

Steps to Build the "Sahayak" MCP Server & Client Using Google AI Technologies

Overview

This guide outlines the architecture and step-by-step approach to building an AI-powered teaching assistant ("Sahayak") for multi-grade, low-resource classrooms in India. The solution leverages only Google AI models and tools, with Gemini 2.0 Flash as the core LLM, and integrates Firebase Studio for rapid development and deployment.

1. Solution Architecture

Component	Technology/Model	Purpose
MCP Server	Cloud Run / GKE	Hosts APIs, orchestrates AI tasks
MCP Client	Web/mobile app via Firebase Studio	Teacher-facing interface, handles user input/output
Core AI Model	Gemini 2.0 Flash (Vertex AI)	Multimodal content generation, language, reasoning
Speech/Audio	Vertex AI Speech-to-Text, TTS	Audio assessments, text-to-speech for local language
Video Generation	Veo2 (Google)	Generate educational videos from audio or prompts
Storage	Cloud Storage, Firestore	Store materials, user data, logs
Integration	Google Docs, Forms APIs	Export worksheets/quizzes for HITL

2. Key Features & Model/Tool Mapping

Feature	Model/Tool Used
Hyper-local content generation	Gemini 2.0 Flash
Differentiated worksheets from images	Gemini 2.0 Flash (multimodal)
Instant knowledge base (Q&A)	Gemini 2.0 Flash
Visual aid generation (drawings/charts)	Gemini 2.0 Flash (image output)
Audio-based reading assessments	Vertex AI Speech-to-Text

Lesson planners, game generation	Gemini 2.0 Flash
Video generation for content	Veo2, Google TTS
Export to Docs/Forms	Google Workspace APIs

3. Step-by-Step Implementation

A. MCP Server (Backend)

1. Set Up Project in Firebase Studio

- Create a new full-stack project using Firebase Studio for rapid prototyping and seamless Google Cloud integration^{[123](#)}.

2. API Orchestration

- Use Cloud Run or GKE to deploy stateless REST/gRPC APIs that serve client requests.
- Integrate Vertex AI Gemini 2.0 Flash for all LLM and multimodal tasks (text, image, audio)^{[45](#)}.
- Implement endpoints for:
 - Content generation (text, images, charts)
 - Worksheet creation (from images)
 - Q&A (knowledge base)
 - Visual aid generation
 - Audio assessment (Speech-to-Text)
 - Video generation (Veo2 + TTS)
 - Export to Google Docs/Forms

3. Authentication and Security

- Use Firebase Authentication for secure user access.

4. Data Storage

- Store generated materials, user data, and logs in Firestore and Cloud Storage.

5. Monitoring & Logging

- Integrate Cloud Logging & Monitoring for observability.

B. MCP Client (Frontend)

1. Build UI in Firebase Studio

- Use the App Prototyping agent or code-based approach to create a responsive web/mobile app¹².
- Features for teachers:
 - Local language input (text/audio)
 - Upload images (textbook pages)
 - Receive/download generated content
 - View/print visual aids
 - Conduct audio assessments
 - Export materials to Docs/Forms

2. Multimodal Input Handling

- Allow teachers to input prompts in their local language (Gemini supports multilingual input/output)⁴⁶.
- Enable image uploads for worksheet generation.

3. Real-Time Feedback

- Use Gemini's fast response and multimodal output for instant content delivery⁴⁵.

4. Export/Integration

- Integrate with Google Docs and Forms APIs for exporting worksheets and quizzes for further editing and distribution.

C. Advanced Features

- **Video Generation:** Use Veo2 to create short educational clips; generate corresponding audio with Google TTS, then merge⁷.
- **Audio Assessments:** Use Vertex AI Speech-to-Text to analyze student reading and provide feedback.
- **Lesson Planning & Games:** Use Gemini 2.0 Flash to generate weekly lesson plans and simple educational games on demand.

4. Model & Tool Selection Rationale

- **Gemini 2.0 Flash:** Chosen for its superior multimodal capabilities, speed, and support for large context windows—ideal for rapid, diverse content generation in multiple languages and formats⁴⁵.
- **Vertex AI Speech-to-Text & TTS:** For accurate audio assessments and local language TTS output.
- **Veo2:** For video generation from prompts and audio, enhancing visual learning.
- **Firebase Studio:** Enables rapid prototyping, seamless Google Cloud integration, and easy deployment¹²³.
- **Google Workspace APIs:** For exporting content into familiar tools for teachers.

5. Deployment Workflow

1. Develop and Test in Firebase Studio

- Use built-in emulators for local testing².

2. Deploy Backend to Cloud Run

- One-click deployment from Firebase Studio to Cloud Run for scalable, managed hosting⁸.

3. Publish Client via Firebase Hosting/App Hosting

- Deploy the web/mobile app for teacher access.

4. Monitor and Iterate

- Use Cloud Logging & Monitoring for continuous improvement.

6. Example Workflow for a Teacher

1. Teacher opens the Sahayak app and selects "Create Worksheet."
2. Uploads a photo of a textbook page.
3. Selects target grade levels.
4. Gemini 2.0 Flash processes the image and generates differentiated worksheets.
5. Teacher exports the worksheet to Google Docs for review/editing.
6. Optionally, the teacher asks Sahayak to generate a simple chart, which is provided as a line drawing.
7. For reading assessments, the teacher records a student's reading; the app provides instant feedback via Speech-to-Text analysis.

7. Extending the Platform

- Add support for additional Indian languages as Gemini expands.
- Incorporate more advanced analytics and personalized recommendations using Firestore data.
- Enable offline-first capabilities for low-connectivity environments.