5.2　教育コンテンツ——演習課題





















S2-02（演習）基本プロンプトでの結果の揃え方

a　受講者向け課題（提出物仕様）

課題趣旨 ▶

配布データ（この課題文に含まれる）▶ ○【資料G】架空会議メモ（スプリント計画）

成果物（提出物）▶

作業タスク▶

S2-02（演習）基本プロンプトでの結果の揃え方

b　教員向け：実施計画（評価基準含む）

事前準備▶

当日進行（90分想定）▶

評価（100点満点）▶

観点	配点	到達水準（合格）
目的・条件・形式の明示	30	基本プロンプトで意図が伝わっている
現代語情報と変数記載	25	会話例・変数が正確に書けている
安定性（出力）の確認性	25	条件追加・修正指示で対応できる
品質ゲート／停止基準	20	不明点・リスクが適切に扱われている
再利用・共有の工夫	10	他者が使いやすい形で整理されている

S2-02（演習）基本プロンプトでの結果の揃え方

c　教員向け：シナリオ（進行台本）

S2-04（演習）再プロンプトによる改善ループ

a　受講者向け課題（提出物仕様）

課題趣旨 ▶

配布データ▶
（この課題文に含まれる）

○【資料E】一次情報（要件メモ・議事）

○【資料F】生成AIが作った"誤りを含む要約"（改善対象）

S2-04（演習）再プロンプトによる改善ループ

a　受講者向け課題（提出物仕様）

作業タスク ▶

成果物（提出物）▶

S2-04（演習）再プロンプトによる改善ループ

b　教員向け：実施計画（評価基準含む）

事前準備（教員）▶

当日進行（90分想定）▶

評価（100点満点）▶

観点	配点	到達水準（合格）
問題点抽出の網羅性	30	誤り・断定を漏れなく特定
情報源引用（一次情報参照）	25	資料名や該当箇所を明示できる
再プロンプトの妥当性	25	誤りが修正されるよう条件が書けている
資料要約の品質	20	資料に基づき、不要や余計な事実がない

S2-06（演習）パターン組合せによる業務プロンプトの作成

a　受講者向け課題（提出物仕様）

課題趣旨 ▶

○【資料G】顧客からの問合せ（架空）

○【資料H】社内方針（抜粋）

作業タスク ▶

S2-06（演習）パターン組合せによる業務プロンプトの作成

a　受講者向け課題（提出物仕様）

成果物（提出物）▶

回答（基礎版）フォーマット（箇条）▶

S2-06（演習）パターン組合せによる業務プロンプトの作成

b　教員向け：実施計画（評価基準含む）

事前準備 ▶

当日進行（90分想定）▶

評価（100点満点）▶

観点	配点	到達水準（合格）
パターン選択の適切性	30	3点とも3パターンを適切に使えている
事実性・確定指標	25	確定と未決を区別、要確認を明示
再現性（形式指定）	20	指定フォーマットが安定して出る
出力評価と再調整	20	品質・再現性の観点で説明できる
作業記録（履歴・根拠）	10	作業の経緯が残る

S2-06（演習）パターン組合せによる業務プロンプトの作成

c　教員向け：シナリオ（進行台本）

5.3　教育コンテンツ——自習問題

科目	コマ	問題	出題形式	問題文	正答肢	誤答肢	誤答肢	誤答肢
S1	01	01	空欄補充	「生成AIへの指示では、目的と条件に加えて回答の（　）を明示すると結果が揃い易い」という文の空欄に入る最も適切なものはどれか。	出力形式	推論設定	入力制約	監査ログ
S1	01	02	空欄補充	「生成AI利用の安全ルールでは、個人情報や機密等、外部送信できない情報を「（　）」に区分して入力を避ける」という文の空欄に入る最も適切なものはどれか。	入力禁止	出力形式	作業記録	改善ループ
S1	01	03	事象理解	生成AIにおいて、同一の入力に対して毎回同じ出力が返る性質を何と言うか。	決定論的	確率的	反復的	適応的
S1	01	04	事象理解	生成AIに対して目的と条件と出力形式を含む指示文を設計する行為を何と言うか。	プロンプト設計	監査ログ	データ分類	品質ゲート
S1	01	05	正誤判定	「生成AIの出力は確率的であり、同一の入力でも条件や設定によって結果が揺れることがある」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S1	02	01	空欄補充	「生成AIの出力を確かめる場合、複数の資料を（　）して一致点と相違点を確認する」という文の空欄に入る最も適切なものはどれか。	照合	要約	暗号化	量子化
S1	02	02	空欄補充	「生成AIの出力を共有する際、根拠が不明な箇所は言い切りを避けて（　）表現に修正する」という文の空欄に入る最も適切なものはどれか。	断定回避	推論設定	出典表示	監査ログ
S1	02	03	事象理解	生成AIが事実でない内容を事実のように尤もらしく出力する現象を何と言うか。	幻覚	過学習	正規化	蒸留
S1	02	04	正誤判定	「生成AIの出力を検証する際、複数の資料を照合して共通点と矛盾点を確認することは有効である」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S1	02	05	正誤判定	「生成AIの出力は内部で自動的に出典確認が行われているため、利用者による出典確認は不要である」という文の内容は正しいか、誤っているか。	誤っている	正しい		
S1	03	01	空欄補充	「生成AIに入力する文章に氏名・連絡先等の（　）が含まれると、プライバシー侵害のリスクが高まる」という文の空欄に入る最も適切なものはどれか。	個人情報	機密情報	公開情報	メタデータ
S1	03	02	事象理解	他者の著作物を利用する際、作者名や出典を明示する表示を何と言うか。	出典表示	出典確認	断定回避	データ分類
S1	03	03	事象理解	生成AIへの入力可否や共有範囲に迷った場合、責任者に判断を仰ぐ行為を何と言うか。	エスカレーション	スロットリング	量子化	モニタリング
S1	03	04	正誤判定	「生成AIが生成した文章であっても、第三者の著作物の表現が含まれる可能性があるため、著作権やライセンスの確認は必要である」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S1	03	05	正誤判定	「社外秘情報であっても、利用目的が社内業務であれば、生成AIに入力してよい」という文の内容は正しいか、誤っているか。	誤っている	正しい		
S1	04	01	空欄補充	「組織内AIポリシーでは、生成AIの利用を後から監査できるようにシステム側で自動的に残す（　）を定める必要がある」という文の空欄に入る最も適切なものはどれか。	監査ログ	作業記録	出力形式	推論設定
S1	04	02	空欄補充	「生成AIの出力が同じ入力と条件で複数回実行した場合に同程度に再現されるかを点検する観点が（　）である」という文の空欄に入る最も適切なものはどれか。	再現性	事実性	一貫性	透明性
S1	04	03	事象理解	生成AIの出力評価において、内容が現実の事実と合致しているかを点検する観点を何と言うか。	事実性	一貫性	再現性	安全性
S1	04	04	事象理解	生成AIの長文出力で、同一文書内の説明や数値が矛盾しないかを点検する観点を何と言うか。	一貫性	事実性	再現性	網羅性
S1	04	05	正誤判定	「組織内AIポリシーは、一度作れば改訂不要であり、教育や周知の仕組も不要である」という文の内容は正しいか、誤っているか。	誤っている	正しい		
S1	05	01	空欄補充	「生成AIが要約した会議メモの文書で根拠を確認できない箇所は、誤情報の断定を避けるため、「（　）」と明示して残すことがある」という文の空欄に入る最も適切なものはどれか。	要確認	要公開	要承認	要自動化
S1	05	02	事象理解	生成AIの出力を点検し、プロンプトを修正して再実行する反復手順を何と言うか。	修正ループ	承認フロー	データ分類	品質ゲート

S1	05	03	事象理解	生成AIの出力改善をいつ打ち切って共有するかを決める基準を何と言うか。	停止基準	チェックリスト	推論設定	監査ログ
S1	05	04	正誤判定	「生成AIが作成した社内共有文を点検する際、数値や固有名詞は検証優先箇所として根拠資料に照合することが望ましい」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S1	05	05	正誤判定	「生成AIの出力に誤りが見付かった場合も、差分や判断根拠を記録せずに修正後の文章だけ残せば、運用上充分である」という文の内容は正しいか、誤っているか。	誤っている	正しい		
S1	06	01	空欄補充	「生成AIの利用ルールを作る際、情報を「公開」「社内」「機密」「個人情報」等に（　　）して入力可否と条件を定める」という文の空欄に入る最も適切なものはどれか。	区分	翻訳	圧縮	正規化
S1	06	02	空欄補充	「氏名や社員番号等、個人を識別できる情報は（　　）に該当し、生成AIへの入力条件や保存条件を慎重に定める必要がある」という文の空欄に入る最も適切なものはどれか。	PII	API	KPI	SLO
S1	06	03	事象理解	生成AIに入力した内容や判断根拠を後から追跡できるように残す記録を何と言うか。	作業記録	推論設定	出力形式	コンテキスト長
S1	06	04	正誤判定	「生成AI利用のルールでは、入力可否だけ決めればよく、生成物の保存や共有の条件を定める必要はない」という文の内容は正しいか、誤っているか。	誤っている	正しい		
S1	06	05	正誤判定	「機密や個人情報を生成AIに入力する必要がある場合、疑似化等の安全措置を講じた上で条件を定めることが望ましい」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S2	01	01	空欄補充	「大規模言語モデルは、入力文を（　　）と呼ばれる単位に分割し、次の語を予測しながら文章を生成する」という文の空欄に入る最も適切なものはどれか。	トークン	テンプレート	パラメータ	ベクトル
S2	01	02	空欄補充	「生成AIの出力の揺れや長さは、温度やTop Pなどの（　　）を変えることで影響を受け得る」という文の空欄に入る最も適切なものはどれか。	推論設定	学習設定	監査ログ	データ分類
S2	01	03	事象理解	大規模言語モデルに与える入力文の長さを何と言うか。	入力長	出力長	学習率	乱数種
S2	01	04	事象理解	生成AIに提示する基本プロンプトの要素として最も適切なものはどれか。	目的条件出力形式	目的装飾語語尾	条件温度乱数	形式比喩絵文字
S2	01	05	正誤判定	「推論設定の変更は、大規模言語モデルの出力の揺れや長さに影響し得る」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S2	02	01	空欄補充	「生成AIの出力の揺れを把握するためには、同じ入力と設定で（　　）して結果を比較する方法が有効である」という文の空欄に入る最も適切なものはどれか。	反復実行	単回実行	手動採点	暗号化
S2	02	02	空欄補充	「生成AIの回答を毎回同じ形で得るためには、回答の箇条書きや表等の（　　）を明示して固定することが重要である」という文の空欄に入る最も適切なものはどれか。	出力形式	入力禁止	監査ログ	学習率
S2	02	03	事象理解	生成AIの出力を複数回比較した場合に内容や形式の違いとして現れる部分を何と言うか。	差分	余白	復号	係数
S2	02	04	正誤判定	「生成AIの出力を安定化するためには、推論設定を毎回変えることが有効である」という文の内容は正しいか、誤っているか。	誤っている	正しい		
S2	02	05	正誤判定	「出力を安定化するために条件追加と形式固定を行った場合、その理由と結果の変化を作業記録に残すことは再現性確認に役立つ」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S2	03	01	空欄補充	「生成AIの回答を点検する際、数値や固有名詞等の誤りが影響し易い箇所を「（　　）」として先に確認する」という文の空欄に入る最も適切なものはどれか。	検証優先	速度優先	装飾優先	感覚優先
S2	03	02	事象理解	生成AIの出力を共有可能と判断するために設定する最低合格水準を何と言うか。	品質ゲート	停止基準	承認フロー	入力制約
S2	03	03	事象理解	生成AIの出力に根拠が乏しい内容が含まれる場合、言い切りを避けて表現を弱める対応を何と言うか。	断定回避	出典照合	形式固定	自動化
S2	03	04	正誤判定	「生成AIの出力評価で再現性を点検するためには、同一入力と設定で複数回実行して結果が大きく揺れないことを確認する」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S2	03	05	正誤判定	「生成AIの出力評価では、一貫性だけ確認すれば充分であり、事実性の確認は不要である」という文の内容は正しいか、誤っているか。	誤っている	正しい		
S2	04	01	空欄補充	「生成AIの出力を点検する場合、問題点を「誤り」「不足」「冗長」等に整理する（　　）を用いると、作業が体系化される」という文の空欄に入る最も適切なものはどれか。	評価表	設計書	暗号鍵	用語集

