**ClassSync: An e-Learning Engagement Suite for Remote K-12 Classrooms**

**Overview**

Provide educators a web-based tool for tracking actionable data on students’ activity and habits during live classroom sessions and home assignments.

User stories:

teachers(admins),

students(regular users),

developers

**MVP**

* Teachers are able to access a live dashboard to monitor students’ activity and engagement levels during live “synchronous” classroom activities. Metrics may include:
  + Time speaking (Web Speech API)
  + Time in front of webcam/webcam turned on (TensorFlow/getUserMedia)
  + Patterns of keyboard/mouse activity (keyboard/mouse DOM events)
* Outside of class time teachers are able to access summary statistics and trends of students’ metrics over time
* Teachers are able to generate “engagement report cards” outputting a form with targeted summaries of an individual student’s metrics and progress over time

**Stretch Goals**

* Further activity monitoring functions:
  + Time spent on activity page vs unrelated tabs/applications (puppeteer, chrome window/browser listeners, extension messages?)
  + Time spent to complete an assignment (will require start/stop buttons on student’s end unless we are integrated with their assignment document)
* Teachers are able to organize students on the live dashboard into their respective breakout rooms and examine data unique to these groups
* Functionality is expanded and refactored for monitoring of “asynchronous” homework activities, able to account for time spent on other activities
* Facial recognition / Optical scanning to give more granular feedback on students’ distractibility/attention
* Teachers are able to review screenshots of students’ screens at key moments
* Speech is analyzed to determine whether it is on-topic
* Teachers can input students’ grades and see visualizations of their correlation with engagement metrics

**Technical Challenges**

* Facial recognition/optical tracking
* Audio processing/voice recognition
* Data visualization
* Real-time activity monitoring (using sockets?)

**Technologies to be Used:**

* Web Speech AP
* TensorFlow/getUserMedia & MediaStream
* Puppeteer
* ChartJS
* Keyboard/mouse DOM events
* Materialize for UI
* React / Redux
* Sequelize / Express