

# 多分野対応用語学習アプリケーションの 開発とその効果の検証

Multi-disciplinary terminology learning application  
development and verification of its effectiveness

大竹優輝<sup>1</sup>

Yuki Ohtake<sup>1</sup>

<sup>1</sup>神戸大学附属中等教育学校

<sup>1</sup>Kobe University Secondary School

## Abstract

本研究は、英語学習アプリが多いことに注目し、特定分野に限定されない多分野対応用語学習アプリケーションの開発とその効果の検証を目的に研究を行った。

コード生成 AI を活用して、学習者が任意の用語とその定義を入力すると、それに基づいた用語学習教材を提供できる学習アプリを開発した。

本研究では開発した学習アプリの効果を検証するため、「音楽用語」という分野に注目して神戸大学附属中等教育学校の吹奏楽部および4年2組を対象に実証実験を実施した。実証実験の前後に行った音楽用語のテストの得点を分析した結果、学習アプリには一定の効果が認められ、また効果量 $\varphi$ の値からも、学習アプリの効果が大きいことが示された。加えて、吹奏楽部と4年2組の平均得点に有意な差があるとは判断されなかったが、効果量 Cohen's d の値より中程度の効果があることが示された。

## 目次

第 1 章	学習アプリケーションの現状と仕掛学に基づくアプローチ	1
第 1 節	学習アプリケーションの現状	1
第 2 節	本研究の目的	1
第 3 節	仕掛けの特性	1
第 2 章	仕掛けの概要	2
第 1 節	多分野対応用語学習アプリケーション	2
第 2 節	仕掛けとしての適性	3
第 3 章	実証実験	4
第 1 節	調査の概要と目的	4
第 2 節	調査方法	4
第 3 節	調査結果	4
第 4 節	考察	5
第 1 項	学習アプリが用語学習に与える効果について	5
第 2 項	吹奏楽部と 4 年 2 組の平均得点の差について	5
第 4 章	結論	6
第 1 節	まとめ	6
第 2 節	本研究の課題点と今後の展望	6
謝辞		6
付録 A	調査用紙と音楽用語習熟度テスト	7
付録 B	コード生成 AI を用いた開発環境	9
付録 C	学習アプリのプログラム	10
参考文献		34

## 第1章 学習アプリケーションの現状と仕掛学に基づくアプローチ

### 第1節 学習アプリケーションの現状

近年、学習アプリケーションが急速に普及している。App Storeの「教育」カテゴリは「Duolingo」や「AI 英会話スピーク」などの英語学習アプリケーションが大半を占めており、特に単語学習アプリケーションに注目しても「英語勉強アプリ mikan」や「ターゲットの友 英単語アプリ」などの英単語を対象としたものがほとんどである。[1]これは英語学習に関する利便性が向上する一方で、古典単語や医療用語、音楽用語といった特定分野の学習アプリケーションが少なく、それらを学びたい学習者にとって十分な学習環境が整っているとは言えない。

### 第2節 本研究の目的

本研究では、特定の分野に限定することなく学習者のニーズに応じた用語学習を可能にするアプリケーション（以下、学習アプリと略記）を開発し、またその効果の検証を行うことを目的とした。そこで、以下の2つの問いを立てた。

- (1) 学習アプリは用語学習に効果を与えるのか
- (2) 学習者によって学習アプリの効果に違いはあるのか

また、学習者に能動的な用語学習を促すために大阪大学大学院経済学研究科教授の松村真宏氏が提唱する「仕掛学」[2]の観点から、その仕掛学の特性を活用した学習アプリの開発に取り組んだ。



図1. 机を整列させる仕掛けを設置したようす（出典：[3]）

### 第3節 仕掛学の特性

仕掛学は従来の行動の選択肢を残したまま、新たな行動の選択肢を追加するものである。図1の「机を整列させる仕掛け」を例に挙げると、机を整列させるという従来の行動の選択肢に、床に貼られたシールに机の脚を合わせるという新たな行動の選択肢を追加したことになる。[3]仕掛学では行動変容を強要しないようにするために、従来の選択肢は残しつつ、より魅力的な選択肢を用意する営みであることに注目してもらいたい。また、日常的に用いられる仕掛学という言葉と仕掛学における「仕掛学」の区別として、仕掛学では、FAD要件を全て満たすものが「仕掛学」と定義されている。[4]

- (1) F(Fairness)要件＝公平性
- (2) A(Attractiveness)要件＝誘引性
- (3) D(Duality of purpose)要件＝目的の二重性

本稿では、第2章第2節でこのFAD要件を本研究で取り扱った学習アプリに当てはめて、仕掛けとしての適性について検討した。

## 第2章 仕掛けの概要

### 第1節 多分野対応用語学習アプリケーション

本研究では特定の分野に限定することなく、学習者が任意の用語とその定義を入力することで多様な分野に及ぶ用語学習教材を提供できる多分野対応用語学習アプリケーションを開発した。また、本研究では近年の生成AIの飛躍的な進化を受けて、学習アプリ開発におけるプログラムコード生成という技術的支援を目的にコード生成AIを使用した開発を試みた。学習アプリには以下の機能を備えた。

- (1) 生成AIのAPIを活用した自動用語定義生成機能（図2）
- (2) 分野や学習目的に応じた問題集のカスタマイズ機能（図3）
- (3) 生成AIのAPIを活用した自動四択選択肢生成機能（図4）
- (4) 誤答問題の再挑戦機能（図5）



図2. 自動用語定義生成機能



図3. 問題集のカスタマイズ機能



図 4. 自動四択選択肢生成機能



図 5. 誤答問題の再挑戦機能

## 第 2 節 仕掛けとしての適性

まず、「公平性」(F 要件) は仕掛けによって誰も不利益を被らないことであり、人を欺くものは「仕掛け」の定義から外れる。本研究では、学習者に適切な学習機会を提供したので、学習アプリは「公平性」(F 要件) を満たす。

次に、「誘引性」(A 要件) は行動を「誘う」仕掛けの性質のことであり、行動変容を「強要」するものは仕掛けの定義から外れる。この要件を満たす前提として、仕掛けが行動の選択肢を増やしていること、および私たちが自分の意志で自由に行動を選択することが必要である。本研究では学習者に学習アプリを提供することで、用語学習の新たな行動の選択肢を提示したが、必ず学習アプリを使うように行動変容を強要しなかった。したがって学習アプリは「誘引性」(A 要件) を満たす。

最後に、「目的の二重性」(D 要件) は、仕掛ける側の目的(解決したい問題)と仕掛けられる側の目的(行動したくなる理由)が異なることであり、この二重性のないものは「仕掛け」の定義から外れる。本研究では筆者が仕掛ける側であり、筆者にとっての目的は「学習者が用語学習する分野のニーズに応えられる」ことである。対して学習者は仕掛けられる側であり、学習者にとっての目的は「四択問題の演習形式で用語をゲーム感覚で学ぶ」ことである。したがって学習アプリは「目的の二重性」(D 要件) を満たす。

