Individual Contributions for Project 2:

- **Design**
    - The whole design and architecture of the application was progressively made based on the ideas of both me and Aakash. My contributions were majorly in the area of providing AWS resources for the purposes of the demo as well as implementation of some features at coding level.
- **Implementation**
    - The application was developed using Python with the help of boto3 libraries which enables the use of AWS resources, ffmpeg library which helps in extracting frames from the video input and lastly face_recognition library which helps in comparing the faces of input from the database to give the match. All of the above was installed in the docker image built using a Dockerfile logic which was pushed to AWS Elastic Container Registry. The AWS Lambda function was then made using the above image with an S3 trigger which would run the application whenever a new input is added to the input S3 bucket. I helped in creating the initial code base as well as a github repository to use as version control. In the handler.py class which is the main class for the application, I implemented the logic to retrieve the video file from the input S3 bucket and download it to the local storage of the docker image. After this I called the ffmpeg library to extract the frame images from the input video which was received and then ran the face_recognition library on the first image to get the image encoding and then matching it with the known image encodings provided in the encoding file. The match was then printed on the console.
- **Testing**
    - I tested the lambda function in different ways on various stages of development and fixed the issues that were encountered. I also performed load testing using the default workload generator which was provided and it took around 5 minutes to complete the whole execution. Unsatisfied with this performance, Aakash made a multi-threaded workload generator for the purpose to test auto-scaling as well as improve performance times. When I used this generator, the output was finished within 2 minutes which is more than 50% faster than the previous performance. With that I concluded that the application is working properly within the proper time limits.