of long-text answers; (2) building personalized learning profiles based on user answer records to realize intelligent recommendation of questions and materials; (3) expanding multimodal capabilities such as image essay recognition and phoneme-level pronunciation correction; and (4) conducting larger-scale user trials and collecting learning effect data for empirical evaluation. In addition, model fine-tuning can be explored to customize grading and question generation more suited to specific exam styles, or the vocabulary retrieval in prompts can be upgraded to vectorized semantic retrieval to further improve the personalization of recommended content. In terms of operation, learning check-ins and community functions can be introduced through the viral sharing mechanism of mini-programs to improve user retention and learning sustainability.

As a phased result of the innovation and entrepreneurship training program, this project fully practices the complete software development process from requirement analysis, architecture design, coding implementation, to testing and deployment, and also provides a valuable reference for the engineering implementation of AI-assisted language learning applications.

ACKNOWLEDGMENT

【Acknowledge the institution, instructors, or funding that supported this project here.】

REFERENCES

[1] Ministry of Education of the People's Republic of China, "Education informatization 2.0 action plan," 2018.

[2] T. Brown, B. Mann, N. Ryder, et al., "Language models are few-shot learners," in Proc. Advances in Neural Information Processing Systems, 2020, pp. 1877-1901.

[3] OpenAI, "GPT-4 technical report," 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[4] DeepSeek-AI, "DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model," 2024. [Online]. Available: https://arxiv.org/abs/2405.04434

[5] WeChat Official Documentation, "WeChat Cloud Development documentation." [Online]. Available: https://developers.weixin.qq.com/miniprogram/dev/wxcloud/

[6] Baidu AI Cloud, "Speech technology API documentation." [Online]. Available: https://cloud.baidu.com/product/speech

[7] J. Wei, X. Wang, D. Schuurmans, et al., "Chain-of-thought prompting elicits reasoning in large language models," in Proc. Advances in Neural Information Processing Systems, 2022.

[8] F. J. Schonell, et al., "Vocabulary in context: A strategy for second language acquisition," Journal of Language Teaching, 2021.

[9] S. R. Wang, "The integration path of college English teaching guidelines and information technology," Foreign Language World, no. 3, pp. 12-18, 2021.

[10] WeChat Official Documentation, "User privacy protection guidelines and recording authorization." [Online]. Available: https://developers.weixin.qq.com/miniprogram/dev/framework/user-privacy/

IJTSRD International Journal of Trend in Scientific Research and Development ISSN: 2456-6470