

Emotion Tracking Browser Extension for use
in Classrooms

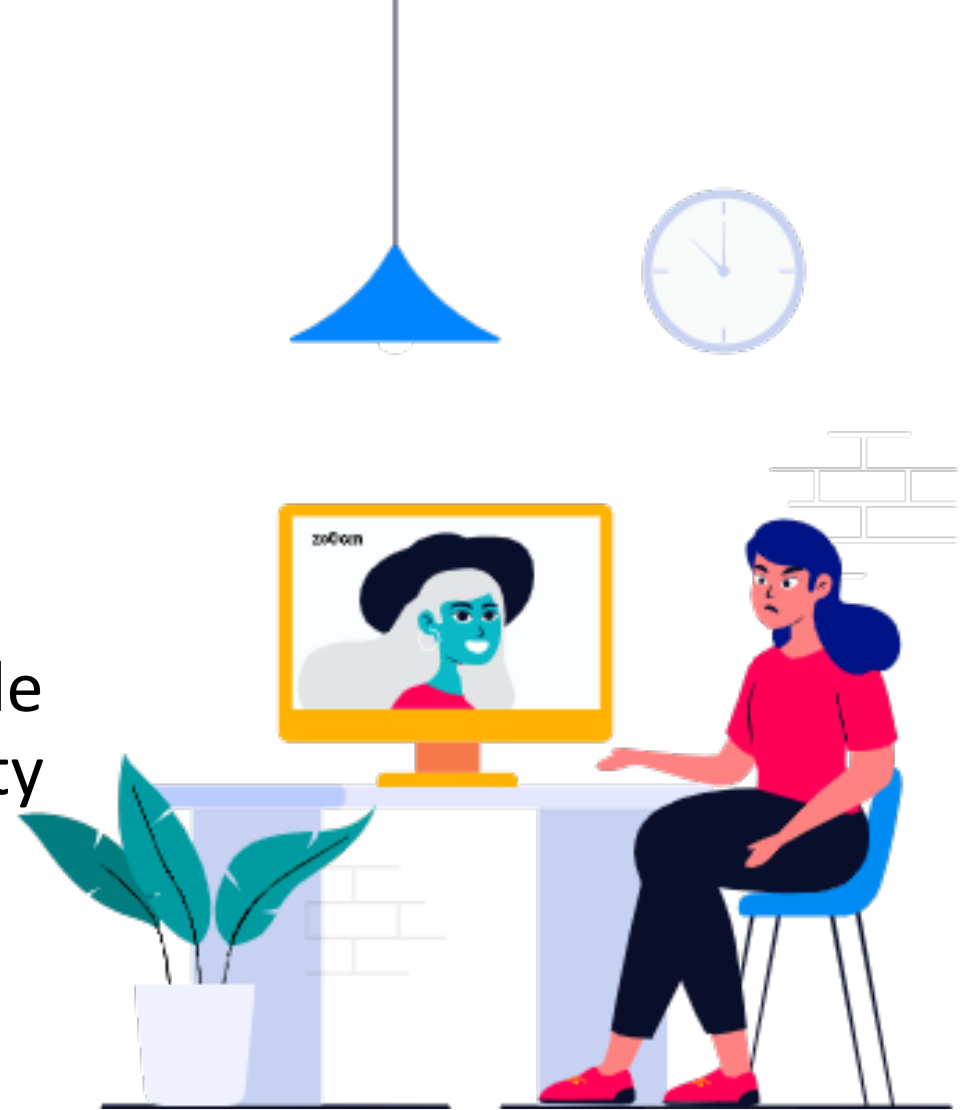
Final Update

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



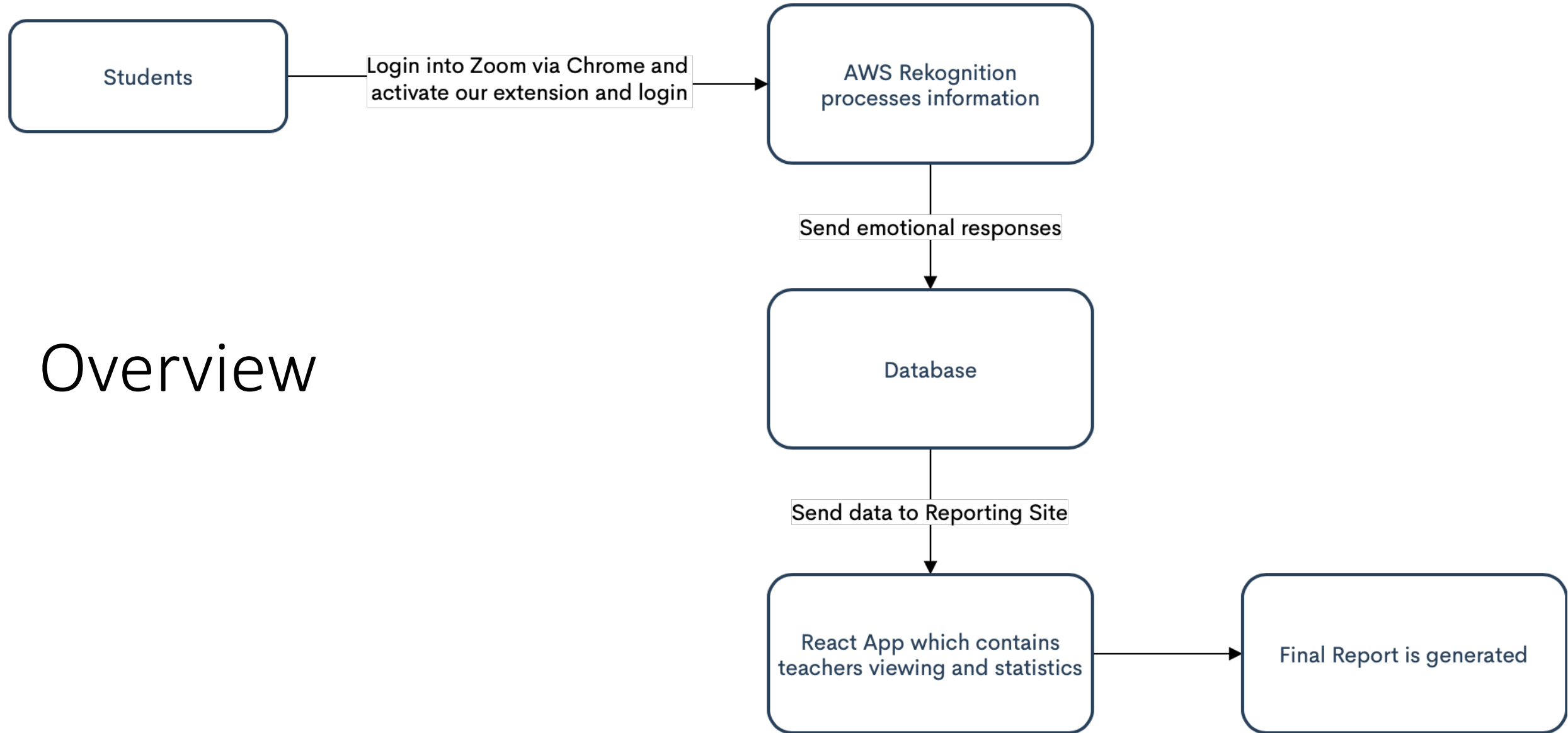
The Team

- Develop an application that allows for interpersonal connections in a classroom during virtual learning
- Bridge the current emotional and physical gap between students and instructors
- Use facial recognition to track student's facial expressions, emotions, and presence to provide near real time feedback of the student's activity for the instructor



OBJECTIVE

Overview

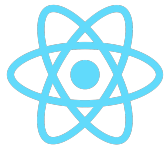


ARCHITECTURE

- All services are built and connected
- All are hosted on VM, except Chrome extension is local for now



