DIGI-KUL Remote classroom for rural colleges  
Team Details :  
Team Name – Nova  
Team ID - 153  
Team lead - Vabhravi Vashishth , Eno - S24CSEU2534

Team member 2 - Rohit Gangwar , Eno - S24CSEU2533

Team member 3 – Saivats , Eno - S24CSEU1436

Team member 4 – Abhigyan , Eno - S24CSEU0622

Team member 5 – Daksh , Eno - S24CSEU2714

Team member 6 – Anushka , Eno – E23CSEU0142

Problem statement - Remote classroom for rural colleges  
**Problem Statement ID -** 25101  
description (by SIH ) - Background: Rural diploma colleges often operate without subject Lecturers in various fields such as artificial intelligence, VLSI, or renewable energy. Students must rely on self-study material or travel to cities for coaching,deepening the urban-rural learning divide. Description: The challenge is less about willingness than about infrastructure. Typical village campuses juggle low and unstable data speeds, making conventional video-conferencing unreliable. Previous interventions failed when platforms assumed high-bandwidth links or required complicated equipment. To succeed, a new approach must embrace low-bandwidth realities, prioritise audio quality, compress visual content, and ensure that learning can continue even during connectivity lapses. It should encourage synchronous interaction yet also provide recordings that remain small enough for easy download on limited data plans. Faculty in cities need a simple way to deliver lectures from any quiet corner, while rural students require a user experience that functions on entry-level smartphones. Interactive elements quizzes, polls, discussion boards-must remain functional at low speeds.Crucially, the entire solution should minimise the leaming curve for educators and be financially sustainable for resource-constrained institutes. By offering a design that blends live engagement with asynchronous access, the college can bring expert instruction to every campus without waiting for large capital investments.This is precisely the sort of context-aware challenge that Smart India Hackathon teams can address through innovative but lightweight software solutions. Expected Solution: Student teams should outline a software-only virtual-classroom ecosystem that delivers clear audio and concise visual content on limited bandwidth, supports interactive live as well as recorded sessions, and allows easy content access for learners who may need to download materials during off-peak hours-all without relying on specialised hardware or costly licences.   
  
WHAT WE HAVE GATHERED - Rural diploma colleges face several concrete ***constraints:***   
**(1) Limited Internet Bandwidth**: Many areas have

only 2G/3G data (~up to 200–300 Kbps) or highly variable mobile signals.   
  
**(2)** **Intermittent Connectivity**: Connection may drop frequently, so solutions must recover gracefully.   
 **(3) Device Constraints**: Students often have only smartphones (sometimes older models) or a single shared device; universities may lack

campus Wi-Fi or computer labs.   
  
**(4) Power Shortages:** Electricity outages (reported by ~75% of rural faculty) mean the system should allow offline preparation.  
  
**(5) Lack of Backup** Infrastructure: High-cost

hardware (HD cameras, dedicated servers) or paid licenses are often infeasible**.   
  
(6) Language and Literacy**: Content may need local language and low digital-literacy interfaces**.   
  
(7) Financial Sustainability**: The solution should be free or very low-cost to run and maintain (government or student-funded resources are

limited). In short, we need a software-only, highly efficient classroom that can function over ~100 Kbps

links, emphasize audio (speech) over video, and allow asynchronous learning (e.g. downloaded notes,

recorded audio) when real-time is impossible. The system must also work on Android phones or basic

desktops and be easy for non-technical teachers to use.  
  
**Proposed Methodology :**Our approach is a hybrid synchronous-asynchronous e-learning system optimized for low bandwidth.

Key strategies include:

•**Audio-First Live Classes**: We use WebRTC for real-time communication, but prioritize audio streams

over video. If bandwidth dips, video frames are dropped and bitrate reduced to preserve intelligible

audio WebRTC’s adaptive codecs (Opus for audio) automatically adjust quality to network

conditions This ensures students can continue hearing the lecture even with poor connectivity.

•

**Lightweight Video/Presentation**: Teachers can still show slides or whiteboard sketches, but these

are compressed or updated as static images rather than full-motion video. If needed, video is

optional and low resolution.

**Asynchronous Content Delivery**: Lessons are also recorded (audio/video) and uploaded to a central

repository (e.g. Supabase Storage). Students can download them when they have connectivity.

Textbooks, PDFs, and simple animations (cached on-device) supplement live classes, allowing offline

study.

•

**Efficient Signaling and Messaging**: A Flask-SocketIO server provides signaling for WebRTC

connections and real-time text chat. This keeps overhead low (persistent socket vs. repeated HTTP).

For instance, when a teacher starts a session, the server notifies all registered students via a

WebSocket and exchanges WebRTC “offer/answer” packets .

**•Adaptive Quality Monitoring**: The system continuously monitors call quality (packet loss, latency) and dynamically lowers video bitrate or frame rate if needed This follows best practices in teleconferencing under bandwidth limits.

**•Push-to-Talk & One-Way Audio (if needed**): Optionally, to save uplink bandwidth on the teacher’s

side, students might mostly receive audio (with the teacher speaking continuously). Students can ask

questions via text chat or brief voice turns.

•**Scheduling and Notifications**: Teachers schedule classes in the portal (React web app), and

students get notified. This reduces wasted connection attempts.

Overall, the methodology is mobile-first and audio-centric, leveraging WebRTC’s peer-to-peer data

pathways to minimize server load, and using common open-source components to keep costs down.  
  
•**Offline App Mode**: Enhance student app to work fully offline once materials are synced; include

interactive quizzes or assessments that can grade without server access, uploading results when

online.

•**Device-to-Device Sharing:** In areas with NO internet, allow one connected student to act as a local

hotspot/edge node and share downloaded content to nearby peers via Wi-Fi Direct or Bluetooth

Mesh.  
  
System Architecture:  
  
A screenshot of a computer program

AI-generated content may be incorrect.