

Smart Exam Assistance for Disabled Students

Project Documentation

1. Introduction

The 'Smart Exam Assistance for Disabled Students' (Accessible Exam Tool) is a web-based examination system designed to empower visually impaired (blind) and hearing-impaired (deaf) students to take exams independently. Traditionally, these students rely on human scribes, which compromises their privacy and independence. This project leverages AI, Computer Vision, and Web Speech APIs to create a seamless, accessible exam environment.

2. Problem Statement

Visually impaired students face significant barriers during written examinations, typically depending on scribes to read questions and write answers. This process is slow, prone to errors, and lacks privacy. Similarly, hearing-impaired students struggle with verbal instructions and communication during exams. There is a critical need for a technology-driven solution that allows these students to demonstrate their knowledge independently.

3. Technology Stack

- Backend: Python (Flask Framework)
- Frontend: HTML5, CSS3, Vanilla JavaScript
- Computer Vision & AI: MediaPipe (Hand Tracking), OpenCV, TensorFlow/Keras (Model Training)
- OCR & Text Extraction: Tesseract OCR (pytesseract), Pillow (PIL), PyPDF
- Speech Services: Web Speech API (SpeechSynthesis & SpeechRecognition)
- Data Storage: JSON-based file storage (Scalable to SQL databases)

4. Key Features

A. Blind Student Mode (Voice-Based Exam)

- Text-to-Speech (TTS): Questions are read aloud automatically.
- Speech-to-Text (STT): Students answer questions using voice commands.
- Keyboard Navigation: Full keyboard support (e.g., 'A' to Answer, 'N' for Next) eliminates mouse usage.
- Audio Feedback: Confirmation messages for actions like submitting an answer.

B. Deaf Student Mode (Sign Language Exam)

- Doppler-Correction / Sign Language Recognition: Uses MediaPipe to detect hand gestures.
- Real-time Processing: Converts sign language gestures into text answers.
- Visual Instructions: All exam instructions are displayed clearly as text.
- Gesture-Based Evaluation: Custom trained models recognize specific answer gestures (e.g., A/B/C/D for MC

Smart Exam Assistance for Disabled Students

Project Documentation

C. Teacher Dashboard

- Question Paper Upload: Supports PDF and Image formats.
- Automated Extraction: Uses OCR to extract text from images and PDFs.
- Exam Management: Create, view, and delete exam papers.

5. System Architecture & Workflow

1. Teacher Upload: The teacher uploads a question paper (PDF/Image) via the dashboard.
2. Processing: The system extracts text using pytesseract (OCR) or PyPDF.
3. Exam Generation: Questions are parsed and stored in structured JSON files.
4. Student Access: Students log in and select their mode (Blind or Deaf).
5. Exam Interface:
 - Blind Mode: The browser reads questions; mic captures answers.
 - Deaf Mode: The camera captures gestures; model predicts text.
6. Submission: Answers are saved to the server for evaluation.

6. Installation & Setup

- 1. Clone the repository.
- 2. Install Python 3.9+.
- 3. Install dependencies: `pip install -r requirements.txt`
- 4. Install Tesseract OCR (e.g., 'brew install tesseract' on macOS).
- 5. Run the application: `python app.py`
- 6. Open browser at `http://127.0.0.1:5000`

7. Future Scope

- Multi-language Support: Add regional languages for TTS and STT.
- Advanced Sign Language Model: Expand gesture vocabulary to full sentences.
- Eye-Tracking Module: For students with severe physical disabilities (motor neuron disease).
- Database Integration: Migrate from JSON to PostgreSQL/MySQL for scalability.
- Secure Browser: Prevent tab switching during exams.