University of Arkansas – CSCE Department
Capstone I – Final Proposal – Fall 2020

Emotion & Gesture Tracking Browser Extension for use in Classrooms

Caleb Duke, Allison Frye, Cassidy McManus, Ryan Rau, Gage Robinson

Abstract

COVID-19 has altered the way we interact with large groups, forcing us to turn to a digital environment. The academic world has been hit hard, forcing college students to learn via Teams, or other video sharing platforms. In the virtual environment, information is still transferred from teacher to student, however the non-verbal cues of the students to the professor are lost. Professors have a harder time of "reading the room" as virtually every student has their camera and microphone turned off.

By creating an emotional and gesture tracking system, we can restore a small piece of human interaction by emulating how a real classroom would function. Creating a Chrome extension that can read in a student's emotion or capture certain gestures in real time and relay that back to the teacher can provide a small sense of normalcy. Teachers can easily see who is interested, confused, or have left the class as naturally as they could if they were face-to-face. This will remove the awkward silences when a question is asked and can also help teachers learn how engaging they truly are in lecture.

1.0 Problem

With the rise in popularity of online classes, students are finding it increasingly more difficult to stay engaged while attending lectures. The lack of any kind of face-to-face interaction between professors and students easily leads to students not having as much of a drive to pay attention and attempt to understand the material being presented. Moreover, professors often find it nearly impossible to gauge how students are feeling based off lack of physical and verbal cues given during the current lecture.

Ultimately, it is the responsibility of the student to attend online lectures, take notes, and ask questions. However, this style of instruction simply does not lend itself to normal classroom interactions amongst students and between students and professors. The implications of this range in severity. Some students may simply do worse in a particular class if they cannot engage like they normally would or have that small sense of peer pressure to remain attentive.

Without a proper solution to this issue, we will see a steady decline of a student's well-being in both academic success and social skills. Without a proper solution to this issue, we will see a steady decline of a student's academic and social well-being. Teachers can no longer engage with their students in the same way they once could, and students often feel like they are missing out on the true college experience, particularly not seeing teachers and fellow students in person, and meeting new people with similar interest. By not having a concrete solution, motivation for holding a productive learning environment will be lost; therefore, emulating human interaction to some degree will help maintain the classroom dynamic.

2.0 Objective

The objective of this project is to develop an application that allows interpersonal connections through online learning. With this application, our group hopes to bridge the current emotional and physical gap between teachers and their students due to virtual learning for more engagement between the two. We plan to do this using facial and gesture recognition to track if a student is feeling confused, content, or has simply left "class" for an extended period. Tracking certain gestures and facial expressions can gauge a student's engagement or presence during the class. This information will be provided as live feedback for the instructor to regain as close to a normal classroom experience as possible.

3.0 Background

3.1 Key Concepts

The first key technology relating to this project is web conferencing. Web conference platforms are the closest medium to reproducing classroom settings online. Attendees log onto a specialized webpage representing the session and can watch their instructor present live lessons. The instructor usually communicates to students through their microphone, camera, and slides. Meanwhile, students generally rely on text-based chat to give feedback to their instructors, at times using their own microphone and camera. A few examples of popular web conferencing tools are Zoom, Microsoft Teams, Wimba, Adobe Connect, and Blackboard Ultra. Our project will have an emphasis on using Zoom and a Chrome extension to extract the live feed.

The second key technology is computer vision, which is the use of artificial intelligence to analyze images and other visual data. With recent innovations in deep-learning and neural networks, researchers and developers have made remarkable improvements on the speed and quality of computer vision techniques. The most relevant application of computer vision for our project is reading facial expression and certain body gestures. Some notable deep learning frameworks we plan to use are Amazon Rekognition, PyTorch, and Teachable Machine.

3.2 Related Work

Developers of web conferencing platforms have largely relied on chat, microphone, and camera usage for in-session student feedback. This implementation can be seen in Zoom, Wimba, Adobe Connect, and Blackboard Ultra. Platforms may also allow students to display emoticons or other special symbols next to their name.

However, non-vocal student feedback, such as facial expression and posture, are still largely under-represented in these platforms. While camera usage allows students to display posture and expression, they often must compete for screen real estate with slides, other lecture aids, and each other. With each student appearing the size of a postal stamp if at all on their instructor's screen, it can prove difficult to perceive subtle non-verbal feedback from larger classes. This means that many instructors essentially present to an invisible audience, unable to use non-vocal feedback to adjust the style and pacing of their presentation to best engage their students.

Recently, non-verbal student feedback detection with AI has been explored in detail by researchers. In a 2015 paper published by Pennsylvania State University, researchers used deep learning to analyze student engagement in a computer-based activity via footage captured on standard web cameras [2]. In the Summer of this year, Beijing-based researchers applied a similar machine learning approach to an online classroom, again using webcam footage to monitor student emotions [3]. Both experiments were carried out in normal settings where lighting may be sub-optimal.

However, both implementations did not primarily focus on the logistics of carrying out facial-expression analysis in real time for the duration of a class period. They also did not cover the distribution of their implementation to students and instructors. For our project, we intend to develop a machine-learning-based expression recognition on display in these papers into a usable application, complete with session-management functionalities, a robust backend database, and an effective and simple user interface for students and instructors alike.

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

Our application falls under two use cases: one as a student and one as a teacher. According to one's role, different views will be presented along with functionality. Below are the respective requirements:

Student Requirements

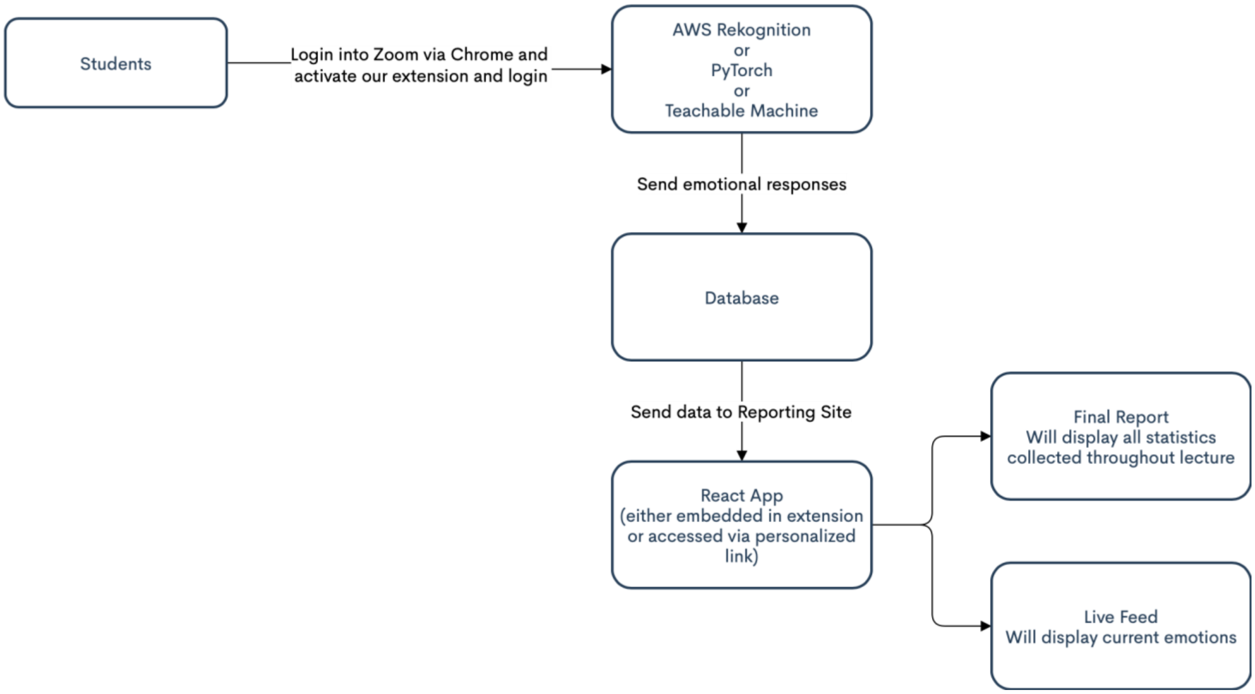
1. Allow students to login and direct them to a modal (small message box) asking for a classroom password. The University of Arkansas's Lightweight Directory Access Protocol (LDAP) may be incorporated. This would allow us to utilize students and professors UARK usernames and passwords without them having to make an additional account for our application.
2. Ask students if they may have access to the camera.
3. Allow students to use Zoom as usual.

Teacher Requirements:

1. Allow teachers to login, while generating a classroom code to send to students.
2. Allow teachers to conduct a Zoom based class experience while being able to view live reactions.
3. At the end of a session, a report will be generated to allow the teacher to analyze their students emotional state over the class period

4.2 High-Level Architecture

We are taking a microservices approach by compartmentalizing the functionality of our project. As these are stand-alone services, the core parts will be the backend, ML Models, and the Chrome Extension. Below is a high-level mock-up of how the sections will connect with each other:



<u>Backend</u>

Our backend will be comprised of two parts: the database and the RESTful APIs. As of now, we plan to test for confusion, contentment, and long periods of inactivity/absence, by taking a succession of pictures and averaging out which mood the student is in. This data will be sent directly to the backend and passed along to AWS Rekognition where it will be analyzed, with the results being stored in our database.

Once we implement other machine learning models for additional gesture recognition, we will follow a similar process. Where the backend will route the incoming data to the appropriate place to be analyzed and its results to be written to the database

With the data we collect, we will create endpoints which will assist in crafting final reports along with providing close to real time reports for the teacher during class. We will limit the amount of personalized data we store, however due to the nature of the program we need to have some level of identification, such as name and email, to go with the data we collect. That said, we plan on making it apparently transparent to the student what is being collected in a terms and conditions which they must accept to before using our program.

The APIs will allow communication among AWS Rekognition, the database, and the extension. The language will be selected when the time comes to connect all the components, however JavaScript and Python are top contenders. The database and RESTful APIs will be hosted on a virtual machine provided by the University of Arkansas. We plan to have a MySQL database with the APIs built with NodeJS/Express or with Flask. This has not been finalized.

Machine Learning Model and Implementation

At its most basic state, our project will consist of AWS Rekognition machine learning models. Rekognition provides us with easy to access pretrained machine learning models which can return the perceived emotions of a person from a still image, video clip, or video stream. Our Chrome extension will provide the data (either images or video clips) needed by Rekognition and return its results of perceived emotions to our backend for further processing and formatting so that it can be displayed to the teacher in an insightful way.

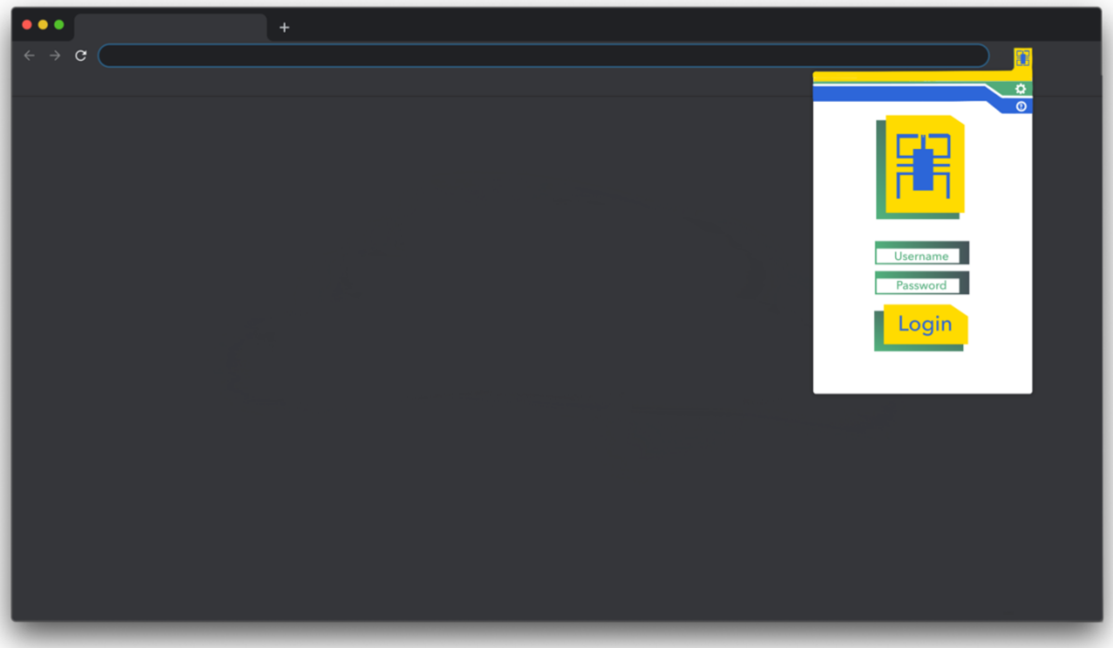
Once base functionality is set up, we will look into expanding the capabilities of what our program can detect from the data provide by the Chrome extension. This will consist of either building off the AWS environment or building our own models using tools such as Google Teachable Machine or PyTorch. These additional models could detect a variety of additional gestures bettering the data we can return to the teacher.

Chrome Extension

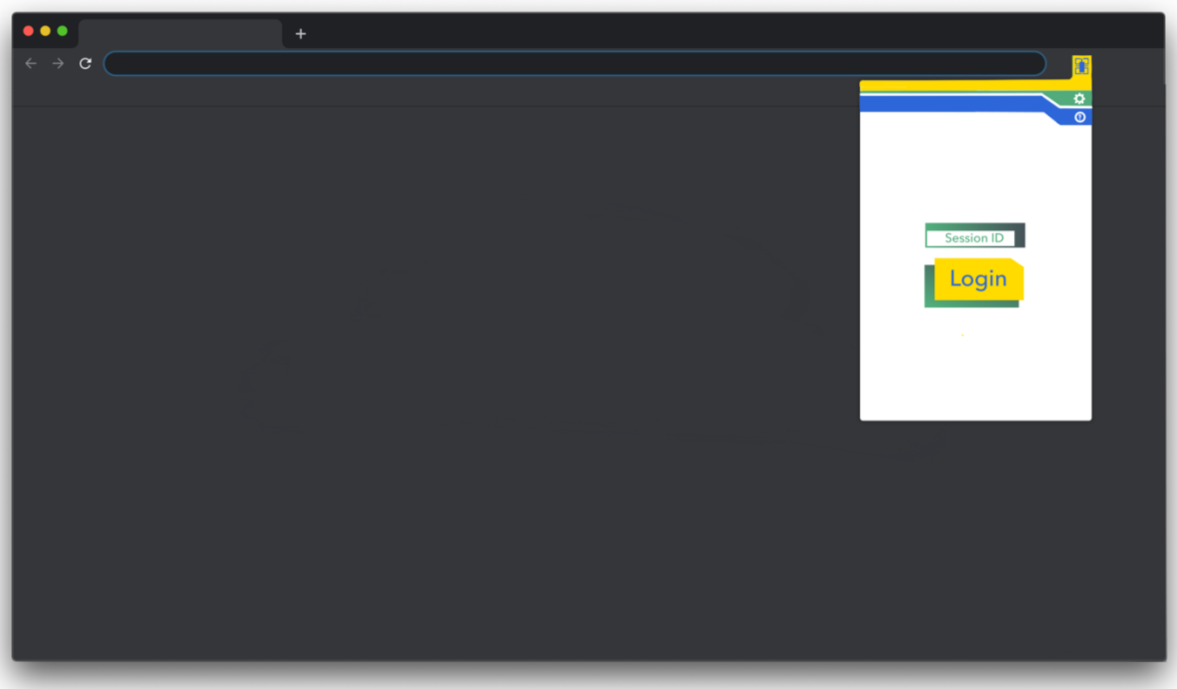
The Chrome extension will provide both the student and teacher a log in interface. For the student, after logging in through LDAP and entering a session ID, they will not have much interaction with the extension as it will then just collect data throughout class time. For the teacher, the extension will provide a quick glance at the emotion and gesture data collected during the class along with a link which will redirect the teacher to a more in-depth report of the class.

By using a Chrome extension, we give our program the capability to use any video conferences solution which allows you to join via web browser. Rather than tying into just one service such as Zoom; the extension approach will allow us to work with Blackboard collaborate, Zoom, and Microsoft Teams to name a few.

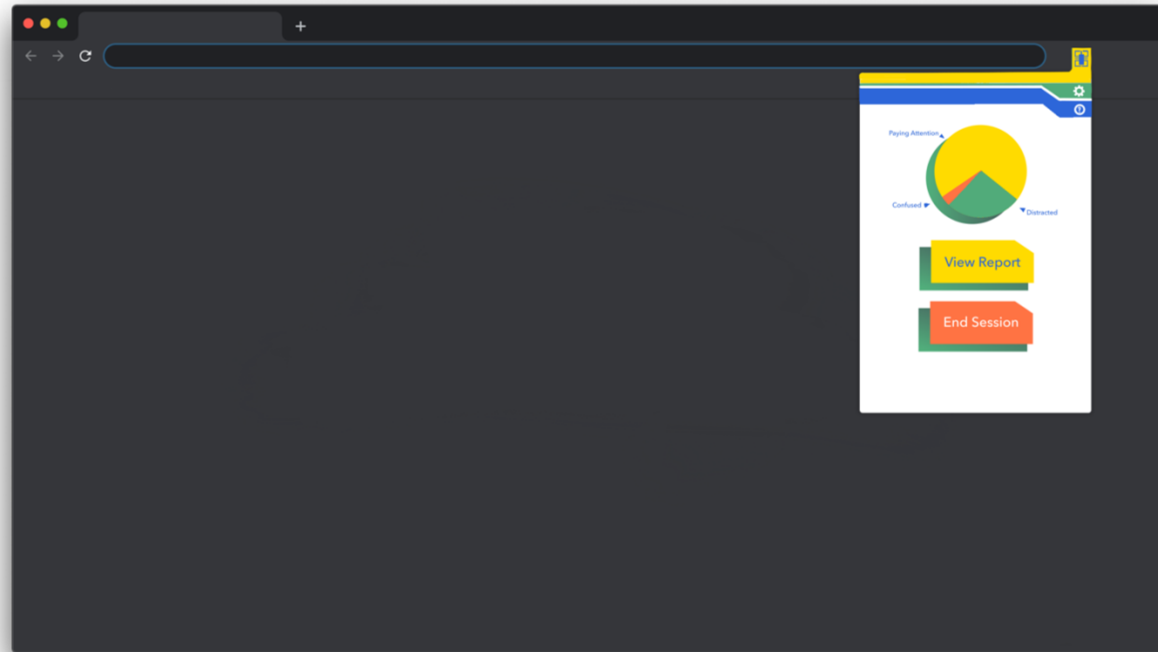
We have fabricated a series of mock-up designs for the user interface:



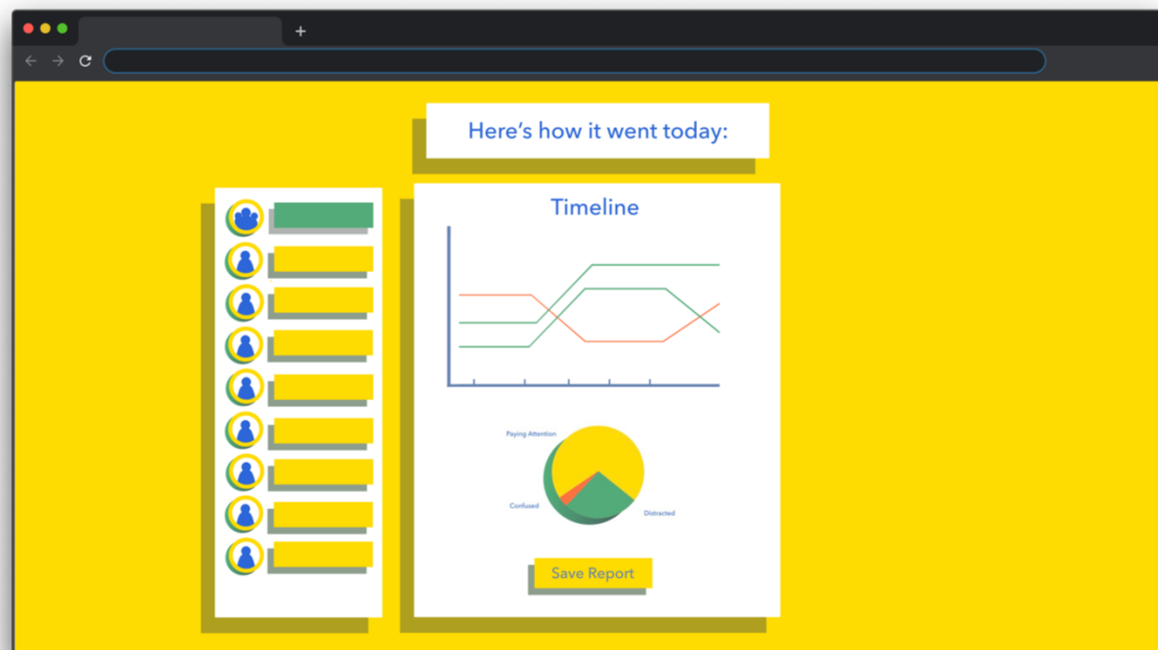
1: User Login for Students and Faculty



2: Session Login page for Students



3: Instructor's view of session



4: Reports page generated for Instructors

4.3 Risks

Risk	Risk Reduction
Performance impact of the Chrome extension's connection with backend and machine learning models	We will test the performance impact of our extension by sending video clips, still images, and a live video feed to determine which is the most optimal in terms of impact on the user's machine and feedback from the machine learning models.
Real time data getting out of sync between users and database	Implement web socket connection between our Chrome extension and backend to ensure that data is updated in real time. Additionally, include a level of error checking that will warn the user when a connection has been lost.
Users accessing data which they should not have access to	Implement a role system to distinguish between students and teachers. Also ensure that a user is authenticated when calls to our API are made.
Privacy concerns	Address privacy concerns in a terms and conditions agreement informing the user of what data is being accessed.

4.4 Tasks

1. **Preliminary Research**: We will explore technologies we plan to use to in depth, thus ensuring that by the Spring we will have a level of familiarity and comfort of what we will be using. During this time, we will get a better idea of what our limitations are when it comes to our ideas that go past the scope of our originally presented project. With that information, we will be able to adjust our future plans and tasks to ensure that everything is reasonable and within our capabilities to ensure success.
2. **Initial Setup**: During this time, we will setup up the basic frameworks for each component within our project ensuring that all team members have a running local copy along with access to the parts of our projects that will be hosted on the University of Arkansas's virtual machine.
3. **Backend, Extension, Machine Learning Models & Implementation**: Splitting up the tasks and working in parallel on the three separate components will allow us to work more efficiently and allow everyone a chance to learn different areas. During this time our focus will be to get each component working as a stand-alone service. In addition, we will have a better understanding of how the data in each step will be formatted which will be helpful later on.
4. **Combining Components**: With the three components being setup independently we will start to bridge the gaps making all the services communicate with each other. We will be thorough when merging the components together and test accordingly to ensure seamless integration
5. **Expand AI**: With AWS Rekognition only providing facial gesture recognition, we believe it would be beneficial to go past that and expand what our program can recognize. That said,

we'll spend this time to either implement other pretrained machine learning models or to train our own so that our program can recognize more than just facial gestures. During this time, we could use tools such as Google Teachable and PyTorch but that's subject to change as we plan to explore more of our options during our preliminary research.

6. **Reporting**: We will want to provide the professor with an attractive interface to see all the data we collect. We will design and build a reporting site that will present that data in a way that can be insightful to the teacher. This will include graphs that will show change in the students' emotions (facial gestures) over a period of time along with any other meaningful data we collect from expanding what our program can recognize.
7. **Optimization and Testing**: Though we plan on testing everything as we build, it is good to take some time to really push it to its limits. We will attempt to break our program exposing any bugs that may have slipped through the cracks and address them as needed. Additionally, we will clean up any spaghetti code that might be hindering performance, optimizing all that we can to improve the user experience.
8. **Final Touches and Report**: At this point everything should be working, and all the kinks should have been worked out. The remaining time will be spent cleaning up the UI, making final tweaks and writing the final report.

4.5 Schedule

Tasks	Dates
Sprint 1 (Preliminary Research) • Iron out database structure • Determine best language(s) for backend • Explore AWS Rekognition capabilities & limitations • Explore other methods of training machine learning models and how to implement them • Look into the capabilities & limitations of the Chrome extension component • Explore frontend frameworks to decide which one the team is most comfortable with	11/7 - 1/11
Sprint 2 (Initial Setup) • Get instances our database and base backend set up on our local machines along with on a shared version on some university virtual machine • Set up AWS environment ensuring all team members have access	1/11 - 1/25
Sprint 3 (Backend, Extension, ML Development) **Backend**	1/26 - 2/9

• Setup the backend API with endpoints for basic CRUD operations on the database tables, • Setup endpoints and logic to allow for a connection to & from both the extension and AWS **AWS** • Setup Rekognition models, preparing it to accept data from backend • Explore best method of data input (video clip, still image, live stream) **Chrome Extension** • Create Chrome extension which has access to the user's camera • Create a background task for image handling preparing data to be sent to backend	
Sprint 4 (Merge Components Together) • Link up extension to send and receive data from backend • Link up backend to send and receive data from AWS • Preform testing to ensure flow from start to end is solid.	2/10 - 2/24
Sprint 5 (Expand AI & Start Reporting Page) • Begin expanding upon what our program can recognize • Start training and implementing new machine learning models • Begin teacher reporting page	2/25 - 3/11
Sprint 6 • Further expand upon gesture recognition capabilities & reporting page • Begin to explore and implement any additional features	3/12 - 3/26
Sprint 7 • Continue with additional features • General optimization & testing project wide	3/27 - 4/12
Sprint 8 (Final Touches, Report, Optimization) • Clean the project up, final touches, final report, etc…	4/13 - 4/27

4.6 Deliverables

- Chrome Extension: A Chrome extension which has access to the user's camera and sends that data to our backend through APIs. Extension will also provide UI for the user to login.
- Backend: API + Database Schema Documentation: Document with all the APIs' endpoints, inputs, and outputs. Included will also be the languages used. In addition to API information, the database schema and language will be listed.
- Machine Learning (Emotion & gesture recognition) models and code: At our projects core functionality this will consist of the required code to communicate with AWS Rekognition along with the necessary documentation to get this code running. Noting any special setup that needs to be done on AWS. Past our project's core functionality, this deliverable will consist of trained machine learning models along with the implementation code of said models.
- Design Document: Document explaining how the three components interact with one another. All technologies will be listed.
- Final Report
- GitHub Repository List: As we are taking the microservices approach, each component will have their own repository, thus a location in which all links are consolidated will be given.

5.0 Key Personnel

Caleb Duke - Duke is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed a majority of the CSCE core classes. He has formerly served as a graphic designer and project lead at the UITS Professional Development department. He is currently working for Charger Marketing in Front-End web development and Graphic Design, as well as a research support assistant at the UARK College of Education and Health Professions. He will be responsible for the front end and creating the Chrome Extension. He is planning to continue to work with Charger Marketing after graduating in May 2021.

Allison Frye - Frye is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed the majority of the core CSCE courses ranging from Programming Foundations to Algorithms. She has completed internships with Michelin and Cerner as a Software Engineering Intern with a focus on backend work. She will be responsible for making the APIs and assisting in the implementation of AWS Rekognition. Currently, she interns with Affirma as a Software Engineering Intern, serves as a First-Year Engineering Peer mentor, and works as an Extra-Help assistant in the CSCE Tech Office. She's set to work at Cerner after graduation.

Cassidy McManus - McManus is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed relevant courses such Programming Paradigms, Software Engineering, and other higher level CSCE courses, all of which have provided foundational skills for this course. McManus currently is a lab member in Dr. Di's TruLogic Lab, has web development experience using Python Django through UATeach, and works as a lab instructor for Digital Design. She will be responsible for assisting in the front end and database implementation. She is planning on attending graduate school at the University of Arkansas after graduating in May.

Ryan Rau - Rau is a 4th year Computer Science and German major at the University of Arkansas. He has completed a majority of the core CSCE courses including courses such as Algorithms and Software Engineering. Additionally, he is currently taking Artificial Intelligence which could prove to be useful when working with AWS Rekognition and building additional gesture recognition systems. He will be responsible for assisting in the implementation of AWS Rekognition and for the machine learning models. Outside of classes, Rau works at Affirma as a Software Engineer Intern working on a variety of projects including frontend design and business data processing.

Gage Robinson - Robinson is a senior Computer Engineering major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed a majority of his Computer Engineering classes. Currently he is taking Embedded Systems and will take Computer Architecture his last semester. He had a web development internship which was cancelled due to COVID-19 restrictions. He will be responsible for creating the database schema and setting up the database service. He is set to begin employment as an Embedded Systems Engineer at Honeywell in Kansas City, Missouri at the end of May 2021.

6.0 Facilities and Equipment

- Laptop with webcam
- Access to AWS Rekognition
- University virtual machine

7.0 References

[1] Galusha, Jill M., "Barriers to Learning in Distance Education", Institute of Education Sciences, 1998
[2] Weiqing Wang, Kunliang Xu, Hongli Niu, Xiangrong Miao, "Emotion Recognition of Students Based on Facial Expressions in Online Education Based on the Perspective of Computer Simulation", https://doi.org/10.1155/2020/4065207, 2020
[3] Nigel Bosch, Sidney D'Mello, Ryan Baker, Jaclyn Ocumpaugh, Valerie Shute, Matthew Ventura, Lubin Wang, Weinan Zhao. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.1022.7898, 2015