Akhila Chinepalli, ac4454

Anusha Holla, ah3816

Rishabh Gupta, rg3334

Sughosh V Kaushik, svk2117

Project Report: Cloud Computing & Big Data [COMSE6998]

Contents

Project Idea : Video Notebook
(Keyword Search, Notes Bookkeeping & Keyphrases Detection)

Description:
We aim to build a system where users can search for particular keywords in a video and navigate to the part of the video where that concept was discussed. The system also allows the user to make notes while viewing a video that will be associated with that particular time in the video. This allows for an organized way of taking notes and associating them with specific segments of the video. In addition to this, the system would automatically extract a set of key topics from a particular video which would help the user understand the key takeaways from the lecture.

Motivation:
The idea for this project came up with the intent to help students refer to the course material videos without them going through the entire video or finding their way to the part of the video which they are interested in learning. This is highly relevant in current times as the entire world has switched to remote learning. Many courses instruct students over recorded video calls, and those recordings are then available for the students to refer to. In cases where going through the entire video isn't feasible, for example, the night before the exam, our system would help students search for a specific concept in the video and directly jump to that part.

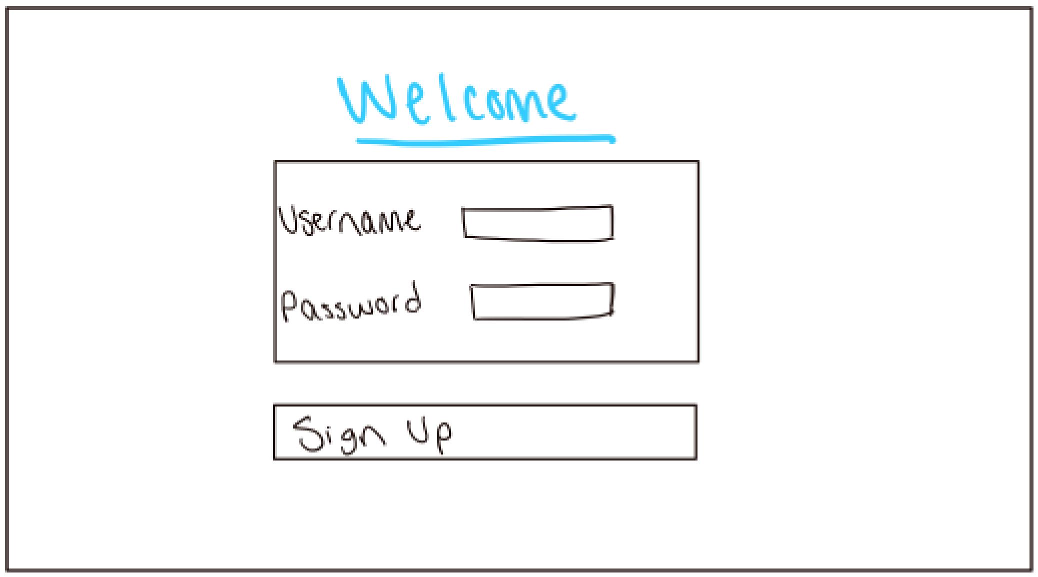
Dataset:
Educational videos will be uploaded by the users using the system.

Feature List:
1. **User App Signup:** The users can sign up for the application using any of the social media accounts.
2. **Video Upload:** The users can upload their course videos.
3. **Keyword Search:** The users can search for any keywords across a video.
4. **Note Taking:** The users can stop the video at any particular time and create a note which is relevant to what is being discussed at that timestamp.
5. **Keyphrase Extraction:** The system would automatically extract and list the keyphrases/topics discussed in a particular video and display where they have been discussed in the video.

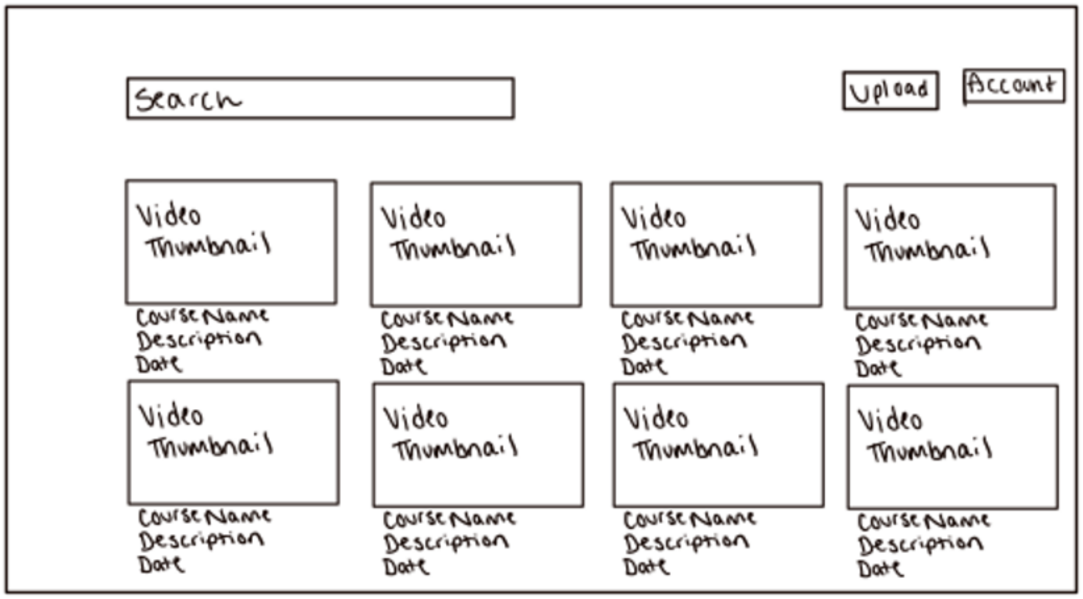
Extensions/Noteworthy Points:
- Our focus is to provide the ability to search anything in a particular video. But if time permits, we would extend the system to search for a keyword throughout the entire dataset of videos that the system possesses.
- The efficacy of the keyphrase extraction is contingent upon the accuracy of the available NLP models which work on this problem such as AWS Comprehend, graph based ranking methods, etc.

Video Notebook Prototype

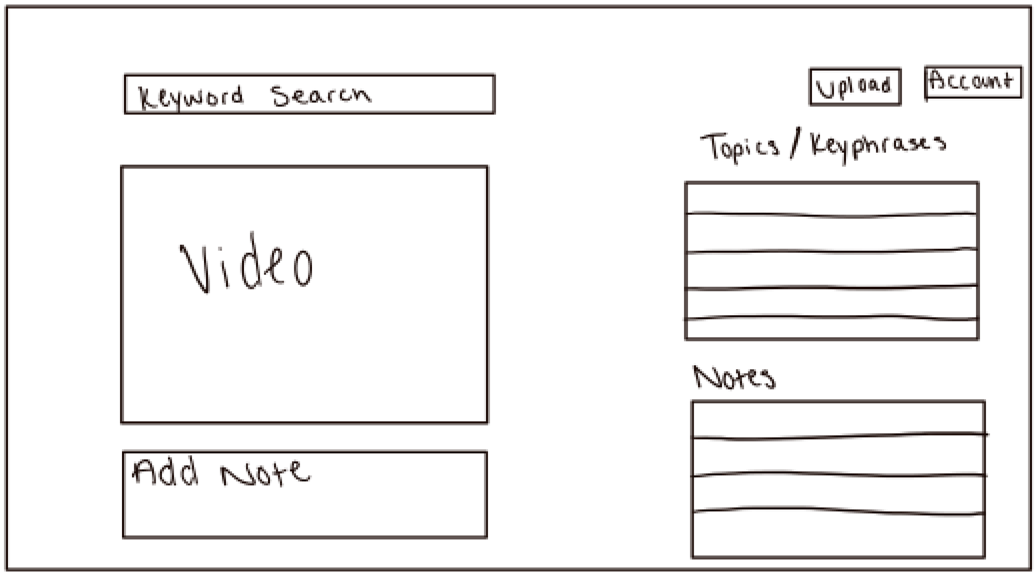
Welcome Page



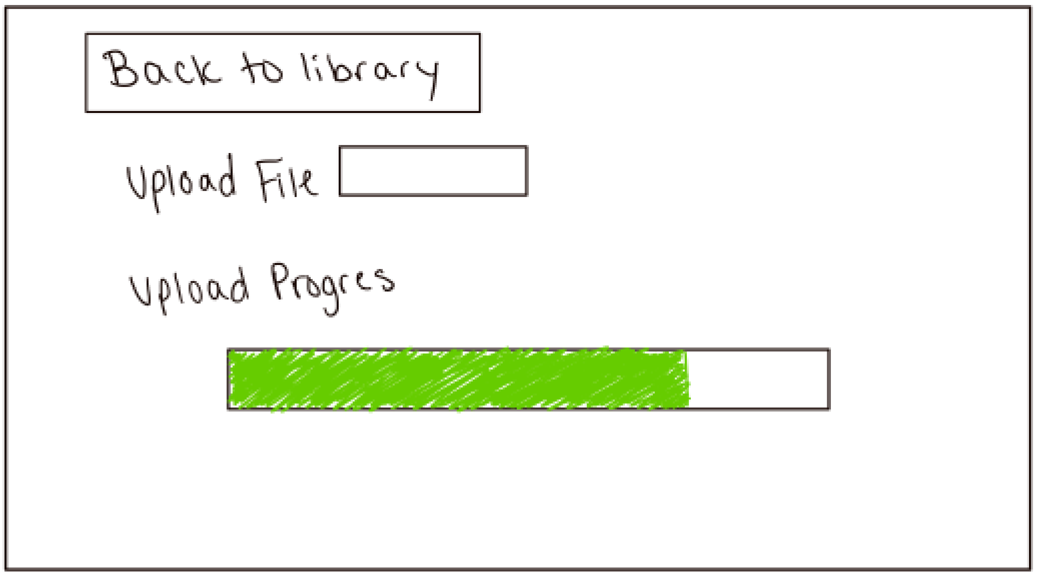
Video Library



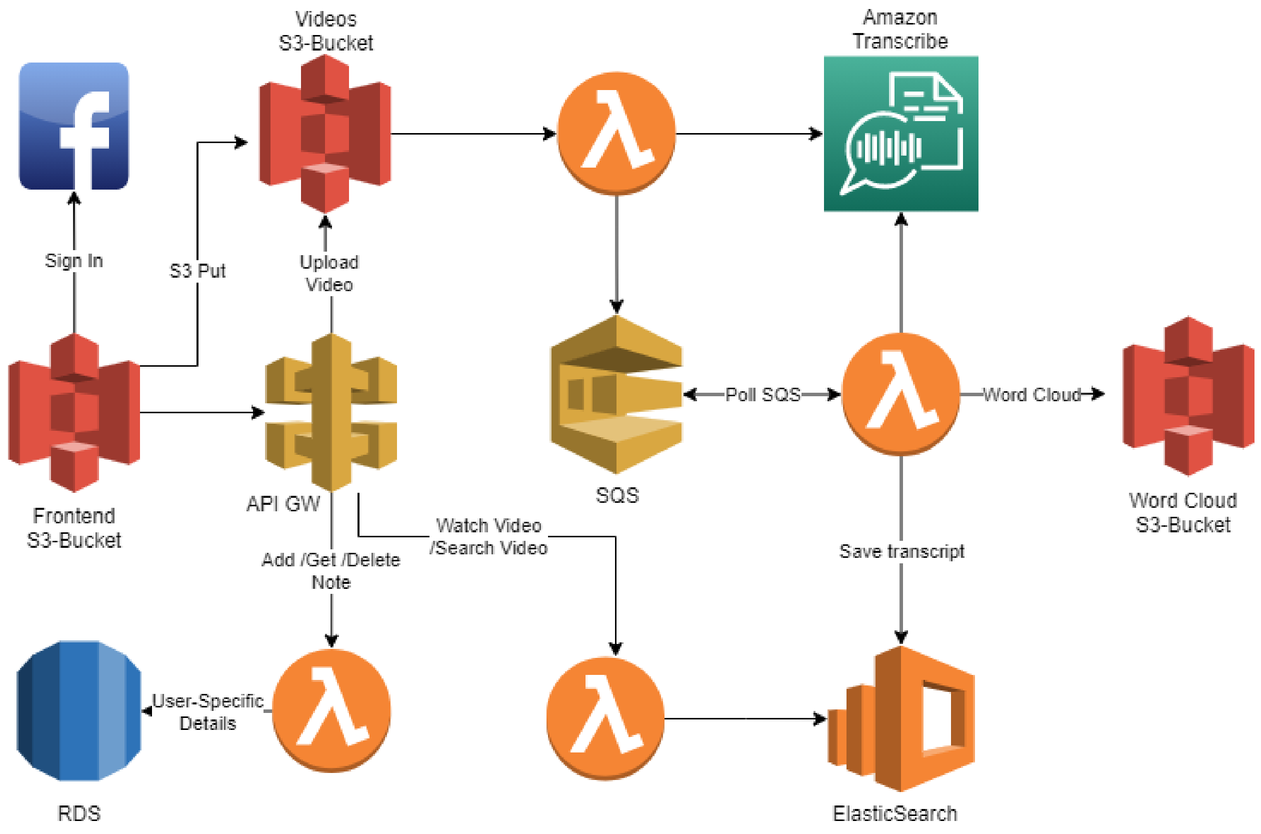
Video



Upload



Architecture Diagram



Architecture Flow

- The user logs in to the application using their Facebook account credentials.
- On successful login, the user is redirected to the home page of the application which lists the videos present in the video library.
- The user can do the following from here:
  - Proceed with the upload of a video.
  - Click on a video and get redirected to the video specific page where the user can perform search on the video transcript and add notes while viewing the video.
- **Upload a video:**
  - The user selects a video and provides a title for the same.
  - Once the video gets uploaded to S3 bucket, a lambda function (*uploadjob-push-to-sqs*) gets triggered (on a PUT event on the bucket) which starts an asynchronous

transcription job using AWS transcribe and pushes the job information to an SQS queue.
- Another lambda function (*uplaodjob-pull-from-sqs*) keeps polling the SQS queue and checks if the transcription jobs lying in the queue have completed or not. On completion of a transcription job, the generated transcript is broken down into sentences and saved in Elasticsearch along with their start time in the *transcripts* index. The same function also creates a word cloud by traversing through the entire transcript and saves it in another S3 bucket. The details of each video (video ID, video URL, video Title and the word cloud URL) are saved in another ES index *video_list*.