これらのことから、学習アプリは「仕掛け」の三要件を満たすものである。

## 第3章 実証実験

### 第1節 調査の概要と目的

本調査<sup>i</sup>は

- (1) 学習アプリが用語学習に与える効果
- (2) 多分野対応という学習アプリの設計がユーザーのニーズに応えられているかについて調べるために行った。そのため「音楽用語」という分野に注目した。音楽との関わりの違いによって音楽用語の習熟度に応じたような差が生まれるかを調べるために実験対象として吹奏楽部と4年2組を選んだ。そして、両方の実験前後で音楽用語習熟度テストを実施し、その結果をもとに分析した。

### 第2節 調査方法

先行研究より、介入経過日数が増えるに伴い仕掛けの効果は低下していくことが示されていた[4]ので、実証実験の期間は2025年1月14日～1月16日の3日間とした。日常の学校生活で音楽と関わりのある集団における学習アプリの効果を検証するために、調査対象として神戸大学附属中等教育学校吹奏楽部の生徒を選んだ。吹奏楽部における調査では生徒を以下の2つの群に分けた。

- (1) 実験群：木管・打楽器を担当する生徒
  - (2) 統制群：金管楽器を担当する生徒
- また、日常の学校生活で音楽と関わりを持たない集団における学習アプリの効果を検証するために、比較対象として同校4年2組の生徒を選んだ。4年2組における調査では生徒を以下の2つの群に分けた。

<sup>i</sup> 本研究は神戸大学附属中等教育学校の人を対象とする調査に係る研究倫理審査を通過している。(承認番号：24-110)

(1) 実験群：教室の右側3列に着席している生徒

(2) 統制群：教室の左側3列に着席している生徒

実験前後に1問1点の10点満点の音楽用語習熟度テストに回答してもらい、その結果をもとに統計的に分析をした。

### 第3節 調査結果

音楽用語習熟度テストの有効回答は吹奏楽部で58人中9人に、4年2組で37人中13人から得られた。結果は表1のとおりである。(調査前、調査後の列の数値は得点)

表1. 音楽用語習熟度テストの結果

所属	群	調査前	調査後
吹奏楽部	実験群	5	4
吹奏楽部	実験群	2	5
吹奏楽部	実験群	4	7
吹奏楽部	実験群	8	9
吹奏楽部	統制群	3	3
吹奏楽部	統制群	9	8
吹奏楽部	統制群	8	7
吹奏楽部	統制群	5	2
吹奏楽部	統制群	4	1
4年2組	実験群	4	7
4年2組	実験群	5	6
4年2組	実験群	4	7
4年2組	実験群	2	4
4年2組	実験群	1	3
4年2組	実験群	2	5
4年2組	実験群	3	6
4年2組	統制群	3	3
4年2組	統制群	7	4
4年2組	統制群	10	10
4年2組	統制群	1	2
4年2組	統制群	1	3
4年2組	統制群	4	2

## 第4節 考察

### 第1項 学習アプリが用語学習に与える効果について

本考察では、学習アプリの使用・不使用とテストの得点の改善あり・なしの独立性を検証することを目的にカイ二乗検定を行った。テストの得点の改善とは調査後のテストの得点が調査前のテストの得点より高くなったことと定義する。まず、吹奏楽部と4年2組の調査結果を用いてカイ二乗検定<sup>i</sup>を行った。その際の仮説は  
 帰無仮説  $H_0$ : 学習アプリの使用・不使用とテストの得点の改善あり・なしとは互いに独立である  
 対立仮説  $H_1$ : 学習アプリの使用・不使用とテストの得点の改善あり・なしとは独立でない（関係がある）  
 である。ただし有意水準5%で検定を行った。調査結果を2×2分割表にまとめたものは表2である。  
 検定統計量  $T$  は 7.79、棄却域は 3.84 と算出されたため、 $T=7.79 > 3.84$  より  $H_0$  を棄却する。よって学習アプリの使用・非使用とテストの得点の改善あり・なしとは独立でない（関係がある）。したがって、学習アプリには一定の効果が確認された。

表2. 全体におけるテストの結果の2×2分割表

		学習アプリを		計
		使用	非使用	
改善	あり	8	4	12
	なし	0	10	10
計		8	14	22

<sup>i</sup> 2×2分割表中に4以下の値が含まれていたため、正確な検定を行う目的で Yates の補正式を適用した。

そして実質的な効果の大きさを測るために、カイ二乗検定の効果量  $\phi$  の値を算出した[5]ところ、0.60 という値を得た。すなわち学習アプリの効果は大きいことが示された。[6]

### 第2項 吹奏楽部と4年2組の平均得点の差について

吹奏楽部と4年2組のテストの平均得点の差を検証することを目的に2標本  $t$  検定を行った。最初に、等分散であるかどうかの  $F$  検定を行った。その際の仮説は  
 帰無仮説  $H_0$ :  $\sigma_1^2 = \sigma_2^2$   
 対立仮説  $H_1$ :  $\sigma_1^2 \neq \sigma_2^2$   
 である。ただし有意水準5%で検定を行った。吹奏楽部、4年2組における調査結果を表にまとめたものはそれぞれ表3、表4である。（一番左の列はナンバー、調査前、調査後の列の数値は得点）  
 検定統計量  $T$  は 1.25、棄却域は 3.512 と算出されたため、 $T=1.25 < 3.512$  より  $H_0$  を棄却しない。よって2つの母分散は異なるとはいえない。つまり、等分散であると判断できた。

次に、等分散であることがわかったので、 $t$  検定を行う。その際の仮説は  
 帰無仮説  $H_0$ :  $\mu_1 = \mu_2$   
 対立仮説  $H_1$ :  $\mu_1 \neq \mu_2$   
 である。ただし有意水準5%で両側検定を行った。

検定統計量  $T$  は -1.61、棄却域は 2.086 と算出されたため、 $T=-1.61 < 2.086$  より  $H_0$  を棄却しない。よって吹奏楽部と4年2組の平均得点は異なるとはいえない。

そして実質的な効果の大きさを測るために、 $t$  検定の効果量 Cohen's  $d$  の値を算出した[5]ところ、0.70 という値を得た。すなわち中程度の効果があることが示された。[6]

表 3. 吹奏楽部におけるテストの結果

吹奏楽部	調査前	調査後	差分
1	5	4	-1
2	2	5	3
3	4	7	3
4	8	9	1
5	3	3	0
6	9	8	-1
7	8	7	-1
8	5	2	-3
9	4	1	-3