S2	04	02	空欄補充	「生成AIの出力を是正するために、指示文を修正して同じ課題を再生成させる操作を（　）として実施する」という文の空欄に入る最も適切なものはどれか。	再プロンプト	再学習	再配布	再起動
S2	04	03	事象理解	生成AIの出力を検証する際、原契約書や原仕様書等、一番に根拠になる資料を何と言うか。	一次情報	二次資料	参考意見	推論設定
S2	04	04	事象理解	生成AIの出力を評価して再プロンプトで是正し、品質ゲートを充すまで繰り返す手順を何と言うか。	改善ループ	承認フロー	データ分類	スロットリング
S2	04	05	正誤判定	「再プロンプトでの是正では、誤りの根拠を一次情報に照合した上で断定回避を含む修正を行うことが望ましい」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S2	05	01	空欄補充	「生成AIに特定の立場や役割を与えて回答方針を揃えるパターンが（　）である」という文の空欄に入る最も適切なものはどれか。	役割付与	量子化	暗号化	蒸留
S2	05	02	事象理解	複雑な要求を論点毎に分け、各論点の回答を統合するように生成AIに指示する設計を何と言うか。	分割統治	役割付与	出力例提示	スロットリング
S2	05	03	事象理解	生成AIが誤誘導されないよう、してはいけないことや参照範囲を明示する記述を何と言うか。	制約条件	推論設定	監査ログ	疑似化
S2	05	04	正誤判定	「生成AIへの要求が曖昧な場合に追加質問を先に行わせる設計は、誤解のまま回答を生成するリスクを下げ得る」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S2	05	05	正誤判定	「分割統治を用いる場合、論点を分けずに一括で回答させる方が再現性が高くなる」という文の内容は正しいか、誤っているか。	誤っている	正しい		
S2	06	01	空欄補充	「同じ業務課題に対して二つのプロンプト案を作り、品質と再現性を比べて採択する行為は（　）に当る」という文の空欄に入る最も適切なものはどれか。	比較評価	暗号化	発散学習	量子化
S2	06	02	空欄補充	「複数案から最終案を選んだ理由や注意点を短文で整理して（　）として残す」という文の空欄に入る最も適切なものはどれか。	設計意図	学習率	監査ログ	エンコード
S2	06	03	事象理解	業務で利用する回答の様式を揃えるために、生成AIへの指示で項目名や構造を固定することを何と言うか。	出力形式固定	推論設定変更	データ分類	例外承認
S2	06	04	正誤判定	「複数のプロンプト案を比較する際、同一入力と同一推論設定で反復実行し、出力の揺れも含めて評価することが有効である」という文の内容は正しいか、誤っているか。	正しい	誤っている		
S2	06	05	正誤判定	「業務プロンプトを作成する際、役割付与や制約条件を入れずに自由に回答させた方が誤誘導を避け易い」という文の内容は正しいか、誤っているか。	誤っている	正しい		

AI 補完型 IT エンジニア養成プログラム
プラットフォーム設計書

文書分類	設計書
システム名称	AI 補完型 IT エンジニア養成プログラム プラットフォーム
バージョン	4.1.0
最終更新日	2025/12/20

目次

1. システム概要 P.1
2. システム構成 P.2
3. DB テーブル一覧 P.3
4. DB テーブル定義詳細 P.3
5. セキュリティ仕様 P.6
6. インフラ構成仕様 P.7
7. 外部インターフェース一覧（インフラ層） P.8

---

1. システム概要

1.1. 背景

従来のオンプレミス型や VM（仮想マシン）ベースのインフラストラクチャでは、スケーラビリティの確保やサーバーメンテナンスに多大なコストと人的リソースが必要とされていました。特に、教育機関向けシステムでは、テスト実施時など特定の時間帯にアクセスが集中する傾向があり、ピーク時に合わせた過剰なリソース確保や、急激な負荷変動への対応が課題となっていました。 また、AI 技術の進展に伴い、セキュリティを担保しつつ、外部 API との安全な連携や高負荷な計算処理を実行できる柔軟な基盤が求められています。

1.2. 目的

本プラットフォームは、Google Cloud Platform (GCP) および Firebase を全面的に採用した、フルマネージドかつサーバーレスなアーキテクチャの構築を目的としています。 インフラ管理の負担を最小限に抑えつつ、高い可用性と柔軟なスケーラビリティを実現します。また、機密性の高い教育データや個人情報を保護するため、認証・認可基盤として Firebase Authentication と Security Rules を活用し、多層的なセキュリティ対策を実装します。さらに、Python コードの実行環境として隔離されたサンドボックス（Cloud Run）を用意し、安全なプログラミング学習環境を提供します。

1.3. 導入効果

1.3.1. 運用コストの最適化と負荷耐性

- **完全なサーバーレス構成**: サーバーのプロビジョニングやパッチ適用が不要となり、運用工数を大幅に削減します。
- **オートスケーリング**: Cloud Functions と Cloud Run により、トラフィックの増減に応じてリソースが自動的にスケールアウト/インするため、アクセス集中時でも安定したパフォーマンスを維持しつつ、アイドル時のコストを最小化します。

1.3.2. 高度なセキュリティと信頼性

- **堅牢な認証・認可基盤**: Firebase Authentication による安全な ID 管理と、Firestore Security Rules による詳細なアクセス制御により、不正アクセスやデータ漏洩のリスクを低減します。

- **隔離された実行環境**: ユーザーが記述した任意のコードは、ネットワークから隔離された Cloud Run コンテナ上で実行されるため、ホストシステムへの影響を防ぎ、安全性を担保します。

1.3.3. 開発・デプロイの俊敏性

- **高速なコンテンツ配信**: Firebase Hosting (CDN) により、低レイテンシでアプリケーションにアクセス可能です。
- **統一された開発体験**: フロントエンドからバックエンドまで一貫した Google Cloud エコシステムを利用することで、開発サイクルの短縮と品質向上を実現します。

---

2. システム構成

2.1. 全体アーキテクチャ

本システムは **Google Cloud Platform (GCP)** および **Firebase** を基盤とした、フルマネージドでスケーラブルなサーバーレスアーキテクチャを採用している。

- **Frontend Hosting**: **Firebase Hosting**
  - 静的アセット（React/Vite ビルド生成物）の高速配信（CDN）。
- **Backend Logic**: **Cloud Functions for Firebase (2nd Gen)**
  - Node.js 20 ランタイム。
  - API エンドポイントおよびバッチ処理、Firestore トリガーの実行。
- **Database**: **Cloud Firestore**
  - NoSQL ドキュメントデータベース。リアルタイム同期、オフラインサポート。
- **Storage**: **Cloud Storage for Firebase**
  - 講義資料（PDF/動画）、画像アセットの格納。
- **Authentication**: **Firebase Authentication**
  - ID/Password による認証基盤。
- **External Compute**: **Cloud Run** (Executor Service)
  - Python コード実行用の隔離されたサンドボックス環境（コンテナ）。

2.2. ネットワーク構成

- **リージョン**:
  - **Firestore / Cloud Functions**: asia-northeast1 (東京) 推奨。
    レイテンシ（通信遅延）低減のため、ユーザー地理的所在地に合わせる。
  - **Cloud Run**: 同一リージョン（asia-northeast1）に展開し、
    通信遅延を最小化。
- **接続制御**:
  - **Cloud Functions -> Cloud Run**:
    - インターネット経由の HTTPS 通信を使用（将来的な VPC Connector 利用を想定）。
    - 認証: IAM 認証（roles/run.invoker）を導入予定。

- **Cloud Functions -> Gemini API**:
  - Google の公開 API エンドポイントへアクセス（API Key 認証）。

3. DB テーブル一覧（Firestore Collections）

コレクション名	ドキュメント ID	用途・格納データ概要
users	uid (Auth ID)	ユーザーの基本情報（表示名, メール, role, 作成日時）
courses	uuid (自動生成)	コースメタデータ、サムネイル URL、公開ステータス
enrollments	courseId_userId	受講登録情報。ユーザーとコースの多対多リレーション、進捗率
questionBank	uuid (自動生成)	問題データ（本文, タイプ, 選択肢, 正解/コード, スキルタグ）
testSets	uuid (自動生成)	テストセット情報（タイトル, 問題 ID 配列, 合格ライン, 制限時間）
testSubmissions	uuid (自動生成)	テスト回答結果（ユーザーID, テスト ID, 各問の正誤・スコア, 合否）
resources	uuid (自動生成)	講義資料メタデータ（タイトル, ファイル種別, Storage パス）
skillCriteria	uuid (自動生成)	スキル評価基準（ルーブリック）, 階層化されたスキル定義
dashboard_metrics	latest (固定)	ダッシュボード表示用の集計済みデータ（ランキング, 不正解率等）
lectureMaterials	-	*(Legacy: 実装上は resources に統合対象)*