Backend

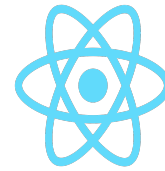


- Pinned Tab
- Reporting Page



amazon Rekognition

- Python Server



Chrome Extension



Database



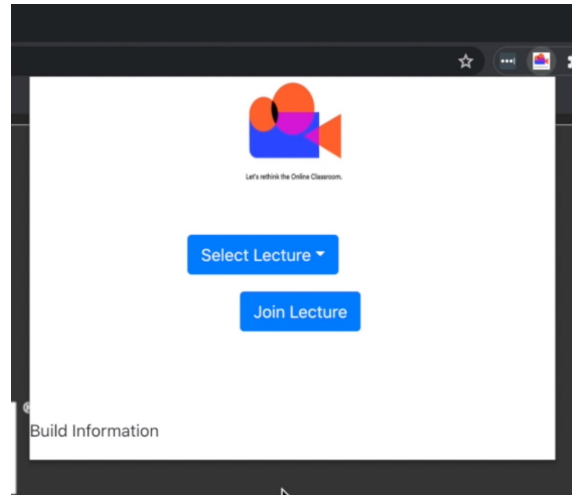
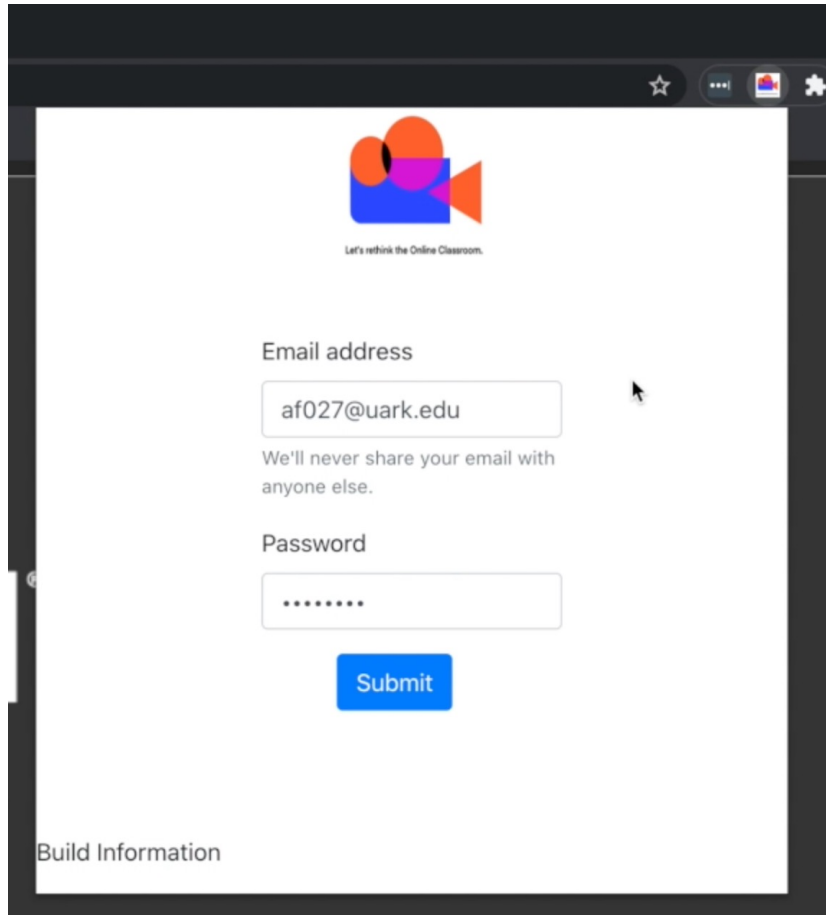
Tools



UNIVERSITY OF
ARKANSAS

Virtual Machine

Previously....



- Chrome Extension
- Capturing page
- General graphing page



Collecting data...

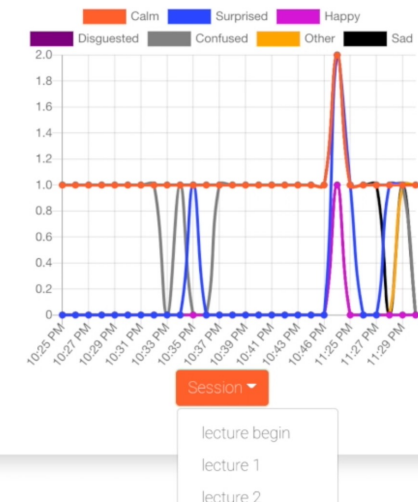
Data is currently being collected for:

User ID: 1

Lecture ID: 122

No pictures are saved! All pictures are disposed of after emotion analysis.