表 4. 4年2組におけるテストの結果

4年2組	調査前	調査後	差分
1	4	7	3
2	5	6	1
3	4	7	3
4	2	4	2
5	1	3	2
6	2	5	3
7	3	6	3
8	3	3	0
9	7	4	-3
10	10	10	0
11	1	2	1
12	1	3	2
13	4	2	-2

が認められた。そして $\phi$ は0.60であり、大きな効果があることが示された。また、吹奏楽部と4年2組の平均得点に有意な差があるとは判断されなかったが、Cohen's dは0.70であり、中程度の効果があることが示された。

## 第2節 本研究の課題点と今後の展望

本研究では、学習アプリの開発過程において適切な開発環境の選定に時間を要した。また、実証実験において被調査者に与える負担が大きく、十分な量のデータを収集できなかった。標本サイズが小さいため、各種仮説検定の検出力が低下した可能性がある。吹奏楽部における調査では木管・打楽器を担当する生徒を実験群、金管楽器を担当する生徒を統制群に分けたが、木管・打楽器を担当する生徒と金管楽器を担当する生徒の間で音楽用語の知識に差がある可能性について考慮していなかった。実験群の生徒に学習アプリを使用して良かった点について感想を集めたところ、「クイズ形式で覚えられる点」、「選択肢の難易度が適切で、ほどよく難しかった点」、「間違えた問題を繰り返して学習できる点」という回答が得られた。このように、学習アプリの効果に寄与した機能の特定を進め、より学習効果の高い学習アプリの開発に取り組みたい。

## 第4章 結論

### 第1節 まとめ

本研究では神戸大学附属中等教育学校吹奏楽部と4年2組を対象に実証実験を実施した。実証実験の結果を分析したところ、学習アプリの使用・非使用とテストの得点の改善あり・なしとは独立でない（関係がある）。つまり、学習アプリには一定の効果

## 謝辞

本研究を進めるにあたり、神戸大学附属中等教育学校の林兵馬先生にご指導いただきました。また学習アプリの開発段階で島根県立出雲商業高等学校の吉田利幸先生にご指導いただきました。最後に、神戸大学附属中等教育学校吹奏楽部、4年2組のみなさまに本研究の実証実験にご協力いただきました。ここに記して感謝申し上げます。



## 付録 A 調査用紙と音楽用語習熟度テスト

第4章で述べた実証実験で用いた調査用紙を図6に、調査前後で行った音楽用語習熟度テストを図7、8に示す。

図6. 調査用紙

令和7年1月14日

**用語学習アプリを用いた調査へのご協力のお誘い**

13回生の大竹優輝です。私は4KPで用語学習アプリの効果について研究を行っております。  
つきましては、以下の点にご留意のうえ、ご協力をお願い申し上げます。

- 調査目的  
本調査は、用語学習アプリの効果を確認するため、コントロール群との比較を行うものです。
- 利用可能期間  
アプリは、2025年1月14日(水)から1月16日(木)の期間にご利用ください。
- 共有に関して  
調査の正確性を保つため、アプリの取組を他の人に共有しないようお願いいたします。

用語学習アプリ (BooBa) のURL: <https://booba.created.app/>



図7. 調査前テストの内容

### 音楽用語定着度テスト(前)

13回生の大竹優輝です。私は4KPで用語学習アプリの効果について研究を行っております。  
この研究では実験前に音楽用語定着度テスト(前)に、実験後に音楽用語定着度テスト(後)に回答していただきその結果をもとに学習アプリの効果を分析します。  
所要時間は約3分で、回答締め切りは2025年1月14日です。実験・アンケートのご協力をお願いします。

13023\_ohtake\_yuki@schoools.kobe-u.ac.jp  
[アカウントを切り替える](#)

共有なし

\* 必須の質問です

出席番号 (4桁) \*

回答を入力

所属 \*

吹奏楽部  
 4年2組

**Allegro \*** 1ポイント

穏やかに  
 速く  
 落ち遅いて  
 力強く

**Lento \*** 1ポイント

おそく  
 速く  
 軽やかに  
 力強く

**più mosso \*** 1ポイント

穏やかに  
 突然止まる  
 今までより速く  
 今までより遅く

**rallentando \*** 1ポイント

次第に静かに  
 次第に強く  
 次第に遅く  
 次第に速く

**subito \*** 1ポイント

ゆっくりと  
 強く  
 穏やかに  
 すぐに

**maestoso \*** 1ポイント

繊細に  
 荘厳に  
 滑らかに  
 情熱的に

**agitato \*** 1ポイント

優しく  
 憂うつに  
 愉快に  
 激しく

**con fuoco \*** 1ポイント

巧妙に  
 情熱的に  
 元気に  
 荒々しく

**cantabile \*** 1ポイント

愛らしく  
 悲しげに  
 優雅に  
 歌うように

**sempre \*** 1ポイント

常に  
 短く  
 速く  
 単調に

図 8. 調査後テストの内容

**音楽用語定着度テスト(後)**

13回生の太竹優輝です。私は4 KPで用語学習アプリの効果について研究を行っております。この研究では実験前に音楽用語定着度テスト(前)に、実験後に音楽用語定着度テスト(後)に回答していただきその結果をもとに学習アプリの効果进行分析します。所要時間は約3分で、回答締め切りは2025年1月16日です。実験・アンケートのご協力をお願いします。

13023\_ohatake\_yuki@schools.kobe-u.ac.jp  
[アカウントを切り替える](#)  
 共有なし

**\* 必須の質問です**

**出席番号 (4桁) \***

回答を入力

**所属 \***

吹奏楽部 (木管・打楽器パートの方)  
 吹奏楽部 (金管楽器パートの方)  
 4年2組 (運動場側3列の座席の方)  
 4年2組 (廊下側3列の座席の方)

**Adagio \*** 1ポイント

遅く  
 ゆるやかに  
 軽やかに  
 興奮して

**Presto \*** 1ポイント

きわめて速く  
 急に  
 穏やかに  
 静かに

**meno mosso \*** 1ポイント

穏やかに  
 突然止まる  
 今までより速く  
 今までより遅く

**stringendo \*** 1ポイント

次第に静かに  
 次第に強く  
 次第に遅く  
 次第に速く

**a tempo \*** 1ポイント

今までより速く  
 今までより遅く  
 もとの拍子で  
 もとの速さで

**con tenerezza \*** 1ポイント

繊細に  
 愛情をこめて  
 滑らかに  
 荘厳に

**grandioso \*** 1ポイント

感情的に  
 甘美に  
 愉快に  
 壮大に

**espressivo \*** 1ポイント

巧妙に  
 表情豊かに  
 重々しく  
 華やかに

**dolce \*** 1ポイント

やさしく  
 哀げに  
 優雅に  
 歌うように

**simile \*** 1ポイント

同様に  
 笑顔で  
 十分に  
 単調に

**用語学習アプリに関するアンケート**

用語学習アプリ(<https://oboeru.created.app/>)を使用しましたか \*

使用した  
 使用しなかった

このアプリで特に良かった点を教えてください。

回答を入力

改善してほしい点や機能があれは教えてください。

回答を入力

その他ご意見や感想があればお聞かせください。

回答を入力

## 付録 B コード生成 AI を用いた開発環境

本研究では、コード生成 AI を使用した学習アプリの開発を試みた。今後のアプリケーション開発を伴う研究の一助とするため、ここに各種コード生成 AI の特徴を示す。