---

4. DB テーブル定義詳細

4.1. users (ユーザー)

フィールド名	型	必須	説明
uid (DocID)	String	Yes	Auth ユーザーID
displayName	String	Yes	表示名
email	String	Yes	メールアドレス
role	String	Yes	'teacher' または 'student'
createdAt	Timestamp	Yes	アカウント作成日時

4.2. courses (コース)

フィールド名	型	必須	説明
uuid (DocID)	String	Yes	コース ID (自動生成)
title	String	Yes	コースタイトル
description	String	No	コース概要
skillCriteriaId	String	No	紐づくスキル評価基準 ID (skillCriteria への参照)
isPublished	Boolean	Yes	公開状態 (true: 公開, false: 非公開)
authorId	String	Yes	作成者のユーザーID (users への参照)
thumbnailURL	String	No	サムネイル画像のダウンロード URL
thumbnailStoragePath	String	No	サムネイル画像の Storage パス（削除用）
activities	Array	No	学習活動（教材・テスト）のリスト
└ id	String	Yes	活動ブロック ID

└ type	String	Yes	'resource'（教材）または 'test'（テスト）
└ title	String	Yes	活動タイトル
└ contentId	String	No	実コンテンツの ID (resources または testSets への参照)
└ keywords	Array	No	検索用キーワード
createdAt	Timestamp	Yes	作成日時
updatedAt	Timestamp	Yes	更新日時

4.3. enrollments（受講登録）

フィールド名	型	必須	説明
courseId_userId (DocID)	String	Yes	コース ID とユーザーID の複合キー
userId	String	Yes	受講者のユーザーID
courseId	String	Yes	受講コース ID
progress	Number	Yes	進捗率（0-100）
completedBlocks	Array	Yes	完了した活動ブロック ID のリスト
startedAt	Timestamp	Yes	受講開始日時

4.4. questionBank（問題バンク）

フィールド名	型	必須	説明
uuid (DocID)	String	Yes	問題 ID（自動生成）
type	String	Yes	'multiple-choice', 'fill-in-the-blank', 'programming'
text	String	Yes	問題文（Markdown 対応）
skillId	String	No	スキル ID（評価基準に基づく階層 ID）
skillArea	String	No	スキル領域名
difficulty	String	No	'level_beginner', 'level_intermediate', 'level_advanced'
（多肢選択）			
options	Array	Cond	選択肢のリスト（文字列配列）
correctOptionIndex	Number	Cond	正解の選択肢インデックス（0 開始）
（穴埋め）			
correctAnswer	String	Cond	正解文字列
（プログラミング）			
language	String	Cond	プログラミング言語（例: 'python'）
initialCode	String	Cond	初期コード（テンプレート）
solutionCode	String	Cond	模範解答コード
testCases	Array	Cond	採点用テストケースのリスト
└ callCode	String	Yes	実行コード
└ expectedOutput	String	Yes	期待される出力

4.5. testSets（テストセット）

フィールド名	型	必須	説明
uuid (DocID)	String	Yes	テストセット ID
title	String	Yes	テストタイトル

description	String	No	説明文
skillCriteriaId	String	No	スキル評価基準 ID
questionIds	Array	Yes	含まれる問題 ID のリスト（questionBank への参照）
duration	Number	Yes	制限時間（分）
passMark	Number	Yes	合格点（0-100）
isPublished	Boolean	Yes	公開フラグ
lectureMaterialId	String	No	関連する講義資料 ID (resources への参照)

4.6. testSubmissions（テスト提出結果）

フィールド名	型	必須	説明
uuid (DocID)	String	Yes	提出 ID
userId	String	Yes	受講者 ID
testId	String	Yes	テストセット ID
testTitle	String	Yes	受験時のテストタイトル（スナップショット）
score	Number	Yes	総合スコア（0-100）
passed	Boolean	Yes	合否結果
submittedAt	Timestamp	Yes	提出日時
answers	Map	Yes	各問題の回答データ（Key: questionId）
└ answer	Any	Yes	ユーザーの回答
└ isCorrect	Boolean	Yes	正誤
└ score	Number	Yes	問題ごとのスコア

4.7. resources（教材リソース）

フィールド名	型	必須	説明
uuid (DocID)	String	Yes	リソース ID
title	String	Yes	教材タイトル
type	String	Yes	'document', 'video', 'excel' 等
storagePath	String	Yes	Firebase Storage 上のパス
createdAt	Timestamp	Yes	アップロード日時

4.8. skillCriteria（スキル評価基準）

フィールド名	型	必須	説明
uuid (DocID)	String	Yes	評価基準 ID
rubricTitle	String	Yes	評価基準セットのタイトル
evaluationItems	Array	Yes	評価項目リスト
└ **skillArea**	String	Yes	スキル領域名
└ **level_***	Object	Yes	各レベル(beginner, intermediate, advanced）の定義
└ **description**	String	Yes	レベルの説明
└ **tags**	Array	Yes	関連スキルタグ

4.9. dashboard_metrics (集計データ)

フィールド名	型	必須	説明
latest (DocID)	String	Yes	固定 ID
dailySubmissions	Array	Yes	日別提出数推移データ
lowProficiencySkills	Array	Yes	苦手スキルランキングデータ
studentRanks	Map	Yes	テストセット別の受講者ランキング
passFailRate	Map	Yes	テストセット別の合格率データ
lastUpdated	Timestamp	Yes	最終集計日時

---

5. セキュリティ仕様

5.1. 認証 (Authentication)

- **プロバイダ**: Firebase Authentication (Identity Platform)
- **認証方式**: Email/Password 認証を採用。
- **セッション**: Firebase SDK による JWT (JSON Web Token) 管理 (ID Token)。
  クライアント側で自動リフレッシュ。
- **パスワードポリシー**: Firebase Default 設定に準拠 (6 文字以上)。

5.2. アクセス制御 (Authorization)

アクセス制御は「データアクセス層 (Firestore)」と「ビジネスロジック層 (Functions)」で多層防御を行う。

A. Firestore Security Rules (データアクセス制御)

firestore.rules により、データレベルで厳格な権限管理を行う。

コレクション	読み取り (Read/Get/List)	書き込み (Create/Update/Delete)	備考
users	認証済みユーザー全体	自分のみ (request.auth.uid == uid)	他人のプロフィール改ざんを防止
courses	認証済みユーザー全体	教員のみ (resource.data.role == 'teacher')	受講者は閲覧のみ
questionBank	認証済みユーザー全体	教員のみ	
testSets	認証済みユーザー全体	教員のみ	
testSubmissions	教員 または 本人	本人のみ (作成) / 教員のみ (削除)	他人の成績閲覧を防止
enrollments	教員 または 本人	本人のみ	
resources	認証済みユーザー全体	教員のみ	

B. Cloud Functions Security (サーバーサイドロジック制御)

Callable Functions 内でコンテキスト (context.auth) を検証し、不正な実行を阻止する。

- **ロールベース制御 (RBAC)**:
  - 教員専用関数 (createUser, updateUser 等) は、実行ユーザーの token.role または DB 上のロールを確認し、teacher 以外ならエラー (PermissionDenied) を返す。

- **認証必須制御**:
    - 一般関数（submitTest, generateTest）は context.auth が存在することを確認し、未認証ならエラー（Unauthenticated）を返す。

5.3. API キー・環境変数管理（Secrets Management）

- **Gemini API Key**:
    - ソースコードへのハードコードは厳禁。
    - **Firebase Functions Config** (functions.config().gemini.key) または **Google Cloud Secret Manager** で管理。
    - 関数実行時にサーバーサイドでのみ参照する。
- **Client Config**:
    - firebase.js には apiKey, authDomain, projectId 等の公開情報を安全に含める（これらは公開してもリソースへの不正アクセスは Security Rules で防がれるため問題ない）。

---

6. インフラ構成仕様

6.1. 実行環境（Cloud Run Executor）

信頼できないコードを実行するためのサンドボックス環境。

- **Image**: Python 3.9+ ベースの軽量カスタムイメージ（Alpine or Slim）。
- **Dependencies**: データサイエンス・数値計算用の主要ライブラリをプリインストール。
    - numpy, pandas, scikit-learn, matplotlib
- **セキュリティ要件（Sandbox）**:
    - **ネットワーク遮断**: 外部への通信を一切許可しない（不正なデータ送信や踏み台化の防止）。
    - **リソース制限**:
        - CPU: 1 vCPU 以下
        - Memory: 512MB - 1GB
        - Timeout: 10 秒 - 30 秒（無限ループ防止）
    - **ステートレス**: 実行ごとにコンテナ/環境はリセットされ、ファイルは永続化されない。

6.2. ストレージ（Cloud Storage）

- **バケット構成**:
    - course_thumbnails/{uuid}_{filename}: コースのサムネイル画像。
    - resources/{uuid}_{filename}: 講義資料ファイル（PDF, Video 等）。

- **アクセス権**:
    - 読み取り: 認証済みユーザー全体に許可。
    - 書き込み: 教員アカウントのみに許可（storage.rules）。

6.3. スケーラビリティと制限

- **Cloud Functions**:
  - インスタンス数: 自動スケーリング（0 to N）。
  - コールドスタート対策: 必要に応じて minInstances を設定（現状は未設定、コスト優先）。
- **Firestore**:
  - 書き込み制限: 約 1 回/秒（ドキュメント単位）。バッチ書き込み時は 500 件/リクエスト制限。
  - クエリ制限: IN 句は最大 30 要素まで（アプリ側でチャンク分割処理を実装済み）。

---

7. 外部インターフェース一覧（インフラ層）

7.1. 外部接続詳細

接続先	プロトコル	認証	用途
Google Gemini API	gRPC / REST	API Key	テスト生成、フィードバック生成 AI。 モデル: gemini-3 系を推奨（2025 年 12 月現在）
Google Cloud Run	HTTPS	IAM (OIDC Token) / Internal	Python コード実行サービスのエンドポイント。
Firebase Auth	HTTPS	API Key	ユーザー認証、トークン発行。
Firebase Storage	HTTPS	Security Rules / Signed URL	ファイルのアップロード・ダウンロード。

7.2. 内部環境変数定義

バックエンド（functions/.env または Runtime Config）で管理すべき値。

- GEMINI_API_KEY: 生成 AI 利用のための API キー。
- EXECUTOR_URL: Cloud Run サービスの URL。
- TZ: Asia/Tokyo（ログおよびスケジューラ用）。

以　上

5.5　GPT-TA評価システム――設計書

AI 補完型 IT エンジニア養成プログラム
アプリケーション設計書

文書分類	設計書
システム名称	AI 補完型 IT エンジニア養成プログラム アプリケーション
バージョン	5.1.0
最終更新日	2025/12/20

目次

1. システム概要 P.1
2. コーディング規約 P.2
3. 機能一覧 P.3
4. 処理機能記述 P.4
5. 画面一覧 P.7
6. 画面仕様（主要画面詳細） P.9
7. 項目ラベル名一覧 P.15
8. エラーメッセージ P.17
9. ファイル一覧 P.18
10. 外部インターフェース仕様（API） P.19
11. バッチ処理一覧 P.21
12. アプリケーション仕様 P.21

---

1. システム概要

1.1. 背景

現代の IT 社会において、高度なスキルを持つ IT エンジニアの不足は深刻な課題となっています。従来の教育手法では、座学中心の知識習得に偏りがちであり、実務で求められる「実践的なコーディング能力」や「問題解決能力」を効率的に育成することが困難でした。 また、教育現場においては、学習者一人ひとりのコードを詳細にレビューし、適切なフィードバックを行うために多大な人的リソースが必要とされ、指導者の負担増加やフィードバックの遅れが課題となっています。

1.2. 目的

本システム「AI 補完型 IT エンジニア養成プログラム アプリケーション」は、最新の生成 AI (Gemini) とクラウド技術を活用し、これらの課題を解決することを目的としています。 教員に代わって AI が問題作成、採点、フィードバックを行うことで、質の高い教育をスケーラブルに提供します。特に、プログラミング学習においては、実際にコードを実行・検証する環境を備え、単なる知識確認にとどまらない実践的なスキルの習得を支援します。

1.3. 導入効果

1.3.1. 受講者への効果

- **リアルタイムフィードバックによる学習効率の向上**: 提出したコードや回答に対し、即座に採点結果と AI による詳細な解説が得られるため、躓きをすぐに解消し、学習サイクルを加速させることができます。
- **実践的スキルの習得**: 実際のプログラム実行環境を用いたテストにより、現場で通用するコーディング能力を養うことができます。
- **モチベーションの維持・向上**: ダッシュボードやスキル分析機能により、自身の学習進捗や苦手分野が可視化され、自律的な学習を促進します。

1.3.2. 教育者・管理者への効果

- **業務負荷の劇的な削減**: テストの自動生成・自動採点機能により、準備や評価にかかる時間を大幅に削減し、より本質的な指導やメンタリングに時間を割くことが可能になります。
- **データに基づく指導の最適化**: 受講者全体の進捗や誤答傾向を AI が分析し、講義資料の改善点や重点的に指導すべきポイントを提案するため、教育の質を継続的に向上させることができます。
- **教材作成の効率化**: AI アシスタントにより、多様なレベル・形式の問題や教材を短時間で作成することができます。

2. コーディング規約

2.1. 基本方針

- **フロントエンド**: React (Vite) + Tailwind CSS
- **バックエンド**: Firebase Cloud Functions (Node.js 20, ES Modules)
- **補助システム**: Google Cloud Run (Python Executor Service)

2.2. リンター・フォーマッター設定

フロントエンド（frontend/）

- **Lint Config**: eslint.config.js (Flat Config)
- **Extends**: js.configs.recommended, reactHooks.configs.recommended, reactRefresh.configs.vite
- **ルール**:
  - no-unused-vars: エラー（_で始まる変数は無視）
  - react-refresh/only-export-components: 警告
- **フォーマット規約** (Observed):
  - **インデント**: **4 スペース**
  - **クォート**: シングルクォート（'）推奨（JSX 属性はダブルクォート）
  - **セミコロン**: 必須
  - **コンポーネント定義**: アロー関数（const Comp = () => { ... }）

バックエンド（functions/）

- **Lint Config**: .eslintrc.cjs
- **Extends**: google (Google JavaScript Style Guide 準拠)
- **ルール特記事項**:
  - quotes: ダブルクォート（"）必須（テンプレートリテラルは許可）
  - prefer-arrow-callback: エラー
  - no-restricted-globals: name, length の使用禁止
- **フォーマット規約**:
  - **インデント**: **2 スペース** (Google Style)
  - **セミコロン**: 必須
  - **Console 出力**: console.log ではなく firebase-functions/logger を使用すること

2.3. 命名規則

- **変数名・関数名**: camelCase (例: fetchUserRole, calculateStudentRanks)
- **定数**: UPPER_SNAKE_CASE (例: IN_QUERY_LIMIT)
- **React コンポーネント**: PascalCase (例: AnalysisPage, StudentDashboard)
- **ファイル名**:
  - コンポーネント: PascalCase.jsx (例: TestPage.jsx)
  - フック: camelCase.js (例: useAuthState.js)
  - 設定・ユーティリティ: camelCase.js または kebab-case.js

2.4. コメント規約

- **JSDoc**: 関数やクラスの定義時には、引数・戻り値を説明する JSDoc (/** ... */) を記述する。
- **TODO**: 未実装箇所や修正が必要な箇所には // TODO: または // FIXME: を付与する。

2.5. エラーハンドリング規約

- **バックエンド（Cloud Functions）**:

- クライアントへのエラー返却には必ず HttpsError を使用する。
- 予期せぬエラーは internal コードでラップし、詳細は logger.error に記録する。

---

3. 機能一覧

3.1. 教員用機能（Teacher Role）

大分類	機能 ID	機能名称	詳細機能・備考
ダッシュボード	T-001	ダッシュボード表示	・システム全体の統計（受講者数、テスト実施数） ・要注意受講者のアラート表示 ・システム稼働状況の確認
コース管理	T-002	コース編集・作成	・コース新規作成、基本情報（タイトル、概要）編集 ・サムネイル画像のアップロード ・公開/非公開の設定
	T-003	活動（Activity）管理	・コースへの教材・テストの追加・削除 ・ドラッグ＆ドロップによる順序変更 ・活動ごとのキーワード設定
	T-004	進捗管理	・コースごとの受講者の進捗率確認 ・完了/未完了ステータスの管理
問題・テスト管理	T-005	問題作成（AI 生成）	・Gemini API を用いた自動生成 ・対応形式: 選択式、穴埋め、プログラミング ・生成パラメータ: トピック、難易度、問題数
	T-006	問題作成（手動）	・マニュアル入力による問題作成 ・プログラミング問題のテストケース定義 （入力/出力/セットアップコード）
	T-007	問題バンク管理	・作成済み問題の一覧表示、検索、フィルタリング ・既存問題の再編集・削除
	T-008	テストセット管理	・複数の問題を組み合わせた「テストセット」の作成 ・合格点、制限時間（将来構想）等の設定
分析・評価	T-009	マルチモーダル分析	・テスト結果と講義資料（PDF/テキスト） を連携させた AI 分析 ・誤答傾向に基づく教材改善提案の生成
	T-010	スキル/ルーブリック管理	・評価基準（ルーブリック）の定義 ・スキルセットの編集とコースへの紐付け
	T-011	結果詳細確認	・受講者個人のテスト提出内容の閲覧 ・プログラミングコードの diff 確認 ・実行結果ログの確認
ユーザー管理	T-012	受講者管理	・受講者アカウントの一覧表示、登録、削除 ・（管理者機能としての側面あり）

3.2. 受講者用機能（Student Role）

大分類	機能 ID	機能名称	詳細機能・備考
学習	S-001	ダッシュボード表示	・学習進捗の概況表示 ・最新のお知らせ確認
	S-002	コース受講	・受講可能なコースの一覧閲覧 ・コース詳細画面へのアクセス
	S-003	教材閲覧	・PDF やテキスト形式の講義資料の閲覧
テスト	S-004	テスト受験	・Web インターフェース上での解答 ・プログラミングコードの記述とブラウザ上での実行（バックエンド処理） ・リアルタイム正誤判定（選択式など）
	S-005	フィードバック確認	・採点結果（スコア、合否）の即時確認 ・AI によるコードレビュー・改善アドバイスの閲覧
履歴	S-006	結果履歴	・過去に受験したテストの結果履歴一覧 ・自身のスキル習得状況の可視化（レーダーチャート等）

---

4. 処理機能記述

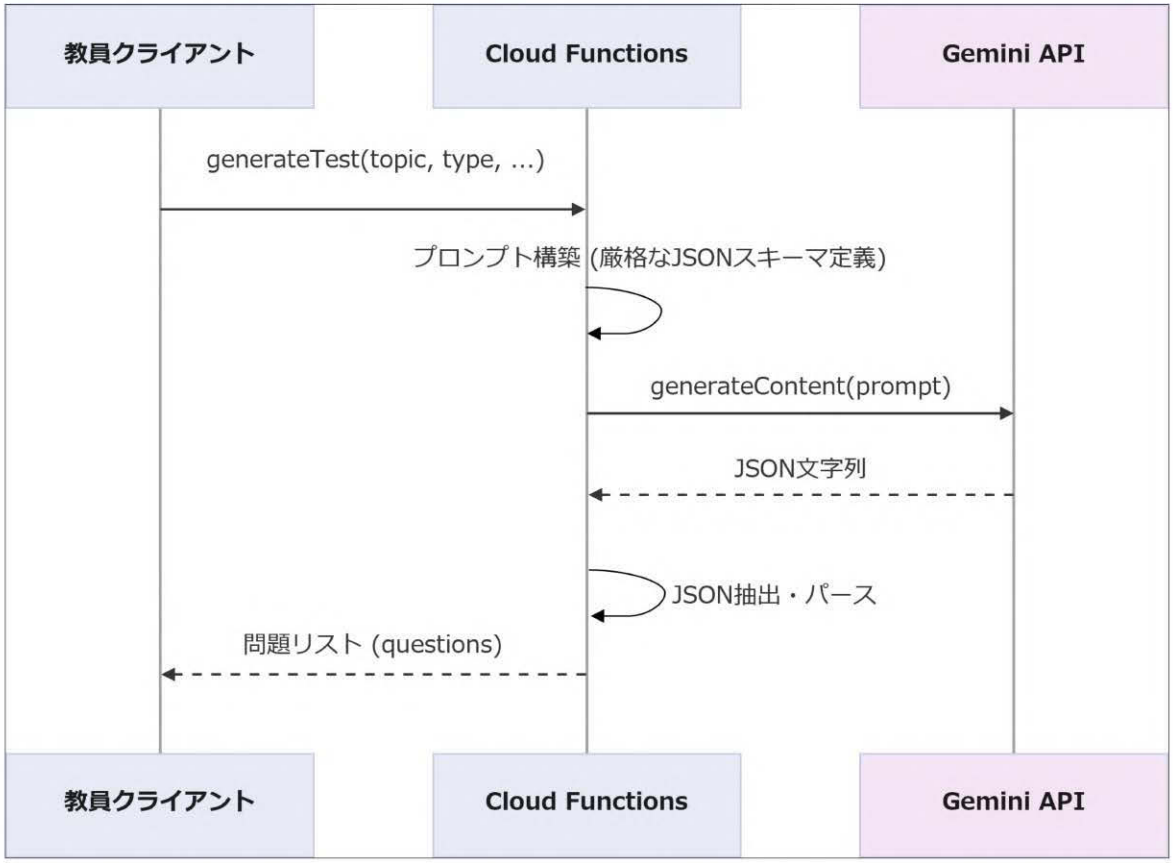
4.1. テスト生成（generateTest）

概要: 教員の指定したトピックに基づいて、AI がテスト問題を自動生成します。特にプログラミング問題では、正解コードとテストケースの整合性を担保する仕組みがあります。

処理フロー:

1. **リクエスト**: 教員がトピック、問題数、形式、言語を指定。
2. **プロンプト構築**:
    - プログラミング問題の場合：「Python の専門家」としての役割を与え、JSON 形式（solutionCode, testCases を含む）での出力を強制。
    - **自己検証指示**: AI に対し、「自身の生成した正解コードを実行し、テストケースの期待値と一致すること」を強く指示。
3. **生成とパース**: Gemini API からの応答（テキスト）から JSON 部分を抽出し、オブジェクト化。
4. **レスポンス**: 生成された問題リストをクライアントに返却（この時点では DB 保存されない）。

テスト生成（generateTest）シーケンス図



4.2. テスト提出・採点（submitTest）

概要: 受講者の解答を受け取り、自動採点を行います。プログラミング問題は Google Cloud Run 上で稼働する Executor Service を使用して実際にコードを実行して評価します。

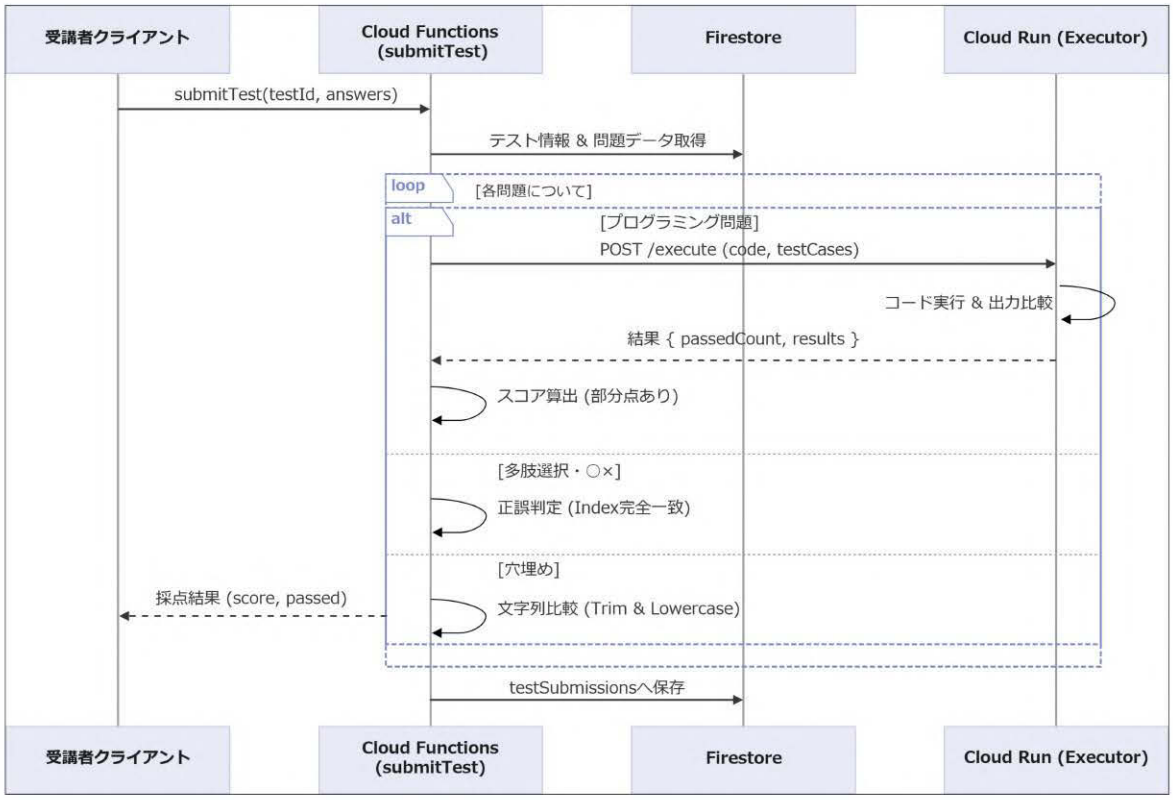
処理フロー:

1. **データ取得**: testId を元にテスト情報を取得。問題 ID リストから IN クエリ（30 件分割）を用いて全問題データを取得。
2. **ループ処理**: 各問題について採点を実施。
    - **多肢選択（Multiple Choice）**:
        - **○×問題（True/False）** を含む。
        - ロジック: userAnswer（選択された index）が correctOptionIndex と完全一致する場合に正解（100 点）。それ以外は 0 点。
    - **穴埋め（Fill-in-the-blank）**:
        - ロジック: userAnswer（文字列）と correctAnswer（文字列）を比較。
        - **正規化**: 両方の文字列に対して .trim().toLowerCase() を適用し、大文字・小文字の差異や前後の空白を無視して比較する。完全一致で 100 点。
    - **プログラミング**:
        - Cloud Run 上の Executor Service へ language, studentCode, testCases を送信。
        - Executor はコンテナ環境でコードを実行。
        - テストケースごとの通過数（passedCount）に基づいて部分点を算出。
3. **集計・保存**: 総合スコアと合否判定を行い、testSubmissions コレクションに保存。

Executor Service の検証ロジック:

- Python の pandas, numpy, matplotlib 等をサポート。
- **数値判定**: 浮動小数点の誤差（1e-9）を許容。
- **特殊判定**: K-Means や PCA など、順序が不定な出力のための専用比較ロジックを実装。

テスト提出・採点（submitTest）シーケンス図



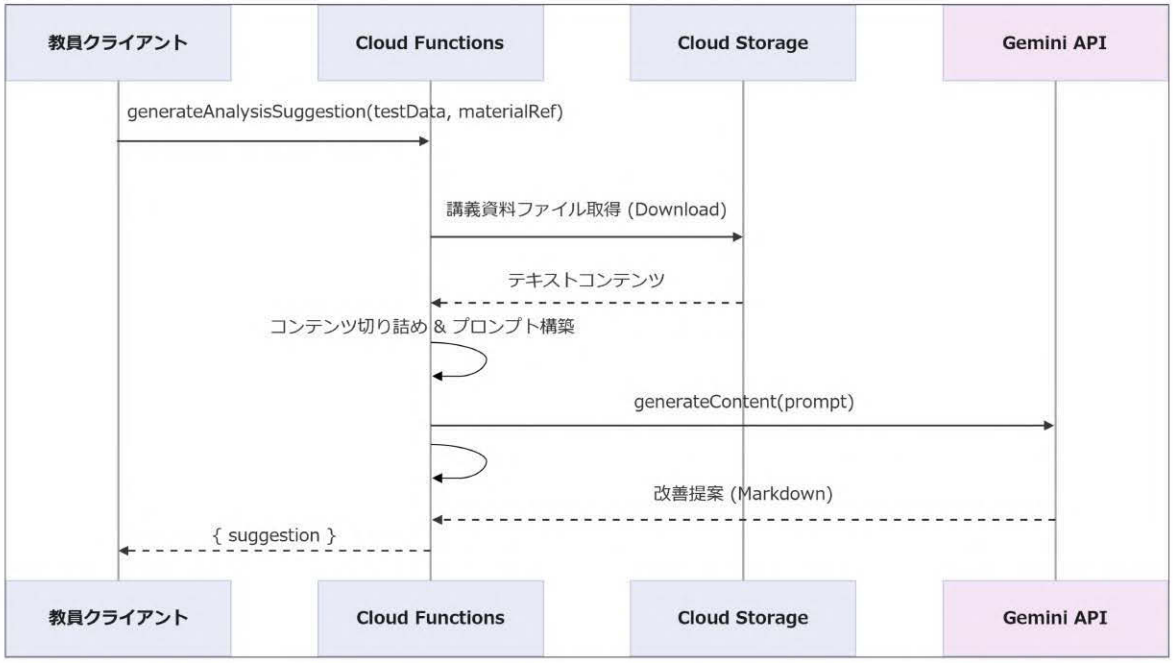
4.3. AI 分析提案（generateAnalysisSuggestion）

概要: テスト結果と講義資料を突き合わせ、AI が講義資料の改善案を提示します。

処理フロー:

1. **データ準備**:
   - クライアントから「誤答率の高い問題リスト」を受領。
   - **講義資料取得**: まず Firebase Storage からのダウンロードを試行。失敗または未設定時は Firestore のスニペットを使用。
2. **プロンプト構築**:
   - 「教育工学の専門家」としての役割を設定。
   - 誤答パターンから学生の躓きポイントを推測させ、資料のどの部分をどう修正すべきか（具体的アクション）をリスト形式で出力するように指示。
3. **生成**: Gemini API により改善提案マークダウンを生成。

AI 分析提案（generateAnalysisSuggestion）シーケンス図



5. 画面一覧

5.1. 共通・認証（Common / Auth）

画面 ID	画面名称	URL / Route	ページコンポーネント	主な機能・アクション
G-001	ログイン画面	/login	LoginPage	・メールアドレス/パスワード認証 ・ログイン失敗時のエラー表示
G-000	リダイレクター	/	RootRedirector	・ロールに基づくダッシュボードへの自動転送 ・認証状態のチェック

5.2. 教員用画面（Teacher Role）

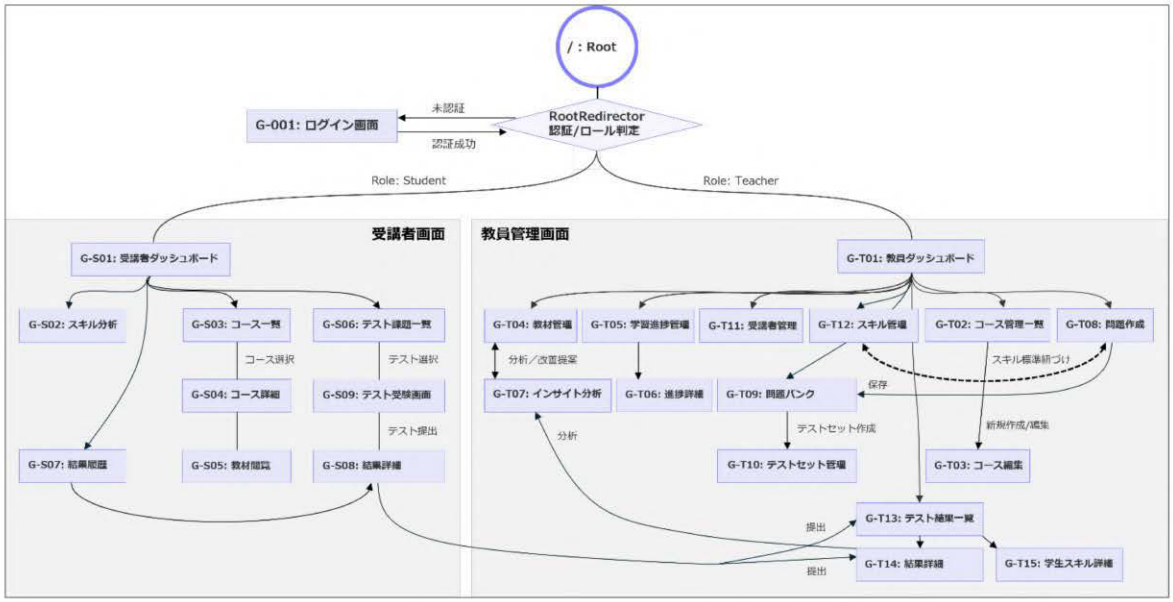
画面 ID	画面名称	URL (/teacher/...)	ページコンポーネント	主な機能・アクション
G-T01	ダッシュボード	/ (index)	Dashboard	・システム全体の統計確認 ・要注意学生のアラート確認
G-T02	コース管理一覧	/course-management	CourseManagementPage	・作成済みコースの検索・一覧表示 ・新規作成ボタン、編集ボタン
G-T03	コース編集	/course-management/edit/:id /course-management/new	CourseEditPage	・基本情報（タイトル、サムネイル）編集 ・活動（教材・テスト）の追加・並べ替え
G-T04	教材管理	/resources	ResourceManagementPage	・講義資料（PDF 等）のアップロード ・資料のプレビュー、削除
G-T05	学習進捗管理	/progress	ProgressManagementPage	・コースごとの進捗率一覧表示

				・詳細ページへの遷移
G-T06	進捗詳細	/progress/:courseId	CourseProgressDetailPage	・受講者ごとの進捗状況マトリクス表示
G-T07	インサイト分析	/analysis	AnalysisPage	・テスト結果の AI 分析実行 ・改善提案の閲覧とコピー
G-T08	問題作成（各種）	/create /edit/general /edit/programming	QuestionCreationPage	・AI による問題生成（Gemini 使用） ・手動による問題編集・作成
G-T09	問題バンク	/manage/questions	QuestionBank	・ストック問題の一覧、検索、編集、削除
G-T10	テストセット管理	/manage/test-sets	TestSetManagement	・問題セットの組み立て ・配点、合格点設定
G-T11	受講者管理	/manage/students	StudentManagementPage	・受講者アカウントの作成、編集、削除
G-T12	スキル管理	/manage/skills	SkillManagementPage	・評価基準（ルーブリック）の定義 ・スキルレベル設定
G-T13	テスト結果一覧	/results	TestResultsPage	・全受講者のテスト提出履歴一覧
G-T14	テスト結果詳細	/results/:submissionId	TeacherResultDetailPage	・個別の提出内容（コード、正誤）の詳細確認
G-T15	学生スキル詳細	/skills/:userId	StudentSkillsPage	・特定学生のスキル習得状況確認

5.3. 受講者用画面（Student Role）

画面 ID	画面名称	URL (/student/...)	ページコンポーネント	主な機能・アクション
G-S01	ダッシュボード	/ (index)	StudentDashboardContent	・学習の進捗概要、お知らせ表示
G-S02	スキル分析	/skills	StudentSkillsPage	・自身のスキル習得状況の可視化
G-S03	コース一覧	/courses	StudentCourseList	・受講可能コースの閲覧
G-S04	コース詳細	/courses/:courseId	StudentCourseViewerPage	・コース内アクティビティ（教材・テスト）の選択
G-S05	教材閲覧	/resource/:resourceId	ResourceViewerPage	・PDF/テキスト資料の学習
G-S06	テスト課題一覧	/tests	AssignmentList	・未受験/受験済みテストの一覧
G-S07	結果履歴	/results	StudentResultsPage	・過去のテスト結果一覧表示
G-S08	結果詳細	/results/:submissionId	StudentResultDetailPage	・採点結果の確認 ・AI フィードバックの確認
G-S09	テスト受験画面	/test/:testId (Root Path)	TestPage	・問題解答（選択、記述、コード） ・リアルタイム採点実行 ・(注: Role に関わらずテスト実行時はここを使用)

5.4. 画面遷移図



---

6. 画面仕様（主要画面詳細）

6.1. 共通・認証（Common / Auth）

[G-001] ログイン画面

- **URL**: /login
- **コンポーネント**: LoginPage.jsx
- **UI 要素**:
  - メールアドレス入力フィールド（type="email", required）
  - パスワード入力フィールド（type="password", required）
  - 「ログイン」ボタン（Submit）
  - エラーメッセージ表示領域（赤文字）
- **主なアクション**:
  - **ログイン**: Firebase Auth（signInWithEmailAndPassword）を実行。
  - 成功時は / へ遷移。失敗時はエラーを表示。
- **備考**: すでにログイン済みの場合は / へ自動リダイレクト。

[G-000] リダイレクター（Root）

- **URL**: /
- **コンポーネント**: main.jsx (RootRedirector)
- **ロジック**:
  - 未認証 → /login へリダイレクト
  - 教員（role='teacher'）→ /teacher へリダイレクト
  - 受講者（role='student'）→ /student へリダイレクト
  - 読み込み中 → "読み込み中..." ローディング表示

6.2. 教員用画面（Teacher Role）

[G-T01] 教員ダッシュボード
- **URL**: /teacher
- **コンポーネント**: Dashboard.jsx
- **UI 要素**:
  - **統計ウィジェット**:
    - テストセットごとの総回答数（折れ線グラフ + バーチャート）
    - 合格率/不合格率（円グラフ）
    - コース進捗率（バーチャート）
  - **ランキング**:
    - 正解率が高い受講者 Top 5
    - 正解率が低い受講者 Top 5（要注意学生）
  - **スキル分析**: 習熟度が低いスキル Top 5（バーチャート）
- **主なアクション**:
  - **フィルタリング**: テストセット ID、年月、コース ID による表示データの絞り込み。
  - **手動更新**: データの再集計トリガー
    （Cloud Functions callAggregateDashboardMetrics 実行）。
  - **詳細トグル**: 各ウィジェットの拡大/縮小表示切り替え。

[G-T02] コース管理一覧
- **URL**: /teacher/course-management
- **コンポーネント**: CourseManagementPage.jsx
- **UI 要素**:
  - コース一覧テーブル（タイトル、スキル基準、作成日、公開/非公開、進捗）
  - [インポート/エクスポート] ボタン（JSON 形式）
  - [+新規コースを作成] ボタン
- **主なアクション**:
  - **一覧表示**: Firestore courses コレクションから一覧を取得・表示。
  - **削除**: コースの物理削除。
  - **進捗確認**: コンテキストメニューから進捗詳細画面へ遷移。

[G-T03] コース編集・作成
- **URL**: /teacher/course-management/edit/:id / new
- **コンポーネント**: CourseEditPage.jsx
- **UI 要素**:
  - **一般設定フォーム**:
    - コース名（必須）、評価基準選択、概要
    - サムネイル画像アップロード（プレビュー付き）
    - 公開設定（公開/非公開ラジオボタン）
  - **活動リスト（Drag & Drop）**:
    - 活動ブロックの追加/削除/並べ替え（react-beautiful-dnd）
    - 各ブロック: タイトル、種別(教材/テスト)、コンテンツ選択、キーワード
- **主なアクション**:
  - **保存**: Firestore courses コレクションへの保存。
    サムネイルがある場合は Storage へアップロード。

[G-T04] 教材管理
- **URL**: /teacher/resources
- **コンポーネント**: ResourceManagementPage.jsx
- **UI 要素**:
  - 教材一覧テーブル（タイトル、種類、作成日時）
  - [+新規教材を追加] ボタン（モーダル起動）
  - 編集モーダル: タイトル入力、ファイルアップロード（Drag & Drop or Select）
  - ファイル種別自動判定（PDF, Word, Video 等）

- **主なアクション**:
  - **アップロード**: Firebase Storage へファイルをアップロードし、メタデータを Firestore resources に保存。
  - **編集/削除**: タイトル変更、ファイルの差し替え、物理削除。

[G-T05] 学習進捗管理（一覧）
- **URL**: /teacher/progress
- **コンポーネント**: ProgressManagementPage.jsx
- **UI 要素**:
  - コース別進捗サマリーテーブル（受講者数、平均進捗率）
  - [エクスポート（JSON)] ボタン
- **機能**:
  - コースごとの全受講者の進捗平均を計算して表示。

[G-T06] 学習進捗詳細（コース別）
- **URL**: /teacher/progress/:courseId
- **コンポーネント**: CourseProgressDetailPage.jsx
- **UI 要素**:
  - 受講者一覧テーブル
  - 受講者ごとの進捗バー（0-100%）
  - 未完了の活動リスト表示
- **機能**:
  - 特定コース内の各受講者の進捗状況を詳細表示。

