

Gemini×学力向上 AIで振り返り問題生成

◆どのような際に活用できるか

▷授業後に行っている振り返りのレベルに応じた問題を自動で生成することで、自身の準備時間を削減したいとき

◆この方法で見込める効果

▷レベルに応じた学習に取り組ませることで、学力を向上させることができる。
▷授業後のプリントの準備時間の大幅な削減ができる。

◆使用までの手引き（画像付き）

①Geminiを立ち上げる



②左上から2.5Proを選択する



③以下の文章をコピーペーストし、Canvas機能を使う

```
import React, { useState, useRef, useEffect } from 'react';
import { initializeApp } from 'firebase/app';
import { getAuth, signInAnonymously, onAuthStateChanged, signInWithCustomToken } from 'firebase/auth';
import { getFirestore } from 'firebase/firestore';

// --- Firebase Initialization ---
const appId = typeof __app_id !== 'undefined' ? __app_id : 'default-app-id';
const firebaseConfig = typeof __firebase_config !== 'undefined' ? JSON.parse(__firebase_config) : {};
let app, auth, db;

try {
  app = initializeApp(firebaseConfig);
  auth = getAuth(app);
  db = getFirestore(app);
} catch (e) {
  console.error("Firebase initialization error:", e);
}

// --- Main App Component ---
const App = () => {
  // --- State Hooks ---
  const [subject, setSubject] = useState('社会');
  const [gradeLevel, setGradeLevel] = useState('中学1年生');
  const [summary, setSummary] = useState('');
  const [goal, setGoal] = useState('');
  const [materialFiles, setMaterialFiles] = useState([]);
  const [referenceFiles, setReferenceFiles] = useState([]);
  const [page, setPage] = useState('input'); // 'input' or 'preview'
  const [loading, setLoading] = useState(false);
  const [message, setMessage] = useState('');
  const [fullReviewContent, setFullReviewContent] = useState(''); // Stores Markdown content
  const [userId, setUserId] = useState(null);
  const [isPdfsLoaded, setIsPdfsLoaded] = useState(false);
  const [isMarkedLoaded, setIsMarkedLoaded] = useState(false);

  // --- Refs for file inputs ---
  const materialFileInputRef = useRef(null);
  const referenceFileInputRef = useRef(null);

  // --- useEffect to load external libraries ---
  useEffect(() => {
    const loadScript = (src, onLoad, onError) => {
      const script = document.createElement('script');
      script.src = src;
      script.onload = onLoad;
      script.onerror = onError;
      document.body.appendChild(script);
      return script;
    };

    const pdfScript = loadScript(
      'https://cdnjs.cloudflare.com/ajax/libs/pdf.js/2.16.105/pdf.min.js',
      () => {
        window.pdfjsLib.GlobalWorkerOptions.workerSrc = `https://cdnjs.cloudflare.com/ajax/libs/pdf.js/2.16.105/pdf.worker.min.js`;
        setIsPdfsLoaded(true);
      },
      () => setMessage("エラー: PDF読み込みライブラリの読み込みに失敗しました。")
    );

    const markedScript = loadScript(
      'https://cdn.jsdelivr.net/npm/marked/marked.min.js',
      () => setIsMarkedLoaded(true),
      () => setMessage("エラー: 表示用ライブラリの読み込みに失敗しました。")
    );

    return () => {
      document.body.removeChild(pdfScript);
      document.body.removeChild(markedScript);
    };
  }, []);

  // --- useEffect for Firebase Auth ---
  useEffect(() => {
    if (!auth) return;
    const initAuth = async () => {
      try {
        const token = typeof __initial_auth_token !== 'undefined' ? __initial_auth_token : null;
        if (token) {
          await signInWithCustomToken(auth, token);
        } else {
          await signInAnonymously(auth);
        }
      } catch (error) {
        console.error("Firebase auth error:", error);
      }
    };
    initAuth();
    const unsubscribe = onAuthStateChanged(auth, (user) => user ? setUserId(user.uid) : setUserId(null));
    return () => unsubscribe();
  }, []);

  // --- PDF Text Extraction Function ---
  const extractTextFromPdf = async (file) => {
    if (!isPdfsLoaded) throw new Error("pdf.jsがまだ読み込まれていません。");
    const arrayBuffer = await file.arrayBuffer();
    const pdf = await window.pdfjsLib.getDocument(arrayBuffer).promise;
    let fullText = '';
    for (let i = 1; i <= pdf.numPages; i++) {
      const page = await pdf.getPage(i);
      const textContent = await page.getTextContent();
      fullText += textContent.items.map(item => item.str).join(' ') + '\n\n';
    }
    return fullText;
  };

  // --- API Call Function ---
  const callGeminiApi = async (prompt) => {
    let attempts = 0;
    const maxAttempts = 3;
    const apiKey = ""; // Canvas環境から自動的に提供されることを期待
  }
}
```