### (1) Thinkable を使った開発

Thinkable<sup>i</sup>はネイティブアプリ（モバイルアプリ）の開発ができるノーコードツールであり、2025年1月27日現在、無料で使用することができる。Thinkable を活用すると、ネイティブアプリ（モバイルアプリ）の開発から配信準備までが行える。Thinkable の特徴として、コードレスでアプリを開発することが可能であり、フロントエンドに関しては、画面上のコンポーネントをはじめとした UI を直観的にデザインすることができる。また、バックエンドに関しては、画面上のブロックをつなぎ合わせてプログラムを作る仕様になっており、ノーコードで視覚的にアプリの開発が行える。API 連携が可能のため、サービスを拡張できる。

筆者は MIT が公開している Scratch でブロックを使ったプログラミングを経験していたので、研究当初はこの Thinkable を使ったアプリの開発を試みていた。

しかしながら、ブロックを使ったプログラミング、いわゆるビジュアルプログラミング言語はプログラムが複雑になるにつれて画面上が煩雑化し、プログラムの把握が困難になった。また、視認性が重視されているため、プログラムの処理速度が遅かった。

### (2) bolt を使った開発

bolt<sup>ii</sup>はブラウザベースの開発環境を提供

している StackBlitz 社が開発した、フロントエンドとバックエンド双方をブラウザ上で同時に開発するフルスタック開発ができるノーコードツールであり、2025年1月20日現在、無料で使用することができる。bolt を活用すると、ウェブサイトやアプリケーションの開発からデプロイまで行える。bolt の特徴として、日本語や英語などの自然言語で開発の指示をすることで、AI がコードを生成する。また、生成したアプリをリアルタイムで確認・編集が可能のため、修正指示が出しやすい。

しかしながら、自動四択選択枝生成のための生成 AI の API を導入できなかったことや、プログラムが複雑になるにつれてコード生成の段階でエラーが生じやすくなった。

### (3) create を使った開発

create<sup>iii</sup>も bolt と同様にフルスタック開発ができるノーコードツールであり、2025年1月20日現在、無料でアプリ開発を行うことができる。create はコード生成 AI とのチャットを通じてアプリの開発を進めることができる。また Thinkable や bolt の課題点だった各コンポーネントへの修正指示が個別に出しやすく、細かい部分まで精密に制御することができる。また、API 連携が行えることから、サービスの拡張が可能である。特に create ではコードを生成した段階で毎回プログラムをアーカイブに保存しているため、コード生成に失敗したときなどに過去のバージョンに戻って開発をやり直せる点が優れていた。

---

<sup>i</sup> <https://thinkable.com/>

<sup>ii</sup> <https://bolt.new/>

---

<sup>iii</sup> <https://www.create.xyz/>

## 付録 C 学習アプリのプログラム

### (1) 基本構造

```
"use client";
import React from "react";

import { useHandleStreamResponse } from "../utilities/runtime-helpers";

function MainComponent() {
  const [selectedWordList, setSelectedWordList] = React.useState("");
  const [words, setWords] = React.useState([]);
  const [newWord, setNewWord] = React.useState({
    term: "",
    definition: "",
    listName: "",
  });
  const [showAddForm, setShowAddForm] = React.useState(false);
  const [activeTab, setActiveTab] = React.useState("wordlist");
  const [isTranslating, setIsTranslating] = React.useState(false);
  const [showNewFolderModal, setShowNewFolderModal] = React.useState(false);
  const [newFolderName, setNewFolderName] = React.useState("");
  const [wordLists, setWordLists] = React.useState([]);
  const [currentQuestion, setCurrentQuestion] = React.useState(0);
  const [score, setScore] = React.useState(0);
  const [showScore, setShowScore] = React.useState(false);
  const [options, setOptions] = React.useState([]);
  const [selectedAnswer, setSelectedAnswer] = React.useState(null);
  const [streamingDefinition, setStreamingDefinition] = React.useState("");
  const [isGeneratingDefinition, setIsGeneratingDefinition] =
    React.useState(false);
  const [formErrors, setFormErrors] = React.useState({});
  const [showSuccess, setShowSuccess] = React.useState(false);
  const [incorrectQuestions, setIncorrectQuestions] = React.useState([]);
  const [loadingOptions, setLoadingOptions] = React.useState(false);
  const [totalQuestions, setTotalQuestions] = React.useState(0);
  const handleStreamResponse = useHandleStreamResponse({
    onChunk: setStreamingDefinition,
    onFinish: (message) => {
      setNewWord((prev) => ({ ...prev, definition: message }));
      setIsGeneratingDefinition(false);
    }
  });
}
```

```

    setStreamingDefinition("");
  },
});
const generateDefinition = async (term) => {
  setIsGeneratingDefinition(true);
  try {
    const response = await fetch("/integrations/chat-gpt/conversationgpt4", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({
        messages: [
          {
            role: "system",
            content: "あなたは用語の意味を簡潔に言い換える専門家です。",
          },
          {
            role: "user",
            content: `「${term}」の意味を短い言葉で言い換えてください。`,
          },
        ],
        stream: true,
      }),
    });
    handleStreamResponse(response);
  } catch (error) {
    console.error(error);
    setIsGeneratingDefinition(false);
  }
};
const generateWrongAnswers = async (term, correctDefinition) => {
  try {
    const response = await fetch("/integrations/chat-gpt/conversationgpt4", {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({
        messages: [
          {
            role: "system",
            content: "あなたは教育コンテンツ作成の専門家です。",
          },

```

```

    {
      role: "user",
      content: `「${term}」の誤った定義を3つ生成してください。正しい定義は「${correctDefinition}」です。誤った定義は、一見正しそうに見えるが実際は間違っているものにしてください。各定義は20文字以内でお願いします。`,
    },
  ],
  json_schema: {
    name: "wrong_answers",
    schema: {
      type: "object",
      properties: {
        answers: {
          type: "array",
          items: {
            type: "string",
          },
        },
      },
      required: ["answers"],
      additionalProperties: false,
    },
  },
}),
});
const data = await response.json();
return JSON.parse(data.choices[0].message.content).answers;
} catch (error) {
  console.error(error);
  return [];
}
};

const handleAddWord = async () => {
  const errors = {};
  if (!newWord.term) errors.term = "用語を入力してください";
  if (!newWord.definition) errors.definition = "定義を入力してください";
  if (!newWord.listName) errors.listName = "教材を選択してください";

  setFormErrors(errors);

