Sahayak.AI Project Documentation

1. Project Overview
    - **Project Title:** Sahayak.AI: The AI CO- teacher
    - **Description:** In many schools across India, especially in villages and small towns, there are not enough teachers. One teacher often must handle students from different classes and different age groups in the same room. This is very difficult because every child learns at a different speed, and the teacher has a little time to give personal attention. Teachers also struggle to prepare learning materials in local languages, or to make lessons interesting for all levels of students. As a result, some children fall behind, and teachers feel stressed and overworked. The problem is not just about teaching; it is about making sure every child gets equal chances to learn. The challenge is to create an AI assistant that can help teachers by reducing their workload, creating useful study materials, and giving personalized support to students.
    - **Problem Statement:** Teachers in under- resourced schools face difficulty providing personalized, engaging education to diverse groups of students due to limited time and resources. Manual preparation of multi-level, culturally relevant materials and individual student support is overwhelming. There is a need for an AI assistant that streamlines these tasks and ensures equitable learning opportunities for all children.
    - **Solution:** Sahayak.AI is not just another edtech tool, it's a true co-teacher designed to ease teacher workloads and make classrooms more engaging. Built for real classrooms where one teacher manages many students with diverse needs, it handles the heavy lifting: creating lesson plans, generating practice sheets, drawing visuals, and even producing personalized stories and illustrations. With just a few clicks or voice commands, it explains, it explains tough topics simply, answers student questions instantly, and adapts content to different learning speeds and styles.
      What makes this different is that it thinks like a partner, not just a tool. It transforms late- night preparations into quick, effortless tasks and brings joy into classrooms with interactive learning games that adapt to each child's level. Teachers gain time and peace of mind, while students learn by exploring, playing, and engaging. It isn't here to replace teachers, it's here to empower them, helping them focus on what they do best: inspiring young minds.

2. Features
    - *Story creator:* Instantly generates engaging, age-appropriate stories and examples that are culturally relevant to students, complete with unique AI- generated illustrations to capture their imagination.

- *Worksheet Generator:* Allows a teacher to simply upload a photo of any textbook page. The AI analyzes the content and automatically produces three distinct worksheets for easy, intermediate and advanced skill levels.
- *Knowledge assistant:* Acts as a real-time conversational AI. It answers complex student questions with simple, easy- to- understand explanations and analogies, and provides text- to- speech audio aid comprehension.
- *Visual Aid generator:* creates clear, simple, black board friendly diagrams and sketches from text prompts, helping teachers explain abstract concepts visually without needing artistic skill.
- *Auto lesson planner:* Transforms a weekly syllabus or list of topics into a detailed, time-structured lesson plan, complete with learning objectives, classroom activities, and assessment methods.
- *Learning games arcade:* A suite of fun, interactive learning games including Math quiz, word scramble, memory match with adaptive difficulty to make learning engaging and reinforce key concepts.
- *Role play script creator:* Creates educational role-paly scripts, complete with multi-speaker audio, to facilitate interactive classroom activities.
- *Ask later asynchronous queue:* Enables teachers to capture student questions during class and receive comprehensive, multi- modal answers (text, image, and audio) later.

3. Tech stack

Frontend:

- Next.js & React: Utilized the App Router for server-rendered pages, ensuring optimal performance and a fast user experience.
- TypeScript: Ensured type safety and maintainability across the project.
- Tailwind CSS & ShadCN UI: Created a professional, responsive, and accessible user interface with a modern design system.

Backend & AI Integration:

- Firebase Genkit: Orchestrated all backend AI logic. Genkit was used to define complex, multi-step "flows" that chain together calls to various AI models.
- Google AI (Gemini Models):

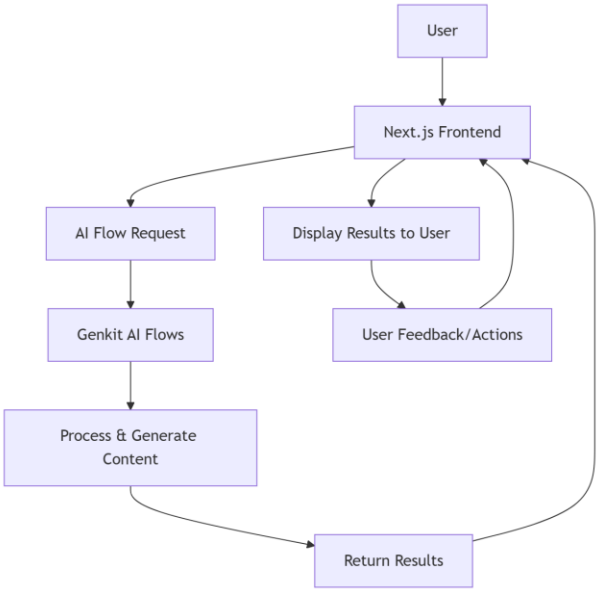
Gemini 1.5 Flash: For all text generation, summarization, and conversational AI tasks.
Gemini Pro Vision: To analyze the content of uploaded textbook images.
Gemini Image & Audio Models: For generating visual aids, illustrations for stories, and text-to-speech functionality.

Deployment:

- Vercel: The application is hosted on Vercel, which provides a seamless environment for deploying Next.js applications, including serverless functions for the backend logic.

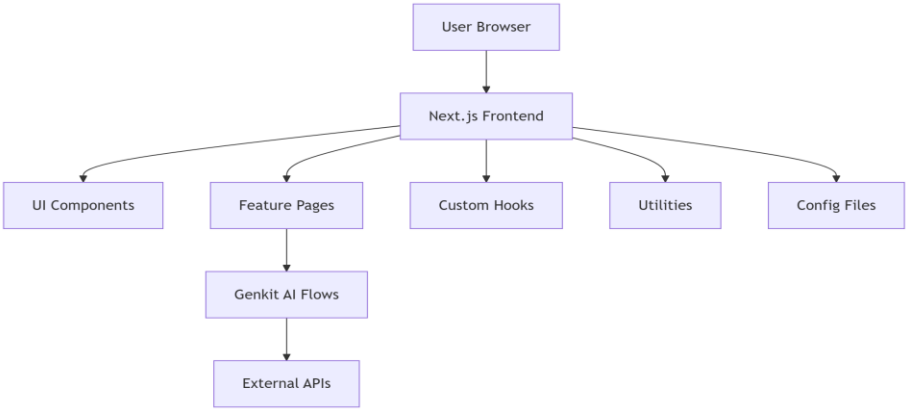
4. Workflow Diagram

   Shows how user requests flow through the frontend, trigger AI processes, and return results for user interaction.



5. Architecture Diagram

   Illustrates the main components and their connections in the app, from user browser to frontend, AI flows, utilities, and external.



6. Repository Structure

```
├── .gitignore
├── .idx/
├── .vscode/
├── Project_idea.md
├── apphosting.yaml
├── components.json
├── next.config.ts
├── package-lock.json
├── package.json
├── postcss.config.mjs
├── src/
├── tailwind.config.ts
├── tsconfig.json
├── venv/
```

src/: Source code for the application.
venv/: Python virtual environment.
.vscode/, .idx/: Editor and indexing configs.
Project_idea.md: Project concept and planning notes.
components.json: UI/component definitions.
apphosting.yaml: Cloud deployment configuration.
next.config.ts, tailwind.config.ts, postcss.config.mjs: Frontend build and styling configs.

7. Perquisites

- Node.js & npm
- Python (for backend/AI tasks)
- Vercel CLI (optional for deployment)

8. Installation

- *Clone the repository*

```
git clone https://github.com/HaarthiV13/Sahayak_AI.git
cd Sahayak_AI
```

- *Install Node.js dependencies*

```
npm install
```

- *Set up Python environment*

```
python3 -m venv venv
source venv/bin/activate
```

Install Python dependencies as needed

Running Locally

- *Start frontend (Next.js)*

npm run dev

- *Run backend Python scripts(optional)*

python src/main.py

9. Live and Video Demo

Link for live demo: https://sahayak-ai-flax.vercel.app/

Link for video demo: https://drive.google.com/file/d/1wu_saMlWdvXc6e-Oju-TayDv3zUvPz5R/view?usp=sharing