Dev Tools



Additional Things Added Since Midpoint

- Teacher's page
- Python Queue System to Tile Images
- Predictive Modeling
- All around styling improvements

Student Usage Demo



Teacher's Page

Key Features:

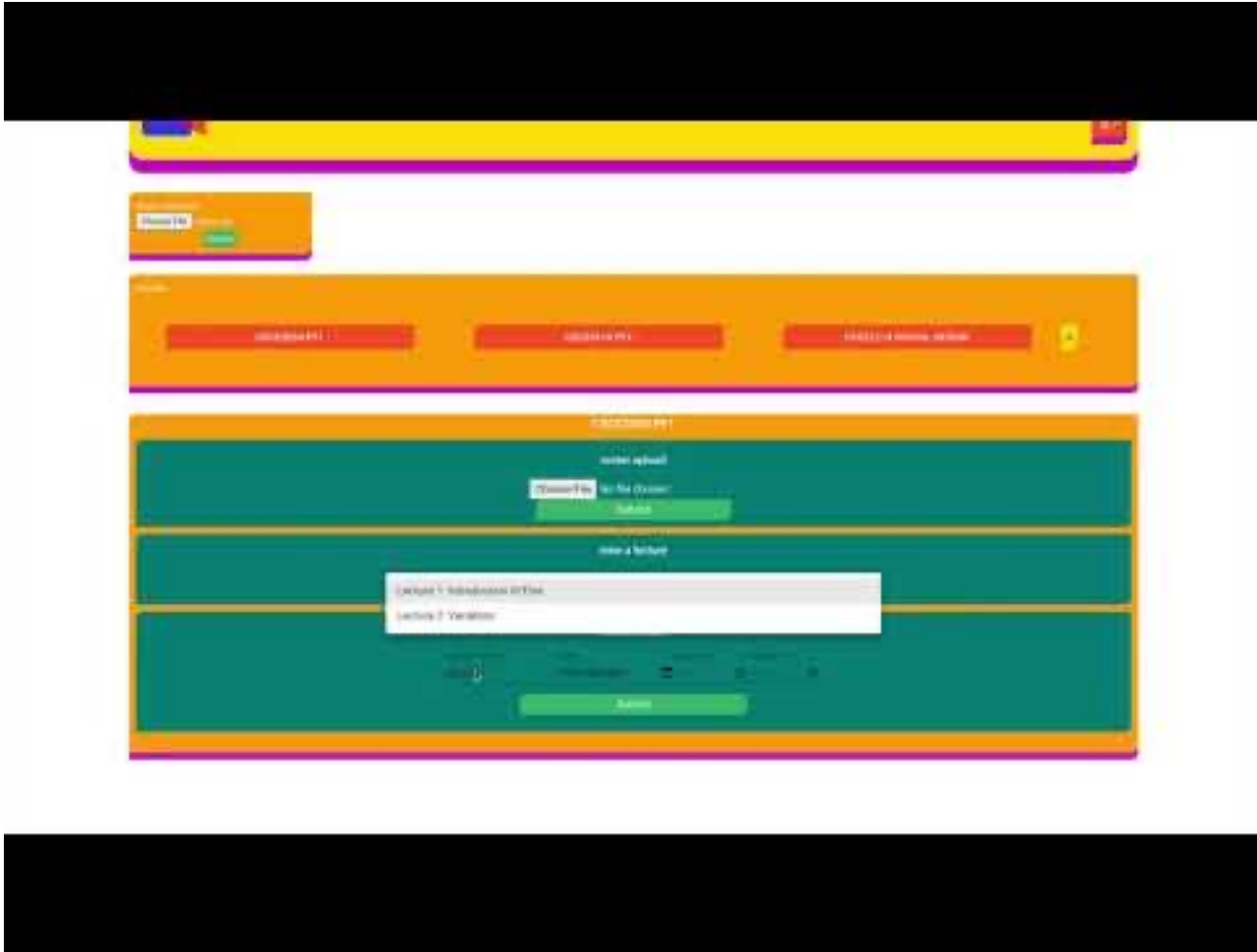
- Import a CSV file of students
- Import roster for a class (1:1 section to class)
- View statistics on a lecture
- Create new lecture

The screenshot displays the Teacher's Page interface with a yellow header and purple accents. At the top right is a menu icon. Below the header is an 'Import Students' section with a 'Choose File' button, 'No file chosen' text, and a 'Submit' button. The 'Classes' section shows three class cards: 'CSCE2004 PF1', 'CSCE2014 PF2', and 'CSCE2114 DIGITAL DESIGN', with a plus sign for adding more. The 'CSCE2004 PF1' class is selected, showing a 'roster upload' section with a 'Choose File' button, 'No file chosen' text, and a 'Submit' button. Below this is a 'view a lecture' section with a dropdown menu showing 'Lecture 1: Introduction If/Else' and 'Lecture 2: Variables'. At the bottom is a 'create lecture' form with fields for 'Lecture Name', 'Date' (mm/dd/yyyy), 'Start time' (calendar icon, --:-- --), and 'End time' (clock icon, --:-- --), and a 'Submit' button.

Using CSV Files for Large Database Operations

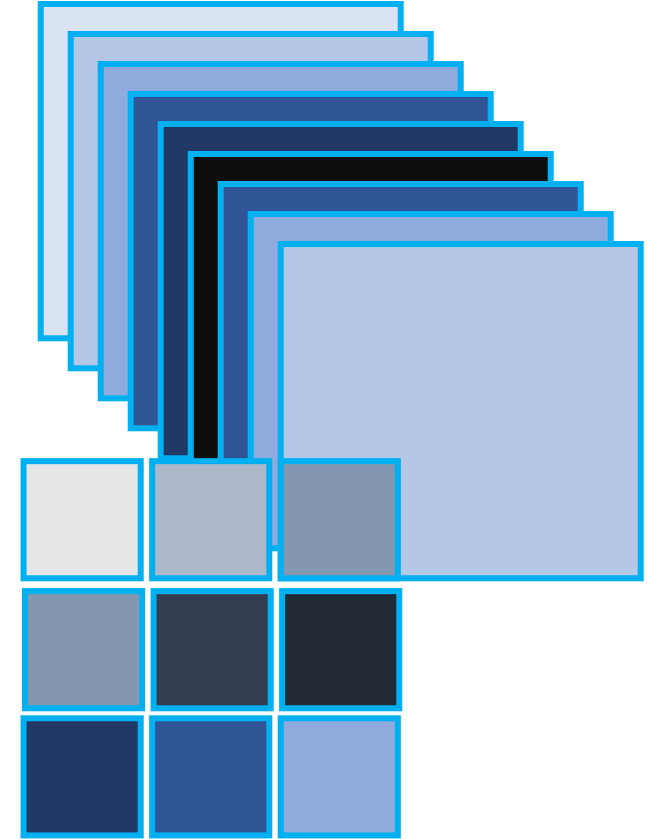
- Take in CSV from teacher's page using OpenCSV, and save endpoints to respective tables in database
- This is particularly useful when wanting to create many users at once, whether it be from an Excel file, or from a list of existing enrollments.
- A brief explanation of how the logic works:
 1. Checking and handling case of an empty file.
 2. Instantiating a CsvToBean object, which allows us to transform data within a CSV into Java objects.
 3. Creating a list of Class Roster DTO objects that can be iterated through.
 4. For loop that iterates through the DTO list and calls entity set methods to assign proper values.
 5. Call to the proper method in the commands class create the new Class Roster.

Teacher Page Demo



Queue and Tiling

- Our back-end enqueues images from each user as they arrive
- The once the queue reaches 9 images, the images are tiled together in a composite image
- This composite image is sent to AWS for processing
- Why do it this way?
 - Reduces the total amount of images sent to Amazon Rekognition which keeps costs low.

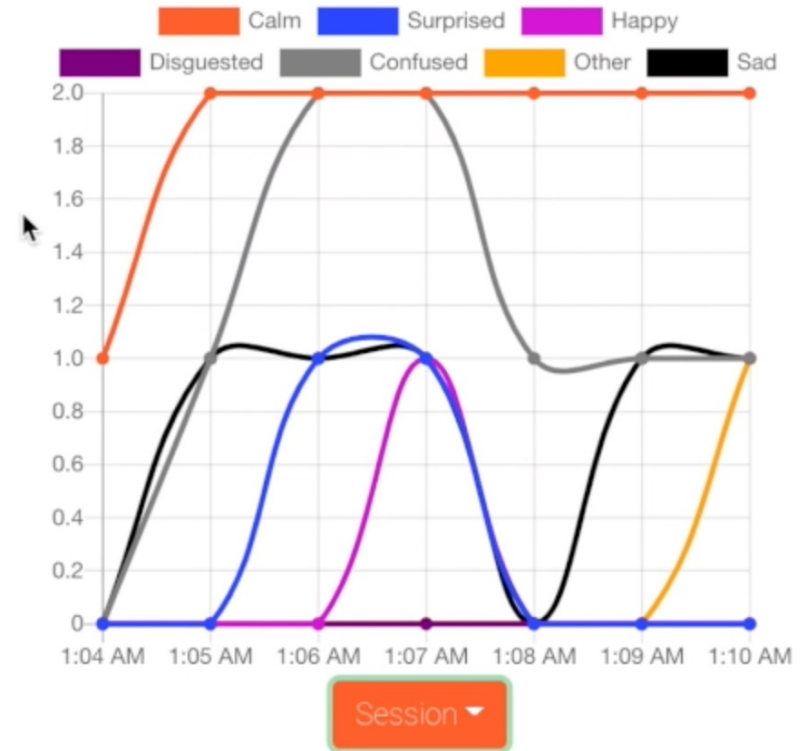


Near Realtime Data Processing Demo



Predictive Modeling

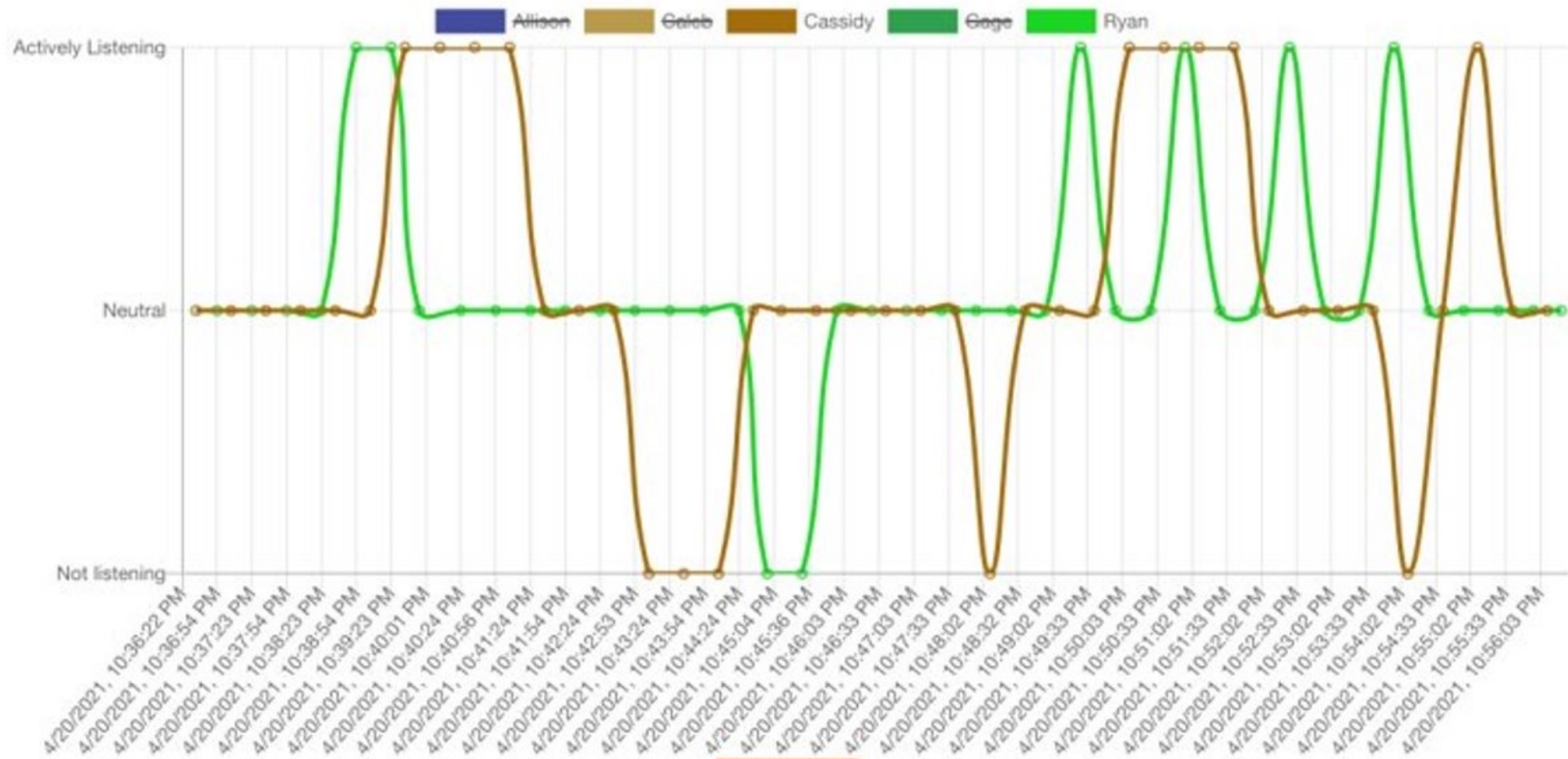
- AWS returns percentages for the following emotions:
 - Calm
 - Surprised
 - Happy
 - Disgusted
 - Confused
 - Sad
- Graphed alone not that useful...




Predictive Modeling Continued...

- Generalize data into 3 groups
 - Not listening
 - Neutral
 - Actively Listening
- Sklearn Logistic Regression model to make predictions

```
"input_data": [  
  [80.96, 7.49, 9.37, 0, 0, 0],  
  [41.7, 40.88, 14.89, 1.58, 0, 0],  
  [26.8, 66.92, 3.89, 3.58, 0, 0],  
  [81.34, 15.49, 1.92, 1.058, 0, 0],  
  [88.05, 4.6, 5.88, 0, 0, 0],  
  [32.32, 51.36, 15.95, 0, 0, 0],  
  [66.08, 20.53, 12.46, 0, 0, 0],  
  [76.59, 2.17, 19.599, 0, 0, 0],  
  [81.62, 10.008, 7.91, 0, 0, 0],  
  [98.27, 1.21, 0, 0, 0, 0],  
  [97.115, 0, 1.37, 0, 0, 0],  
  [98.57, 0, 0, 0, 0, 0],  
  [98.37, 0, 0, 0, 0, 0],  
  [63.31, 34.36, 1.99, 0, 0, 0],  
  [72.21, 21.87, 5.1, 0, 0, 0],  
  [93.61, 3.42, 0, 0, 0, 0],  
  [95.93, 2, 1, 0, 0, 0],  
  [98.07, 0, 0, 0, 0, 0],  
  [22.7, 3.9, 63.61, 0, 1, 8.14],  
  [0, 0, 0, 0, 0, 99.2],  
  [78.46, 0, 8, 0, 6.8, 4.2],  
  [2.8, 6.7, 80.2, 0, 0, 9.04],  
  [75.3, 1.3, 5.02, 1.95, 11.83, 3.12],  
  [75, 0, 21.17, 0, 1.8, 0]  
],  
"output_data": [0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2]
```



Predictive Modeling Demo



The screenshot displays a web-based predictive modeling interface. At the top, there is a black header bar, followed by a yellow bar with a red camera icon on the left and a red square on the right. Below this is a purple bar. The main content area is white and contains the following elements:

- A text input field with the value "demo-user" and a label "Userid for data to be trained for".
- A dropdown menu with the selected option "Choose expected output: Activity Listening".
- A "Toggle Training" button.
- A video feed showing a person wearing a headset, with the text "Collecting data..." overlaid on the bottom left of the video.
- A "Dev Tools" button.
- A "Toggle Timer" button.
- A "Capture photo" button.

For the Future

- Improve accuracy of data which would require more time and different types of people
- Would like to analyze live video streaming, but expensive
- Size/scalability (not really good with lots of people)
- A full statistics page for a student to evaluate their reactions