```

```

if (Object.keys(errors).length === 0) {
  const updatedLists = wordLists.map((list) => {
    if (list.name === newWord.listName) {
      return {
        ...list,
        words: [
          ...list.words,
          { term: newWord.term, definition: newWord.definition },
        ],
      };
    }
    return list;
  });

  try {
    await fetch("/api/db/039", {
      method: "POST",
      body: JSON.stringify({
        query:
          "INSERT INTO `terms` (term, definition, list_name) VALUES (?, ?, ?)",
        values: [newWord.term, newWord.definition, newWord.listName],
      }),
    });

    setWordLists(updatedLists);
    setNewWord({ term: "", definition: "", listName: "" });
    setShowAddForm(false);
    setShowSuccess(true);
    setTimeout(() => setShowSuccess(false), 3000);
    localStorage.setItem("wordLists", JSON.stringify(updatedLists));
  } catch (error) {
    console.error(error);
  }
}

const handleDeleteList = async (listName, e) => {
  e.preventDefault();
  try {
    await fetch("/api/db/039", {
      method: "POST",

```

```

    body: JSON.stringify({
      query: "DELETE FROM `terms` WHERE list_name = ?",
      values: [listName],
    }),
  });
  const updatedLists = wordLists.filter((list) => list.name !== listName);
  setWordLists(updatedLists);
  localStorage.setItem("wordLists", JSON.stringify(updatedLists));
} catch (error) {
  console.error(error);
}
};

const generateOptions = async (questionIndex) => {
  setLoadingOptions(true);
  const correctAnswer = words[questionIndex]?.definition || "";
  const wrongAnswers = await generateWrongAnswers(
    words[questionIndex]?.term || "",
    correctAnswer
  );
  const allOptions = [...wrongAnswers, correctAnswer].sort(
    () => Math.random() - 0.5
  );
  setOptions(allOptions);
  setLoadingOptions(false);
};

const fetchWordLists = async () => {
  try {
    const response = await fetch("/api/db/039", {
      method: "POST",
      body: JSON.stringify({
        query: "SELECT DISTINCT list_name FROM `terms`",
      }),
    });
  };
  const data = await response.json();
  const lists = data.map((item) => ({
    name: item.list_name,
    words: [],
  }));
  setWordLists(lists);
  localStorage.setItem("wordLists", JSON.stringify(lists));
};

```



```

    } catch (error) {
      console.error(error);
    }
  };
const handleCreateNewFolder = () => {
  if (newFolderName.trim()) {
    const newList = { name: newFolderName, words: [] };
    const updatedLists = [...wordLists, newList];
    setWordLists(updatedLists);
    setNewFolderName("");
    setShowNewFolderModal(false);
    localStorage.setItem("wordLists", JSON.stringify(updatedLists));
  }
};
const startPractice = async () => {
  setCurrentQuestion(0);
  setScore(0);
  setShowScore(false);
  setIncorrectQuestions([]);
  setTotalQuestions(words.length);
  await generateOptions(0);
  setActiveTab("quiz");
};
const retryIncorrectQuestions = async () => {
  if (incorrectQuestions.length > 0) {
    setWords(incorrectQuestions);
    setCurrentQuestion(0);
    setSelectedAnswer(null);
    setIncorrectQuestions([]);
    setTotalQuestions(incorrectQuestions.length);
    await generateOptions(0);
    setActiveTab("quiz");
  }
};
const handleDeleteWord = async (term, listName) => {
  if (window.confirm("本当に削除してもよろしいですか?")) {
    try {
      await fetch("/api/db/039", {
        method: "POST",
        body: JSON.stringify({

```

```

        query: "DELETE FROM `terms` WHERE term = ? AND list_name = ?",
        values: [term, listName],
    }),
});

const updatedWords = words.filter((word) => word.term !== term);
setWords(updatedWords);
const updatedLists = wordLists.map((list) => {
    if (list.name === listName) {
        return {
            ...list,
            words: list.words.filter((w) => w.term !== term),
        };
    }
    return list;
});
setWordLists(updatedLists);
localStorage.setItem("wordLists", JSON.stringify(updatedLists));
} catch (error) {
    console.error(error);
}
}
};

```

```

React.useEffect(() => {
    const initializeWordList = async () => {
        const initialList = "楽語";
        const defaultWords = [
            { term: "Andante", definition: "歩くような速さで" },
            { term: "più mosso", definition: "今までより速く" },
            { term: "animato", definition: "活気をもって" },
            { term: "cantabile", definition: "歌うように" },
            { term: "espressivo", definition: "表情豊かに" },
            { term: "grandioso", definition: "壮大に" },
            { term: "grazioso", definition: "優美に" },
            { term: "tutti", definition: "総奏で" },
            { term: "G.P.", definition: "全声部の休止" },
            { term: "a2", definition: "2重奏で" },
        ];
    };

```

```

try {
  const response = await fetch("/api/db/039", {
    method: "POST",
    body: JSON.stringify({
      query: "SELECT * FROM `terms` WHERE list_name = ?",
      values: [initialList],
    }),
  });
  const data = await response.json();

  if (data.length === 0) {
    for (const word of defaultWords) {
      await fetch("/api/db/039", {
        method: "POST",
        body: JSON.stringify({
          query:
            "INSERT INTO `terms` (term, definition, list_name) VALUES (?, ?, ?)",
          values: [word.term, word.definition, initialList],
        }),
      });
    }
    fetchWordLists();
  } catch (error) {
    console.error(error);
  }
};

const storedLists = localStorage.getItem("wordLists");
if (storedLists) {
  setWordLists(JSON.parse(storedLists));
} else {
  initializeWordList();
}
}, []);

React.useEffect(() => {
  const defaultList = {
    name: "楽語",
    words: [
      { term: "Andante", definition: "歩くような速さで" },

```

```

    { term: "più mosso", definition: "今までより速く" },
    { term: "animato", definition: "活気をもって" },
    { term: "cantabile", definition: "歌うように" },
    { term: "espressivo", definition: "表情豊かに" },
    { term: "grandioso", definition: "壮大に" },
    { term: "grazioso", definition: "優美に" },
    { term: "tutti", definition: "総奏で" },
    { term: "G.P.", definition: "全声部の休止" },
    { term: "a2", definition: "2重奏で" },
  ],
};
setWordLists((prevLists) => {
  if (!prevLists.some((list) => list.name === "楽語")) {
    return [...prevLists, defaultList];
  }
  return prevLists;
});
}, []);

const renderContent = () => {
  switch (activeTab) {
    case "wordlist":
      return (
        <div className="space-y-2 relative">
          <div className="flex items-center justify-end mb-4">
            <div className="flex flex-col items-center">
              <button
                onClick={() => setShowNewFolderModal(true)}
                className="bg-[#FFB347] w-12 h-12 rounded-full hover:bg-[#FF9F1C]
transition-colors duration-200 shadow-md flex items-center justify-center mb-1"
              >
                <i className="fas fa-folder-plus text-xl"></i>
              </button>
              <span className="text-sm">新規作成</span>
            </div>
          </div>
          {wordLists.map((list, index) => (
            <div
              key={index}