- **Watch a video:**
  - The user clicks on a video thumbnail on the home page which redirects them to another page where the user can play the video.
  - The page is rendered by calling the *watch* GET api to retrieve the video specific information like the video URL and Title. At the same time, the *notes* GET api is also hit to retrieve the notes specific to this video that this particular user had created earlier. The *watch* API reads data from Elasticsearch while the *notes* API retrieves data from an AWS Aurora DB instance which is where the notes are saved.
  - The rendered page also displays a word cloud which was created after the transcription for this video gets completed. The word cloud is intended to give an idea to the user about the key points that were talked about in the video.
  - The page provides a search box which the user can use to search across the transcript which is serviced through the *search* GET api. The sentences that match the search query are displayed along with their timestamps so that the user can quickly jump to the relevant section.
  - While viewing the video, the user can choose to create new notes through another text box present on the same page whose request is then forwarded to the *notes* POST api. The user is also given the option to delete an older note which goes through the *notes* DELETE method.
- Lastly, there is a logout button present at the top right corner of each page which the user can use to end their session.

Databases Used

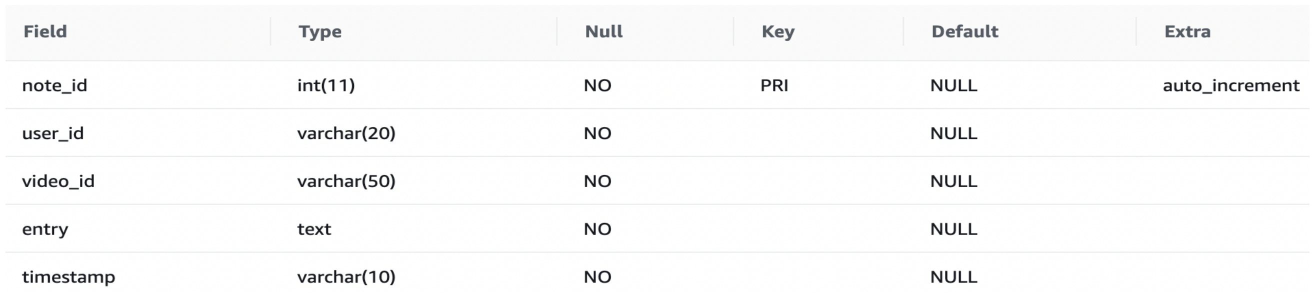
- **Elasticsearch:** Saving video details and transcript

  1. Index: video_list

  Schema:
```
            {
                  "timestamp": { "type": "date"},
                   "video_id": { "type": "text"},
                   "video_title": { "type": "text"},
                   "video_url": { "type": "text"},
                   "wordcloud_url": { "type": "text"}
            }
```
  2. Index: transcripts

  Schema:
```
            {
                  "sentence": {
                        "type": "text",
                        "analyzer": "my_analyzer"
                   },
                   "start_time": { "type": "text"},
                   "video_id": { "type": "text"}
            }
```
  Analyzer Used for Indexing and Search:
```
            "analysis" : {
                  "analyzer" : {
                        "my_analyzer" : {
                              "filter" : [ "lowercase", "stop" ],
                              "tokenizer" : "standard"
                         }
                  }
             }
```
- **AWS RDS Aurora MySQL DB:** Saving the user created notes

  Schema:



Field	Type	Null	Key	Default	Extra
note_id	int(11)	NO	PRI	NULL	auto_increment
user_id	varchar(20)	NO		NULL	
video_id	varchar(50)	NO		NULL	
entry	text	NO		NULL	
timestamp	varchar(10)	NO		NULL	

APIs

1. /: Listing videos in the library
   - GET
2. /watch: Watch a video
   - GET | Query Param: videoID
3. /notes: View/Create/Delete notes
   - GET | Query Params: *videoID, userID*
   - POST | Request Body: *videoID, userID, note, timestamp*
   - DELETE | Query Param: *noteID*
4. /search: Search through a video transcript
   - GET | Query Params: *videoID, searchString*
5. S3 Upload API for uploading videos

Features Implemented

1. User Sign Up/Login via Facebook
2. Video Upload & Subsequent Transcription
3. Keyword Search across Video Transcript
4. Rapid Note-Making corresponding to video timestamps
5. Word Cloud of frequently occurring words