**Project Description**

The project, titled **"Overburdened Teachers and the Need for Personalized Feedback"**, aims to address the challenges faced by educators in providing individualized feedback to students in large classrooms. By leveraging modern AI technologies, the project automates assignment grading and generates personalized feedback, reducing the workload on teachers and ensuring that students receive timely and constructive guidance. The solution aligns with **UN SDG 4: Quality Education**, promoting inclusive and equitable learning opportunities.

**Updated Tech Stack**

**Frontend:**

* **Next.js**: For building a responsive and dynamic user interface with server-side rendering (SSR) for better scalability.
* **Tailwind CSS**: For creating a clean, modern, and accessible design.
* **React Hook Forms**: For handling assignment submissions efficiently.

**Backend:**

* **Next.js API Routes**: To manage the backend logic (grading, OCR processing, feedback generation).
* **Gemini API**: For generating personalized feedback based on assignment content.
* **Google Cloud Vision API**: For extracting text from handwritten assignments using Optical Character Recognition (OCR).

**Database:**

* **Firebase Firestore**: For real-time storage of student submissions, teacher answer keys, and feedback.

**Authentication:**

* **Firebase Authentication**: For managing teacher and student logins securely.

**Hosting & Deployment:**

* **Vercel**: For deploying the Next.js application with seamless CI/CD.

**Colour Scheme**

1. Primary Colors:

* Sky Blue:
  + Main accent color: text-sky-700
  + Background variations: bg-sky-50, bg-sky-900/80
  + Gradient backgrounds using sky blue tones

1. Neutral Colors:

* Grays:
  + text-gray-50, text-gray-600, text-gray-800
  + Used for backgrounds, text, and subtle design elements

1. Accent Colors:

* Amber:
  + bg-amber-500 for call-to-action button
  + Provides a warm contrast to the cool sky blues

**Workflow**

1. **Assignment Submission**:
   * Students upload assignments in the form of handwritten text or MCQs through the submission page.
   * For handwritten assignments, the **Google Cloud Vision API** extracts the text.
2. **Grading Process**:
   * For MCQs, the uploaded answers are compared against the teacher-provided answer key stored in **Firestore**.
   * For descriptive answers, the **Gemini API** is used to analyze the responses and grade them based on relevance and quality.
3. **Feedback Generation**:
   * Personalized feedback is generated using the **Gemini API**, tailored to the student’s performance.
   * Feedback highlights strengths, areas for improvement, and actionable suggestions.
4. **Results Dashboard**:
   * Students view their grades and detailed feedback on a dedicated results page.
   * Teachers can monitor overall class performance and download performance summaries.
5. **Real-Time Updates**:
   * Firebase Firestore ensures that both students and teachers receive real-time updates on submissions, grading, and feedback.

**Key Features**

* **Handwritten Assignment Support**: OCR technology ensures seamless grading of handwritten submissions.
* **Personalized Feedback**: AI-driven feedback improves the quality of learning.
* **Real-Time Dashboard**: Teachers and students can track assignments and feedback in real time.
* **Scalable and User-Friendly**: The platform can handle high student volumes and is designed for easy navigation.

This solution streamlines grading and feedback, empowering teachers to focus on teaching and students to excel academically.