```

```

        className={`bg-[#F5F5F5] text-black p-6 rounded-lg shadow-sm flex
items-center justify-between mb-4 cursor-pointer ${
        selectedWordList === list.name ? "ring-2 ring-[#FFB347]" : ""
    }}
    onClick={() => {
        setSelectedWordList(list.name);
        setWords(list.words);
        setActiveTab("practice");
    }}
>
<div className="flex items-center flex-grow">
    <div className="w-12 h-12 mr-4 flex-shrink-0">
        <i className="fas fa-folder text-4xl text-[#8B4513]"></i>
    </div>
    <div className="flex flex-col">
        <span className="font-crimson-text text-xl">
            {list.name}
        </span>
    </div>
</div>
<button
    onClick={(e) => {
        e.stopPropagation();
        if (window.confirm("本当に削除してもよろしいですか？")) {
            handleDeleteList(list.name, e);
        }
    }}
    className="text-red-500 hover:text-red-700 ml-4"
>
    <i className="fas fa-trash"></i>
</button>
</div>
)}}
{showNewFolderModal && (
    <div className="fixed inset-0 bg-black bg-opacity-50 flex items-center
justify-center z-50">
        <div className="bg-white rounded-xl p-6 w-[90%] max-w-md">
            <h3 className="text-xl mb-4 font-bold">新規フォルダの作成</h3>
            <input
                type="text"

```

```

        value={newFolderName}
        onChange={(e) => setNewFolderName(e.target.value)}
        placeholder="フォルダ名を入力"
        className="w-full p-2 border-2 rounded-lg mb-4 focus:outline-none
focus:border-[#FFB347]"
      />
      <div className="flex justify-end space-x-2">
        <button
          onClick={() => setShowNewFolderModal(false)}
          className="px-4 py-2 rounded-lg bg-gray-200 hover:bg-gray-300"
        >
          キャンセル
        </button>
        <button
          onClick={handleCreateNewFolder}
          className="px-4 py-2 rounded-lg bg-[#FFB347] hover:bg-[#FF9F1C]
text-white"
        >
          作成
        </button>
      </div>
    </div>
  </div>
)}
</div>
);
case "add":
  return (
    <div className="bg-white rounded-xl shadow-lg p-4 max-w-md mx-auto
overflow-y-auto max-h-[calc(100vh-200px)]">
      <h3 className="text-2xl mb-6 font-crimson-text text-center">
        用語登録
      </h3>
      {showSuccess && (
        <div className="bg-green-100 border border-green-400 text-green-700 px-4
py-3 rounded relative mb-4">
          <span className="block sm:inline">用語を追加しました</span>
        </div>
      )}
    </div className="mb-4">

```

```

<select
  value={newWord.listName}
  onChange={(e) =>
    setNewWord({ ...newWord, listName: e.target.value })
  }
  className={`w-full p-2 border-2 rounded-lg focus:outline-none
focus:border-[#FFB347]`} ${{
    formErrors.listName ? "border-red-500" : ""
  }}
>
  <option value="">教材を選択</option>
  {wordLists.map((list) => (
    <option key={list.name} value={list.name}>
      {list.name}
    </option>
  ))}
</select>
{formErrors.listName && (
  <p className="text-red-500 text-sm mt-1">
    {formErrors.listName}
  </p>
)}
</div>
<div className="mb-4">
  <input
    type="text"
    name="term"
    placeholder="用語を入力"
    value={newWord.term}
    onChange={(e) =>
      setNewWord({ ...newWord, term: e.target.value })
    }
    onBlur={(e) => {
      if (e.target.value) {
        generateDefinition(e.target.value);
      }
    }}
    className={`w-full p-2 border-2 rounded-lg focus:outline-none
focus:border-[#FFB347]`} ${{
      formErrors.term ? "border-red-500" : ""
    }}
  >

```

```

    }}
  />
  {formErrors.term && (
    <p className="text-red-500 text-sm mt-1">{formErrors.term}</p>
  )}
</div>
<div className="relative mb-4">
  <textarea
    name="definition"
    placeholder="定義"
    value={
      isGeneratingDefinition
        ? streamingDefinition
        : newWord.definition
    }
    onChange={(e) =>
      setNewWord({ ...newWord, definition: e.target.value })
    }
    className={`w-full p-3 border-2 rounded-lg focus:outline-none
focus:border-[#FFB347] min-h-[100px] ${
      formErrors.definition ? "border-red-500" : ""
    }}
    disabled={isGeneratingDefinition}
  />
  {formErrors.definition && (
    <p className="text-red-500 text-sm mt-1">
      {formErrors.definition}
    </p>
  )}
  {isGeneratingDefinition && (
    <div className="absolute right-3 top-3">
      <i className="fas fa-spinner fa-spin text-[#FFB347]"></i>
    </div>
  )}
</div>
<button
  onClick={handleAddWord}
  disabled={isGeneratingDefinition}
  className="w-full bg-[#FFB347] py-3 rounded-lg hover:bg-[#FF9F1C]
transition-colors duration-200 font-bold shadow-md disabled:opacity-50"

```



```

        >
        追加
      </button>
    </div>
  );
  case "practice":
    return (
      <div className="flex flex-col h-full overflow-hidden">
        {!showScore && words.length > 0 && options.length === 0 ? (
          <div className="flex-grow overflow-y-auto px-4">
            <div className="grid grid-cols-1 md:grid-cols-2 gap-4 w-full max-w-4xl
mx-auto">
              {words.map((word, index) => (
                <div
                  key={index}
                  className="bg-gray-50 p-4 rounded-lg shadow-sm flex
justify-between items-center"
                >
                  <div>
                    <div className="font-bold mb-2">{word.term}</div>
                    <div className="text-gray-600">{word.definition}</div>
                  </div>
                  <button
                    onClick={() =>
                      handleDeleteWord(word.term, selectedWordList)
                    }
                    className="text-red-500 hover:text-red-700 ml-4"
                  >
                    <i className="fas fa-trash"></i>
                  </button>
                </div>
              ))}
            </div>
            <div className="flex justify-center mt-4">
              <button
                onClick={startPractice}
                className="bg-[#FFB347] px-8 py-3 rounded-lg hover:bg-[#FF9F1C]
transition-colors duration-200 font-bold shadow-md"
              >
                学習を始める
            </div>
          </div>
        ) : (
          <div className="flex justify-center align-items-center h-100">
            <div className="text-center">
              <h1>おめでとう</h1>
              <p>あなたは単語を覚えました！</p>
            </div>
          </div>
        )}
      </div>
    );
  }
}