[illegible]

```
<div className="text-center"><h1 className="text-3xl font-bold text-gray-800">問題自動作成</h1><p className="text-md text-gray-600 mt-2">AIが授業内容に合わせた問題プリントを生成します。</p></div>

<div>
  <label htmlFor="subject" className="block text-sm font-medium text-gray-700">教科</label>
  <select id="subject" className="mt-1 block w-full pl-3 pr-10 py-2 text-base border-gray-300 focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm rounded-md"
    value={subject} onChange={(e) => setSubject(e.target.value)}>
    {(['国語', '数学', '理科', '社会', '英語', '音楽', '体育', '保健', '美術', '技術', '家庭科'].map((sub) => (<option key={sub} value={sub}>{sub}</option>)))}
  </select>
</div>

<div>
  <label htmlFor="gradeLevel" className="block text-sm font-medium text-gray-700">対象学年</label>
  <select id="gradeLevel" className="mt-1 block w-full pl-3 pr-10 py-2 text-base border-gray-300 focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm rounded-md"
    value={gradeLevel} onChange={(e) => setGradeLevel(e.target.value)}>
    {(['小学1年生', '小学2年生', '小学3年生', '小学4年生', '小学5年生', '小学6年生', '中学1年生', '中学2年生', '中学3年生', '高校1年生', '高校2年生', '高校3年生'].map((grade) => (<option key={grade}
value={grade}>{grade}</option>)))}
  </select>
</div>

<div>
  <label htmlFor="summary" className="block text-sm font-medium text-gray-700">授業の概要</label>
  <textarea id="summary" rows="3" className="mt-1 block w-full p-2 border border-gray-300 rounded-md shadow-sm" value={summary} onChange={(e) => setSummary(e.target.value)}
placeholder="例：日本の歴史、江戸時代から明治時代にかけての社会の変化について学んだ。"/>
</div>

<div>
  <label htmlFor="goal" className="block text-sm font-medium text-gray-700">授業のめあて</label>
  <textarea id="goal" rows="3" className="mt-1 block w-full p-2 border border-gray-300 rounded-md shadow-sm" value={goal} onChange={(e) => setGoal(e.target.value)} placeholder="例：江戸時代と明治時代の社会の具体的な違いを3つ説明できるようにする。"/>
</div>

<FileUploadArea title="授業資料 (PDF)" fileInputRef={materialFileInputRef} onFileChange={handleMaterialFileChange} onUploadClick={handleMaterialUploadClick} files={materialFiles}/>
<FileUploadArea title="参考にする問題 (PDF)" fileInputRef={referenceFileInputRef} onFileChange={handleReferenceFileChange} onUploadClick={handleReferenceUploadClick}
files={referenceFiles}/>
<div className="pt-4">
  <button onClick={createReviewContent} className="w-full inline-flex items-center justify-center px-6 py-3 border border-transparent text-base font-medium rounded-md shadow-sm text-white bg-indigo-600 hover:bg-indigo-700 disabled:bg-indigo-400 disabled:opacity-50">
    {loading ? '生成中...' : '問題プリントを作成'}
  </button>
</div>
{message && (<div className="mt-4 p-3 text-center text-sm rounded-md ${message.includes('エラー')} ? 'bg-red-100 text-red-700' : 'bg-blue-100 text-blue-700'}>{message}</div>)}
</div>
);

const renderPreviewPage = () => {
  <div className="p-6 max-w-4xl mx-auto bg-white rounded-xl shadow-lg space-y-6">
    <div className="text-center"><h1 className="text-3xl font-bold text-gray-800">プレビューとダウンロード</h1></div>
    {fullReviewContent && isMarkedLoaded && <div className="mt-6 border-2 border-gray-200 bg-gray-50 rounded-lg p-6 h-[32rem] overflow-y-auto prose max-w-none" dangerouslySetInnerHTML={{
__html: window.marked.parse(fullReviewContent) }}></div>
    <div className="flex flex-col sm:flex-row justify-between items-center mt-6 gap-4">
      <button onClick={() => { setPage('input'); setFullReviewContent(''); setMessage(''); }} className="w-full sm:w-auto px-4 py-2 border border-gray-300 rounded-md shadow-sm text-sm font-medium text-gray-700 bg-white hover:bg-gray-50">戻って修正</button>
    </div>
    {fullReviewContent && {
      <div className="flex gap-2 w-full sm:w-auto">
        <button onClick={() => handleDownload(fullReviewContent, 'txt')} className="flex-1 justify-center px-4 py-2 border border-transparent rounded-md shadow-sm text-sm font-medium text-white bg-green-600 hover:bg-green-700">TXT</button>
        <button onClick={() => handleDownload(fullReviewContent, 'html')} className="flex-1 justify-center px-4 py-2 border border-transparent rounded-md shadow-sm text-sm font-medium text-white bg-blue-600 hover:bg-blue-700">HTML</button>
      </div>
    )}
    </div>
    {message && (<div className="mt-4 p-3 text-center text-sm text-green-700 bg-green-100 rounded-md">{message}</div>)}
  </div>
);

return {
  <div className="bg-gray-50 min-h-screen py-10 px-4">
    <style>{`.prose h1,.prose h2,.prose h3{margin-top:1.5em;margin-bottom:.5em}.prose p{margin-bottom:1em}`}</style>
    {page === 'input' ? renderInputPage() : renderPreviewPage()}
  </div>
);
};

export default App;
```

コードに改変は加えず、そのまま複製してください

Gemini

2.5 Flash ▼

こんにちは、

```
<div className="bg-gray-50 min-h-screen py-10 px-4 sm:px-6 lg:px-8 font-sans">
  {page === 'input' ? renderInputPage() : renderPreviewPage()}
</div>
);
};

export default App;
```



ガイド付き学習



動画



Deep Research



Canvas



④送信ボタンを押すとアプリができる

Gemini

2.5 Flash ▼

```
import { useState, useRef }
from 'react';
import { initializeApp } fro...
```



はい、承知いたしました。学年を選択する機能を追加し、生成されるコンテンツに反映するようにコードを更新しました。

Gemini へのプロンプトを入力



ガイド付き学習



東京都板橋区教育委員会のチャットはモデルの改善には使用されません。Gemini は不正確な情報を表示することがあるため、生成された回答を再確認するようにしてください。プライバシーと Gemini

授業振り返りアプ...



コード

✓ プレビュー

共有

授業振り返りコンテンツ作成

授業内容を入力すると、AIが習熟度別の振り返りプリントを自動生成します。

教科

国語

対象学年

中学1年生

授業の概要

例：日本の歴史、江戸時代から明治時代にかけての社会の変化について学んだ。

⑤必要な情報を記入し、データがあれば追加して【振り返りプリントを作成】を押す

The screenshot shows a web application interface for creating a review print. On the left, there is a code editor with the following code:

```
import { useState, useRef }  
from 'react';  
import { initializeApp } from 'firebase/app';
```

Below the code editor, there is a chat area with a Gemini prompt: "い、承知いたしました。学年を選択する機を追加し、生成されるコンテンツに反映するようにコードを更新しました。" and a text input field with the placeholder "gemini へのプロンプトを入力".

On the right, there is a preview area with two sections: "授業資料 (PDF)" and "参考にする問題 (PDF)". Each section has a dashed box with a file icon and the text "ファイルを選択 またはドラッグ & ドロップ".

At the bottom left, there is a disclaimer: "橋区教育委員会 のチャットはモデルの改善には使用されませ mini は不正確な情報を表示することがあるため、生成された を再確認するようにしてください。プライバシーと Gemini".

⑥必要に応じてカスタマイズも可能です。ノーコードでの開発も可能です。