[G-T07] インサイト分析
- **URL**: /teacher/analysis
- **コンポーネント**: AnalysisPage.jsx
- **機能**:
  - 特定のテストを選択し、誤答率の高い問題を抽出。
  - 関連する講義資料と突き合わせ、AI (Gemini) に改善提案を生成させる。
  - 結果は Markdown 形式で表示され、クリップボードにコピー可能。

[G-T08] 問題作成（各種）
- **URL**: /teacher/create / /teacher/edit/...
- **コンポーネント**: QuestionCreationPage.jsx
- **構成**: タブ切り替えによる複数モード対応
  i. **通常問題(手動)**: 多肢選択/穴埋め。スキル評価基準の選択 UI。
  ii. **プログラミング(手動)**: 記述式。テストケース入力、模範解答コード入力。
  iii. **AI 生成**: Gemini API を用いた自動生成。プロンプト入力と生成結果のプレビュー/修正。
- **主なアクション**:
  - **問題バンクへ保存**: 作成した問題を Firestore questionBank に保存。

[G-T09] 問題バンク
- **URL**: /teacher/manage/questions
- **コンポーネント**: QuestionBank.jsx
- **UI 要素**:
  - スキル評価基準ごとのアコーディオンリスト
  - 問題検索/フィルタリング
  - 問題一覧テーブル（スキル領域、レベル、タグ、問題タイプ）
  - 一括操作バー（削除、エクスポート）
  - インポート機能（JSON）
- **機能**:
  - 登録済み全問題の閲覧、編集画面への遷移、一括削除。

[G-T10] テストセット管理

- **URL**: /teacher/manage/test-sets
- **コンポーネント**: TestSetManagement.jsx
- **UI 要素**:
  - テストセット一覧テーブル
  - 編集モーダル:
    - 基本設定（タイトル、説明、制限時間、合格点、公開設定）
    - **紐付け設定**: スキル評価基準、講義資料（分析用）
    - **問題選択**: 基準に基づく問題をチェックボックスで選択可能（グルーピング表示）
- **主なアクション**:
  - **作成/更新**: 問題 ID のリストを含むテストセット定義を Firestore testSets に保存。

[G-T11] 受講者管理

- **URL**: /teacher/manage/students
- **コンポーネント**: StudentManagementPage.jsx
- **UI 要素**:
  - 受講者検索バー（名前/Email）
  - 受講者一覧テーブル
  - [+新規受講者を登録] ボタン（モーダル）
  - CSV エクスポート
- **アクション**:
  - **CRUD**: Cloud Functions (createUser, updateUser, deleteUser) を経由して Firebase Auth ユーザーを管理。

[G-T12] スキル管理

- **URL**: /teacher/manage/skills
- **コンポーネント**: SkillManagementPage.jsx
- **UI 要素**:
  - ルーブリック(評価基準)一覧テーブル
  - 編集フォーム（ルーブリックタイトル、評価項目: スキル領域/レベル別タグ定義）
  - インポート/エクスポート
- **機能**:
  - 問題やコースで使用されるスキル評価基準のマスターデータ管理。

[G-T13] テスト結果一覧

- **URL**: /teacher/results
- **コンポーネント**: TestResultsPage.jsx (List: TestResults.jsx)
- **UI 要素**:
  - 受講者ごとのアコーディオンリスト（展開すると受験履歴を表示）
  - 一覧項目: テスト名、スコア、合否、バッジ、スキルマップへのリンク
  - 削除ボタン
- **機能**:
  - 全受講者のテスト提出履歴を一覧表示。

[G-T14] 結果詳細

- **URL**: /teacher/results/:submissionId
- **コンポーネント**: TeacherResultDetailPage.jsx (Content: SubmissionDetail.jsx)
- **UI 要素**:
  - **サマリー**: スコア、合否、プログレスバー
  - **詳細リスト**: 各問題の正誤、モードに応じた詳細表示
    - プログラミング問題: コード Diff、テストケース実行結果ログ
    - **AI フィードバック生成機能**: 各問題に対して「AI 解説を作成」ボタン（Teacher 権限で実行可能）

6.3. 受講者用画面（Student Role）

[G-S01] 受講者ダッシュボード
- **URL**: /student
- **コンポーネント**: StudentDashboardContent.jsx
- **UI 要素**:
  - **学習状況カード**: 最新の受講コース名とプログレスバー、[コースを続ける] リンク。
  - **挑戦できるテスト**: 最新の公開テスト 3 件のリスト、[挑戦する] ボタン。
  - **最新のテスト結果**: 直近の回答スコアと合否判定、[詳細を見る] リンク。
  - **スキルサマリー**: スキル分析ページへのショートカット。
- **機能**:
  - ログインユーザーの最新データを Firestore からリアルタイム取得し、学習の継続を促す。

[G-S02] コース一覧
- **URL**: /student/courses
- **コンポーネント**: StudentCourseList.jsx
- **UI 要素**:
  - 公開済みコースのカードグリッド表示
  - コースサムネイル、タイトル、説明文
  - [学習を始める] バッジ
- **機能**:
  - 公開フラグ(isPublished: true)が立っている全コースを表示。クリックでコース詳細へ遷移。

[G-S03] コース受講画面
- **URL**: /student/courses/:courseId
- **コンポーネント**: StudentCourseViewerPage.jsx
- **UI 要素**:
  - 活動(Activity)リスト
  - 各活動のステータス表示（完了済み/未完了）
  - コンテンツ種別アイコン（PDF/動画/テスト）
  - アクションボタン: [閲覧する] / [受験する] / [もう一度見る]
- **処理**:
  - 初回アクセス時に enrollments コレクションに受講データを自動作成。
  - 活動をクリックすると進捗（completedBlocks, progress）を更新し、対象コンテンツへ遷移。

[G-S04] 教材閲覧
- **URL**: /student/resource/:resourceId
- **コンポーネント**: ResourceViewerPage.jsx
- **UI 要素**:
  - **PDF の場合**: iframe による埋め込み表示。
  - **動画の場合**: HTML5 Video Player + 再生速度変更プルダウン（0.5x ～ 3.0x）。
- **機能**:
  - 指定された教材コンテンツを表示する。

[G-S05] テスト一覧（課題リスト）
- **URL**: /student/tests
- **コンポーネント**: AssignmentList.jsx
- **UI 要素**:
  - 公開テスト一覧テーブル
  - ステータス表示: 未解答 / 合格 / 不合格
  - アクションボタン: [挑戦する] / [再挑戦する] / [結果を見る]
- **機能**:
  - 受験可能なテストと、自身の過去の受験状況を一覧表示。

[G-S06] テスト受験画面

- **URL**: /test/:testId
- **コンポーネント**: TestPage.jsx
- **備考**: AuthGuard で保護されているが、Header 等は非表示の独立画面（FullScreen 推奨）。
- **機能**: 問題回答、制限時間カウントダウン、回答送信（submitTest 呼び出し）。
- **ステート**:
  - **読み込み中**: スピナー表示
  - **回答中**:
    - 問題リスト（全問表示）
    - 各問題タイプに応じた入力 UI（多肢選択/穴埋め/コードエディタ）
    - 「テストを提出する」ボタン
  - **提出中**: 全画面オーバーレイ。プログレスバーと進捗率(%)を表示。
  - **結果表示**: スコア(%)、合否判定(🎉合格/不合格)、ダッシュボードへ戻るボタン。

[G-S07] テスト結果一覧

- **URL**: /student/results
- **コンポーネント**: StudentResultsPage.jsx
- **UI 要素**:
  - 受験履歴テーブル（テスト名、スコア、合否、日時）
  - [結果を見る] リンク
- **機能**:
  - 過去の全ての受験結果を表示。

[G-S08] 結果詳細

- **URL**: /student/results/:submissionId
- **コンポーネント**: StudentResultDetailPage.jsx (Content: SubmissionDetail.jsx)
- **UI 要素**:
  - スコアサマリー、合否判定
  - 正誤一覧（合格時または設定により正解/解説を表示）
  - AI フィードバック（各問題ごとの解説）
- **制御**: viewerRole="student" としてコンポーネントを利用し、教員用とは異なる表示制御を行う（例: 不合格時は正解を隠すなど）。

[G-S09] スキル分析

- **URL**: /student/skills
- **コンポーネント**: StudentSkillsPage.jsx (Content: SkillsDashboard.jsx)
- **UI 要素**:
  - レーダーチャート: スキル領域ごとの習熟度
  - スキル詳細リスト: 各評価基準ごとのスコア
- **機能**:
  - 自身の全テスト結果に基づいたスキル習熟度を可視化。

---

7. 項目ラベル名一覧

7.1. ログイン画面

項目名	必須	タイプ	初期値/備考
メールアドレス	必須	Email	placeholder: email@example.com
パスワード	必須	Password	placeholder: Password

7.2. コース編集画面（/teacher/course-management）

基本設定

項目名	必須	タイプ	備考
コース名	必須	Text	
紐づけるスキル評価基準	任意	Select	登録済みのスキル基準から選択
概要	任意	Textarea	コースの説明文
サムネイル画像	任意	File	画像ファイル（アップロード時にプレビュー表示）
公開設定	必須	Radio	公開 / 非公開 (デフォルト: 公開)

活動ブロック（リスト項目）

項目名	必須	タイプ	備考
活動タイトル	任意	Text	デフォルト: "新しい活動"
種別	必須	Select	教材（Resource）/ テスト（Test）
コンテンツを選択	必須	Select	種別に応じた登録済みコンテンツを選択
キーワード	任意	Text	カンマ区切りで入力

7.3. 問題作成画面（/teacher/create）

共通項目

項目名	必須	タイプ	備考
問題文	必須	Textarea	Markdown 対応
スキル評価基準	任意	Select	
スキル領域	任意	Select	選択した基準に基づく領域
難易度	任意	Select	初級 / 中級 / 上級
タグ	任意	Text	自由入力

モード別項目

モード	項目名	必須	備考
多肢選択	選択肢	必須	2 つ以上追加。正解の選択肢にチェックを入れる。
穴埋め	正解キーワード	必須	完全一致で判定
プログラミング	模範解答コード	必須	Python コード
プログラミング	テストケース	必須	入力(stdin)と期待される出力(stdout)のペア