```

```

        </button>
      </div>
    </div>
  ): showScore ? (
    <div className="text-center py-12 overflow-y-auto">
      <h3 className="text-3xl mb-6">結果</h3>
      <p className="text-4xl font-bold mb-6">{score}問正解</p>
      {score === totalQuestions ? (
        <div className="space-y-4">
          <div className="flex flex-col items-center">
            <i className="fas fa-trophy text-6xl text-yellow-500 mb-4"></i>
            <p className="text-xl text-green-600 mb-4">
              おめでとうございます！
            <br />
            全問正解です！
          </p>
        </div>
        <button
          onClick={() => {
            setActiveTab("wordlist");
            setShowScore(false);
            setOptions([]);
            setCurrentQuestion(0);
            setScore(0);
            setSelectedAnswer(null);
          }}
          className="bg-[#FFB347] px-8 py-3 rounded-lg hover:bg-[#FF9F1C]
transition-colors duration-200 font-bold shadow-md"
        >
          教材一覧に戻る
        </button>
      </div>
    ): (
      <div className="space-y-4">
        <div className="flex flex-col items-center">
          <i className="fas fa-redo text-6xl text-blue-500 mb-4"></i>
          <p className="text-xl text-blue-600 mb-4">
            もう一度チャレンジしましょう！
          </p>
        </div>
      </div>
    )
  )

```

```

        <button
          onClick={async () => {
            await retryIncorrectQuestions();
            setScore(0);
          }}
          className="bg-[#FFB347] px-8 py-3 rounded-lg hover:bg-[#FF9F1C]
transition-colors duration-200 font-bold shadow-md"
        >
          間違えた問題に再挑戦
        </button>
      </div>
    )}
  </div>
): words.length === 0 ? (
  <div className="flex-grow flex flex-col justify-center items-center">
    <p className="text-xl text-gray-600 mb-4">
      この教材にはまだ用語が登録されていません。
    </p>
    <button
      onClick={() => setActiveTab("add")}
      className="bg-[#FFB347] px-8 py-3 rounded-lg hover:bg-[#FF9F1C]
transition-colors duration-200 font-bold shadow-md"
    >
      用語を登録する
    </button>
  </div>
): (
  <div className="flex-grow flex flex-col overflow-hidden">
    <div className="bg-[#FFB347] p-4 rounded-lg shadow-md mb-4">
      <div className="flex justify-between items-center">
        <span className="text-lg">
          問題 {currentQuestion + 1} / {totalQuestions}
        </span>
        <span className="text-sm">スコア: {score}</span>
      </div>
    </div>
    <div className="flex-grow flex flex-col justify-center px-4 overflow-y-auto
mb-8">
      <div className="text-center mb-4">
        <h3 className="text-2xl font-bold">

```

```

    {words[currentQuestion]?.term || ""}
  </h3>
</div>
{loadingOptions ? (
  <div className="text-center">
    <i className="fas fa-spinner fa-spin text-[#FFB347] text-2xl"></i>
  </div>
): (
  <div className="grid grid-cols-1 gap-2 max-w-md mx-auto w-full">
    {options.map((option, index) => {
      const isCorrect =
        option === words[currentQuestion]?.definition;
      const buttonStyle = selectedAnswer
        ? isCorrect
          ? "bg-green-100 text-green-600"
          : "bg-red-100 text-red-600"
        : "bg-white";
      return (
        <button
          key={index}
          onClick={async () => {
            if (!selectedAnswer) {
              setSelectedAnswer(option);
              if (
                option !== words[currentQuestion]?.definition
              ) {
                setIncorrectQuestions((prev) => {
                  const newQuestion = words[currentQuestion];
                  return prev.some(
                    (q) => q.term === newQuestion.term
                  )
                    ? prev
                    : [...prev, newQuestion];
                });
              } else {
                setScore(score + 1);
              }
            }
            setTimeout(async () => {
              setSelectedAnswer(null);
              const nextQuestion = currentQuestion + 1;

```

```

        if (nextQuestion < words.length) {
            setCurrentQuestion(nextQuestion);
            await generateOptions(nextQuestion);
        } else {
            setShowScore(true);
            setActiveTab("result");
        }
    }, 1000);
}
}}
disabled={selectedAnswer !== null}
className={`p-3 text-left border-2 rounded-xl transition-all
duration-200 shadow-sm ${buttonStyle} ${
    !selectedAnswer &&
    "hover:bg-gray-50 hover:shadow-md"
}`}
>
    {option}
    {selectedAnswer && (
        <span className="float-right">
            {isCorrect ? "○" : "✘"}
        </span>
    )}
</button>
);
}}
</div>
)}
</div>
</div>
)}
</div>
);
case "quiz":
return (
    <div className="flex-grow flex flex-col overflow-hidden">
        <div className="bg-[#FFB347] p-4 rounded-lg shadow-md mb-4">
            <div className="flex justify-between items-center">
                <span className="text-lg">
                    問題 {currentQuestion + 1} / {totalQuestions}
                </span>
            </div>
        </div>
    </div>
);

```

```

    </span>
    <span className="text-sm">スコア: {score}</span>
  </div>
</div>
<div className="flex-grow flex flex-col justify-center px-4 overflow-y-auto
mb-8">
  <div className="text-center mb-4">
    <h3 className="text-2xl font-bold">
      {words[currentQuestion]?.term || ""}
    </h3>
  </div>
  {loadingOptions ? (
    <div className="text-center">
      <i className="fas fa-spinner fa-spin text-[#FFB347] text-2xl"></i>
    </div>
  ) : (
    <div className="grid grid-cols-1 gap-2 max-w-md mx-auto w-full">
      {options.map((option, index) => {
        const isCorrect =
          option === words[currentQuestion]?.definition;
        const buttonStyle = selectedAnswer
          ? isCorrect
            ? "bg-green-100 text-green-600"
            : "bg-red-100 text-red-600"
          : "bg-white";
        return (
          <button
            key={index}
            onClick={async () => {
              if (!selectedAnswer) {
                setSelectedAnswer(option);
                if (option !== words[currentQuestion]?.definition) {
                  setIncorrectQuestions((prev) => {
                    const newQuestion = words[currentQuestion];
                    return prev.some(
                      (q) => q.term === newQuestion.term
                    )
                  ? prev
                  : [...prev, newQuestion];
                }
              }
            }});
      })}
    </div>
  )}

