

SIH 25 – Brief summary OF TOPICS

Problem 25016 – Automated Student Attendance Monitoring & Analytics

Concept

- **Hybrid attendance automation:** Combines **face recognition** (for in-person) + **QR/Geo-tagging** (for online/offline verification).
- Analytics dashboard with **early warning system** → alerts if attendance < threshold.
- Works both **offline-first (local device storage)** and **syncs to cloud** when internet available.

Prototype Build

- Use **OpenCV + Mediapipe** (face recognition) OR **QR Code generator** for online lectures.
- Store attendance in **SQLite (offline)** → sync to **Firebase/Cloud DB**.
- Dashboard → **React (frontend) + Django/Flask backend**.
- Analytics → **attendance trends, heatmaps, auto-mail alerts**.

Workflow

1. **Student enters class** → Camera auto detects & marks presence.
2. **Online student** → Enters class → generates dynamic QR (time-limited).
3. Attendance stored locally → sync to cloud.
4. Faculty dashboard → View real-time attendance, download reports.
5. Analytics → Predictive insights (low attendance alerts, engagement trends).

Zero-Flaw / Innovative Additions

- **Anti-proxy check:** live face detection → no photo uploads.
- **Time-limited QR** → can't share with friends.
- **Works offline** → stores & syncs later.
- **Lightweight AI** → runs on low-end laptops.

Problem 25104 – Language-Agnostic Chatbot

Concept

- **Campus Info Assistant** → Multilingual AI chatbot (Hindi, English + 3 regional).
- Powered by **LLMs + Translation APIs** for natural conversation.
- **Context retention** → remembers follow-up questions.
- **Student-maintainable** → FAQs updated from Google Sheets or simple admin panel.

Prototype Build

- Use **Rasa / Botpress / Dialogflow CX** for chatbot engine.
- Integrate **IndicNLP or Google Translate API** for regional languages.
- Embed chatbot in:
 - College website (iframe + widget)
 - WhatsApp/Telegram (via Twilio / Bot API).
- Logs stored in DB for daily review.

Workflow

1. Student asks question (any language).
2. Bot → translates → matches intent → fetches FAQ from DB.

3. Responds in student's original language.
4. If unanswered → fallback → route to human (email/chat).
5. Admin → Reviews logs → updates FAQ repository.

Zero-Flaw / Innovative Additions

- **Multi-platform** (Website, WhatsApp, Telegram).
- **Daily auto-learning from logs** → updates FAQ.
- **Low infra maintenance** → FAQ stored in Google Sheets.
- **No vendor lock** → Open-source stack (Rasa, Hugging Face models).

Problem 25101 – Remote Classroom for Rural Colleges

Concept

- **Low-bandwidth-first virtual classroom.**
- Prioritises **audio + compressed visuals**.
- Provides **offline download packs (recordings, PDFs, quizzes)**.
- Built like "**Zoom Lite**" for villages" → works on 2G/3G & entry-level smartphones.

Prototype Build

- **WebRTC + Opus Codec** → for high-quality audio at low bandwidth.
- **Slides compressed to images/PDF** → transmitted instead of video-heavy streams.
- Use **progressive download** → if internet drops, recording still downloads later.
- Mobile app (React Native) → supports live + offline content.

Workflow

1. Teacher starts lecture → Streams **audio-first + slide snapshots**.
2. Students join → Receive compressed content (adaptive quality).
3. Interactive layer → **low-data quizzes/polls**.
4. Recording → Auto-compressed → downloadable later.
5. Offline mode → Students download lecture packs overnight (low-data hours).

Zero-Flaw / Innovative Additions

- **Audio prioritized** (class continues even if video fails).
- **Dynamic compression** → works on 100kbps speeds.
- **Offline sync** → Students don't miss learning if network drops.
- **Faculty side simple** → Upload PPT/Doc, app auto compresses & delivers.