7.4. テストセット管理画面（/teacher/manage/test-sets）

項目名	必須	タイプ	備考
タイトル	必須	Text	テストセットの名称
説明	任意	Textarea	受講者に表示される説明
制限時間	必須	Number	分単位（デフォルト: 60）
合格点	必須	Number	0-100 点（デフォルト: 60）
公開設定	任意	Checkbox	チェックで公開
紐づけるスキル評価基準	任意	Select	分析に使用
関連講義資料	任意	Select	分析・復習提案に使用
問題選択	必須	Checkbox List	スキル領域ごとにグループ化された問題リスト

7.5. 教材管理画面（/teacher/resources）

項目名	必須	タイプ	備考
タイトル	任意	Text	未入力時はファイル名が使用される場合あり
ファイル	必須	File	PDF, MP4, Office ファイル等

7.6. 受講者管理画面（/teacher/manage/students）

項目名	必須	タイプ	備考
氏名	必須	Text	表示名
メールアドレス	必須	Email	ログイン ID として使用
パスワード	必須	Password	新規登録時のみ必須（編集時は空欄で変更なし）

---

8. エラーメッセージ

8.1. 認証・権限エラー

メッセージ例	発生条件	対処法
メールアドレスまたはパスワードが間違っています。	ログイン時に誤った認証情報を入力	正しいメールアドレスとパスワードを確認して再入力してください。
この操作を行うには認証が必要です。 / Authentication is required.	未ログイン状態で保護された操作を実行	再度ログインしてください。
この操作は先生（teacher）のみ許可されています。	受講者アカウントで教員用機能にアクセス	教員アカウントでログインし直してください。

8.2. 入力バリデーションエラー

メッセージ例	発生条件	対処法
コース名は必須項目です。	コース作成・編集時にタイトルが空	コース名を入力してください。
メールアドレス、パスワード、表示名は必須です。	受講者登録時に必須項目が不足	全ての必須項目を入力してください。
ファイルをアップロードしてください。	教材保存時にファイルが未選択	アップロードするファイルを選択してください。

8.3. AI 生成・外部連携エラー

メッセージ例	発生条件	対処法
AI response did not contain a valid JSON array.	Gemini が不正な形式のデータを返却	一時的な AI の不調です。数分待って再試行してください。
検証サーバーへの接続中にエラーが発生しました。	Executor Service (Python 実行環境）が応答しない	システム管理者に連絡するか、しばらく待って再試行してください。
AI フィードバックの生成に失敗しました。	Gemini API のエラーまたはタイムアウト	ネットワーク状況を確認し、再試行してください。
Gemini API キーが設定されていません。	サーバー側の環境変数設定ミス	システム管理者に連絡してください。

8.4. システム・通信エラー

メッセージ例	発生条件	対処法
コースの保存に失敗しました。	Firestore への書き込み失敗（オフライン等）	ネットワーク接続を確認し、再試行してください。

受講者リストの取得に失敗しました。	Firestore からの読み込み失敗	ページをリロードするか、ネットワークを確認してください。
サーバー内部で致命的なエラーが発生しました。	予期せぬバックエンドエラー	コンソールログを保存し、開発者/管理者に報告してください。

---

9. ファイル一覧

9.1. 入力ファイル（ユーザーアップロード）

用途	ファイル種別	形式	保存先（Storage）	関連画面
コースサムネイル	画像	JPG, PNG, etc.	course_thumbnails/{uuid}_{filename}	コース編集画面
講義資料	ドキュメント/動画	PDF, MP4, Office, CSV, etc.	resources/{timestamp}_{filename}	教材管理画面
評価基準インポート	データバックアップ	JSON	N/A (クライアントサイドで解析・DB 登録)	スキル管理画面
問題データインポート	データバックアップ	JSON	N/A (クライアントサイドで解析・DB 登録)	問題バンク画面

9.2. 出力ファイル（エクスポート/ダウンロード）

用途	形式	生成方法	トリガー	関連画面
評価基準エクスポート	JSON	クライアント生成（data:text/json）	[エクスポート] ボタン	スキル管理画面
問題データエクスポート	JSON	クライアント生成（data:text/json）	[全件エクスポート] ボタン	問題バンク画面
コース進捗データ	JSON	クライアント生成（data:text/json）	[エクスポート（JSON）] ボタン	進捗管理画面
受講者リスト	CSV	react-csv ライブラリ	[エクスポート（CSV）] ボタン	受講者管理画面

---

10. 外部インターフェース仕様（API）

10.1. Cloud Functions API (Frontend -> Backend)

[F-01] generateTest

- **概要**: Gemini を使用してテスト問題を生成する。
- **認証**: 必須
- **リクエスト**:
- {
- "topic": "Python の基礎",
- "numQuestions": 3,
- "questionType": "programming" | "multiple-choice" | "fill-in-the-blank",
- "language": "python",
- "functionName": "calculate_sum" // プログラミング問題のみ
- }
- **レスポンス**:
- {
- "questions": [
- {
- "type": "programming",

- "text": "...",
- "initialCode": "...",
- "solutionCode": "...",
- "testCases": [...]
- }
-]
- }

[F-02] submitTest

- **概要**: テストの回答を提出し、採点を行う。
- **認証**: 必須
- **リクエスト**:
- {
- "testId": "test_set_document_id",
- "answers": {
- "question_id_1": "user_answer_1",
- "question_id_2": 2 // 選択肢インデックス
- }
- }
- **レスポンス**:
- {
- "submissionId": "submission_document_id",
- "score": 85,
- "passed": true
- }

[F-03] getAIFeedback

- **概要**: 不正解だったプログラミング問題に対して AI フィードバックを取得する。
- **認証**: 必須
- **リクエスト**:
- {
- "studentCode": "...",
- "questionText": "...",
- "solutionCode": "...",
- "testCases": [{ "callCode": "...", "expectedOutput": "..." }] // 失敗したケース
- }
- **レスポンス**:
- {
- "feedback": "Markdown 形式のフィードバックテキスト..."
- }

[F-04] suggestRecipe (Recipe Suggester)

- **概要**: 指定された食材に基づいてレシピを提案する。
- **認証**: 必須
- **リクエスト**:
- {
- "requiredIngredients": [{ "name": "鶏肉", "amount": "300g" }],
- "optionalIngredients": [{ "name": "玉ねぎ", "amount": "1 個" }]
- }
- **レスポンス**:
- {
- "recipes": [
- {
- "name": "鶏肉のトマト煮込み",
- "ingredients": [...],

- "instructions": ["..."],
- "nutrition_estimate_per_2_servings": {...},
- "image_prompt": "..."
- }
-]
- }

10.2. Executor Service API (Cloud Functions -> Cloud Run)

- **概要**: 信頼されていない Python コードをサンドボックス環境で実行する。
- **エンドポイント**: POST /execute
- **リクエスト**:
- {
- "language": "python",
- "code": "def add(a, b): return a + b",
- "testCases": [
- { "setupCode": "", "callCode": "add(1, 2)", "expectedOutput": "3" }
-]
- }
- **レスポンス**:
- {
- "passedCount": 1,
- "totalCount": 1,
- "results": [
- { "status": "Passed", "output": "3", "expected": "3" }
-]
- }

---

11. バッチ処理一覧

処理名称	トリガー/スケジュール	実行内容	関連コレクション
ダッシュボード集計 (aggregateMetricsScheduled)	**毎日 0:00 (JST)** 0 0 * * *	全期間の提出データとユーザー情報を集計し、以下の指標を更新する。 1. **日次提出数**: dailySubmissions 2. **スキル習熟度**: lowProficiencySkills (正答率の低い順) 3. **受講者ランキング**: studentRanks (テストセット別スコア順) 4. **テスト合格率**: passFailRate	testSubmissions users questionBank dashboard_metrics

11.1. 補足: スケジュール実行の仕組み

- **Firebase Google Cloud Scheduler** を利用。
- タイムゾーンは Asia/Tokyo に設定。
- 手動実行も可能（教員用ダッシュボードの「集計実行」ボタン -> callAggregateDashboardMetrics)。

---

12. アプリケーション仕様

12.1. 動作推奨環境

項目	内容
PC ブラウザ	Google Chrome (最新版), Microsoft Edge (最新版), Firefox (最新版), Safari (最新版)
SP/タブレット	レスポンシブ対応（Tailwind CSS により実装）
画面解像度	PC: 1280px 以上推奨（管理画面), SP: 375px 以上

12.2. 言語・タイムゾーン・国際化（i18n）

項目	仕様
対応言語	日本語（JA）のみ（ハードコード実装）
タイムゾーン	Asia/Tokyo (JST)
日時フォーマット	YYYY/MM/DD または YYYY-MM-DD 形式を標準使用
日時処理ライブラリ	date-fns v3.6.0

12.3. 環境変数・構成値（Configuration）

環境変数は .env ファイルを使用せず、以下の方式で管理する。

分類	管理場所	内容・例
Frontend Config	frontend/src/firebase.js	Firebase SDK 初期化情報（API Key, AuthDomain, ProjectID 等）
Backend Config	functions.config()	gemini.key (Gemini API キー), executor.url (コード実行基盤 URL)

12.4. 依存ライブラリ（Dependencies）

主要な依存パッケージとそのバージョン

カテゴリ	パッケージ名	バージョン	用途
Core (Frontend)	react	^18.3.1	UI ライブラリ
	vite	^5.3.4	ビルドツール
	react-router-dom	^6.25.1	ルーティング
Firebase	firebase	^10.12.3	Firebase SDK
	react-firebase-hooks	^5.1.1	React Hooks for Firebase
UI/UX	tailwindcss	^3.4.6	CSS フレームワーク
Editor	@uiw/react-codemirror	^4.23.0	コードエディタ
Chart	recharts	^3.3.0	グラフ描画（ダッシュボード）

以　上

文部科学省委託 専修学校による地域産業中核的人材養成事業――外人口減少地域の職業人材を確保するための専修学校振興プログラム

人口減少地域を支えるAI補完型ITエンジニア養成プログラムの開発

事業報告書 令和７年度

発行日――2026年３月１日　初版第１刷発行

発行者――学校法人電子学園 日本電子専門学校

〒169-8852 東京都新宿区百人町1-25-4
TEL 03-3363-7761　　URL https://www.jec.ac.jp/

本報告書は、文部科学省の教育政策推進事業委託費による委託事業として、
学校法人電子学園 日本電子専門学校が実施した令和７年度「専修学校の国際化推進事業」の成果を取り纏めたものです。