```

```

    } else {
      setScore(score + 1);
    }
    setTimeout(async () => {
      setSelectedAnswer(null);
      const nextQuestion = currentQuestion + 1;
      if (nextQuestion < words.length) {
        setCurrentQuestion(nextQuestion);
        await generateOptions(nextQuestion);
      } else {
        setShowScore(true);
        setActiveTab("result");
      }
    }, 1000);
  }
}}
disabled={selectedAnswer !== null}
className={`p-3 text-left border-2 rounded-xl transition-all
duration-200 shadow-sm ${buttonStyle} ${
  !selectedAnswer && "hover:bg-gray-50 hover:shadow-md"
}`}
>
  {option}
  {selectedAnswer && (
    <span className="float-right">
      {isCorrect ? "○" : "✖"}
    </span>
  )}
</button>
);
}}
</div>
)}
</div>
</div>
);
case "result":
return (
  <div className="text-center py-12 overflow-y-auto">
    <h3 className="text-3xl mb-6">結果</h3>

```

```

<p className="text-4xl font-bold mb-6">{score}問正解</p>
{score} === totalQuestions ? (
  <div className="space-y-4">
    <div className="flex flex-col items-center">
      <i className="fas fa-trophy text-6xl text-yellow-500 mb-4"></i>
      <p className="text-xl text-green-600 mb-4">
        おめでとうございます！
      <br />
      全問正解です！
    </p>
  </div>
  <button
    onClick={() => {
      setActiveTab("wordlist");
      setShowScore(false);
      setOptions([]);
      setCurrentQuestion(0);
      setScore(0);
      setSelectedAnswer(null);
    }}
    className="bg-[#FFB347] px-8 py-3 rounded-lg hover:bg-[#FF9F1C]
transition-colors duration-200 font-bold shadow-md"
  >
    教材一覧に戻る
  </button>
</div>
): (
  <div className="space-y-4">
    <div className="flex flex-col items-center">
      <i className="fas fa-redo text-6xl text-blue-500 mb-4"></i>
      <p className="text-xl text-blue-600 mb-4">
        もう一度チャレンジしましょう！
      </p>
    </div>
    <button
      onClick={async () => {
        await retryIncorrectQuestions();
        setScore(0);
      }}
    >

```



```

        className="bg-[#FFB347] px-8 py-3 rounded-lg hover:bg-[#FF9F1C]
transition-colors duration-200 font-bold shadow-md"
      >
        間違えた問題に再挑戦
      </button>
    </div>
  )}
</div>
);
default:
  return null;
}
};

return (
  <div className="min-h-screen bg-[#FDF6E9] flex flex-col">
    <header className="bg-[#8B4513] text-white p-2 shadow-lg">
      <h1 className="text-center text-2xl font-crimson-text tracking-wider">
        OboeRu
      </h1>
    </header>
    <div className="container mx-auto px-4 py-2 flex-grow overflow-hidden">
      <div className="bg-white rounded-xl shadow-xl p-4 h-[calc(100vh-140px)]">
        {renderContent()}
      </div>
    </div>
    <footer className="bg-[#8B4513] text-white p-2 w-full shadow-lg">
      <div className="grid grid-cols-2 gap-1 max-w-md mx-auto">
        <button
          onClick={() => {
            setActiveTab("wordlist");
            setShowScore(false);
            setOptions([]);
            setCurrentQuestion(0);
            setScore(0);
            setSelectedAnswer(null);
          }}
          className={`py-3 rounded-lg text-center transition-colors duration-200 ${
            activeTab === "wordlist"
              ? "bg-white text-[#8B4513]"

```

```

        : "bg-[#A0522D] text-white hover:bg-[#8B4513]"
      `}`
    >
    <i className="fas fa-book mb-1"></i>
    <div className="text-xs">教材一覧</div>
  </button>
  <button
    onClick={() => setActiveTab("add")}
    className={`py-3 rounded-lg text-center transition-colors duration-200 ${
      activeTab === "add"
        ? "bg-white text-[#8B4513]"
        : "bg-[#A0522D] text-white hover:bg-[#8B4513]"
    }`
  >
    <i className="fas fa-plus mb-1"></i>
    <div className="text-xs">用語登録</div>
  </button>
</div>
</footer>
</div>
);
}

```

```
export default MainComponent;
```

(2) 選択肢の生成

```
async function handler({ term }) {
```

const prompt = `「\${term}」の対義語や類義語に基づいて、間違いやすい3つの誤答選択肢を生成してください。それらは学習者が混乱しやすく、関連性のある単語であるべきです。例：「happy」の場合、「sad」「joyful」「delighted」のように。JSONの配列形式で返してください。`;

```

const response = await fetch("/api/generate-definition", {
  method: "POST",
  headers: {
    "Content-Type": "application/json",
  },
  body: JSON.stringify({ term, prompt }),
});

```

```
const data = await response.json();
```

```

if (!data.definition) {
  return { term, incorrectDefinitions: [] };
}

const definitions = JSON.parse(data.definition);

return {
  term,
  incorrectDefinitions: definitions,
};
}

```

### (3) 定義の生成

```

async function handler({ term }) {
  const prompt = `「${term}」の最も基本的な訳語を1つだけ教えてください。説明的な言葉は避け、辞書の見出し語のような単純な対訳を提供してください。句読点は使用しないでください。

```

例：

book → 本

student → 学生

computer → コンピュータ

「\${term}」 → `;

```

const response = await fetch("/integrations/chat-gpt/conversationgpt4", {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify({
    messages: [
      {
        role: "system",
        content:
          "あなたは翻訳辞書です。入力された単語に対して、最も基本的な訳語を1つだけ返してください。",
      },
      {
        role: "user",
        content: prompt,
      },
    ],
  })

```

```

    ],
  }),
});

if (!response.ok) {
  return { definition: "" };
}

const data = await response.json();
const definition = data.choices[0].message.content
  .trim()
  .replace(/[\^¥p{L}¥p{N}]/gu, "");

return { definition };
}

```

## 参考文献

- [1] Apple (2025). 「App Store のトップ iPhone 教育アプリ - Apple (JP)」 .  
<https://apps.apple.com/jp/charts/iphone/%E6%95%99%E8%82%B2-apps/6017>.  
 2025 年 1 月 27 日.
- [2] 仕掛学研究会 (2025). 「仕掛学研究会のホームページ」 .  
<https://shikakeology.org/>. 2025 年 1 月 27 日.
- [3] 大竹優輝 (2024). 「机を整列させる仕掛けの効果の検証—神戸大学附属中等教育学校 3 年 2 組における実証実験より—」 .  
 第 14 回仕掛学研究会.
- [4] 松村真宏 (2016). 『仕掛学 人を動かすアイデアの作り方』. 東洋経済新報社.
- [5] 水本篤, 竹内理 (2008). 「研究論文における効果量の報告のために—基礎的概念と注意点—」. 『英語教育研究』. 31, 57-66.
- [6] Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum.