TEXAM – 'AI powered Testing Platform'

Problem Statement

"With The advent of Pandemic , The inefficiency of online testing platforms surfaced along with bulk load of miscellaneous activity during online exam."

The Idea

"Building a Robust AI driven System, to conduct procured test, & minimizing the scope of Cheating to The Maximum Extent."

- Advanced Contact-Less System : Selection of options through hand gestures Through video Stream,
- AI Monitored Video & audio stream : With AI Monitoring The Candidate, it overcomes the inefficiency of teachers to monitor multiple candidates online.
  - Penalties when candidate come closer to system or speaks something.
  - 3 penalties lead to one mark deduction.
  - The background Should be a plan wall to Limit external interface.
- Single Executable binary : Extensive Control over Multiple hardware Devices like keyboard, mouse, camera, etc.
  - Access to System Property like vendor information, prohibit the use of Virtual machine & prevent unwanted modification of system.
  - Reduces Server load by Processing AI on Client side system & increases accuracy as each system process a single candidate's Data.
- OQAT Principal: One Question At a Time, for each question there will be specific time allotted in which you have to answer that & you can scroll to next question only when the previous question ends.
  - "Asking questions one at a time allows you to understand your prospect's challenges in a way that is both meaningful and thoughtful to them.

Features of Client-Site AI:

- Face-Recognition Using HaarCascade: This algorithm selects a small number of important features from a large set to give an efficient result of classifiers then use cascading techniques to detect the face in an image.
- Gesture Recognition : Detect Hand and count number of fingers using Convex Hull algorithm in OpenCV lib in Python.
- Advanced Audio analysis: Using Amplitude Array.

Features of Back-End Server:

- Use of REST API: This enables the making of client applications more secure and easy. Our API Blocks Complete access to the Database from the client and provides limited indirect access for storing responses and accessing questions.
- Containerization of infrastructure: the back-end is containerized so that we can achieve high availability, auto-scaling, redundancy, security, etc. features supported by almost every cloud provider.
- CI/CD: Integration of CI/CD pipelines Automates the build and Deployment process. And automates Testing and deploying and updates/rollbacks, patching processes.

Future Scope:

- Smartphone detection & Multi-face distinguisher.
- Emotion Analysis to provide support to emotionally week candidates to prevent Suicides.
- Backup AI rendering in backend server for system that fails to process at client site.
- Production Deployment on Kubernetes Engine on AWS/Azure. API authentication, more safety and security measures.
- Video and Audio stream using WebRTC.

Team:

- Harsh Singhvi (Backend Server)
- Garvit Chouhan (Front-end & AI)
- Devottam Vaishnav (Debugging & Data analysis)