

PROJECT REPORT
ON
ONLINE VIDEO CONVERTER

towards partial fulfillment of the requirement

for the award of degree of

Bachelor Of Computer Application

submitted by

**Bharat Singh Rajput
717042**

under the supervision of

Mr. Rinku Raheja

ACADEMIC SESSION 2017 – 2020

Department Of Computer Science



NATIONAL POST GRADUATE COLLEGE

AN AUTONOMOUS COLLEGE OF UNIVERSITY OF LUCKNOW (NAAC 'A' Grade
Credited)

2 - Rana Pratap Marg, Lucknow, Uttar Pradesh, India.



NATIONAL POST GRADUATE COLLEGE

2 - Rana Pratap Marg, Lucknow, Uttar Pradesh, India.

Certificate

this is to certify that project report entitled

ONLINE VIDEO CONVERTER

being submitted by

**Bharat Singh Rajput
717042**

*towards partial fulfillment of the requirement
for the award of degree of*

Bachelor Of Computer Application

In the Academic Session 2017 - 2020

is a record of students own work
and to the best of my knowledge the work reported here in does not form a part of
any other thesis or work on the basis of which degree or award was conferred on
an earlier occasion to this or any other candidate.

DR. SHALINI LAMBA
Head Of Department

CANDIDATE DECLARATION

I, hereby certify that the work which is being presented in the project work entitled ***Online Video Converter*** is the fulfillment of the requirement for award of the **Bachelor Of Computer Application** submitted in the **Department Of Computer Science of National Post Graduate College**.

This is an authentic record of my work during 5 months period under supervision of **Mr. Rinku Raheja**.

The matter presented in the project work has *not* been submitted by me for the award of any other Degree/Diploma of this or any other University.

Date: / / **2020**

Place: Lucknow, Uttar Pradesh, India.

(Signature Of Candidate)

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the development of any task would be incomplete without the **mention of the people** who make it possible, whose constant guidance and encouragement crowned our efforts with success.

We, the developers feel indebted to a lot of people who during the course of our project not only held us in good stead but also provided us with invaluable services both personal and professional.

At this juncture, when we, at the successful completion of our project are presenting our report it would be unfair not to acknowledge the various acts of kindness done to us by our teachers. They were always present on our every beck and call to help us out of every hurdle and obstacle that we faced.

Our first vote of thanks goes to our **Parents**, with every gesture, every word, every pat on the back, they have always tried to inspire us for better things and attain them. More so with this project, they were with us every instant, sharing our joys - when the code used to work and wiping our tears when machine used to hang.

We wish to place on record our whole hearted gratitude to **Mr. Rinku Raheja** our project guide and **Mr. Amit Srivastava** for making available every facility that we required during the course of our project. Also, we are extremely thankful to you Sir, for extending your kind guidance and words of wisdom from your treasure trove of experience. With friendly advice and guidance at every step, his presence was a welcome sight throughout the project.

We are also deeply indebted to our (Head Of Department) **Dr. Shalini Lamba** for her constant presence, supervision and advice paid off in the culmination of this project and has helped us a great deal with this project with her constant words of encouragement and advice.

Actually, this project report is just an excuse to convey our feelings about how much we appreciate the amount of concern and caring that our teachers exhibit in all our pursuits ranging from anything as simple as the routine lab program to something as taxing as a project. Their patient listening to our problem is itself a major source of strength to us.

Finally, we would like to bind up by paying our heartfelt thanks and prayers to the Almighty, without whose willing nothing is possible in this world and all my dear friends for their support.

Contents

1	Introduction	1
1.1	Abstract Of The Project	1
1.2	Objective Of The Project	1
1.2.1	Problems With Existing System	2
1.2.2	Proposed System	2
1.3	Scope Of The Project	3
1.4	Modules Of The Project	4
1.4.1	Server Side Modules	4
1.4.2	Client-Side Modules	6
2	System Analysis	8
2.1	Software Requirement Specification (SRS)	8
2.2	Risk Analysis	9
3	Preliminary Investigation	11
4	Feasibility Study	11
4.1	Economic Feasibility	11
4.2	Technical Feasibility	12
4.3	Operational Feasibility	12
4.4	Cost Estimation Of The Project	13
5	Technologies Used	16
5.1	Server Side Technology	16
5.2	Client Side Technology	18
6	Hardware And Software Requirements	19
6.1	Requirement Specification	19
6.2	Tools / Platform / Software and Hardware Specifications	19
6.2.1	Software Requirements	19
6.2.2	Hardware Requirements	20
7	System Design	21
7.1	Sequential Phases in the Waterfall Model	21
8	Project Planning	22
8.1	Gantt Chart	22
8.2	Data Flow Diagram	23
8.2.1	Context Level Data Flow Diagram	25
8.2.2	Level-1 Data Flow Diagram	26
8.3	Entity-Relationship Diagram	27
9	Development And Coding	30
9.1	Directory Structure	30
9.2	Client-Side Code	31
9.3	Server-Side Code	93
10	Screenshots	126

11 Implementation	135
11.1 Installation of Hardware and Software	136
11.2 Conversion of Data	136
11.3 System Training	136
12 Testing	136
12.1 Unit Testing	136
12.2 Integration Testing	137
12.3 Program Testing	137
12.4 System Testing	137
13 System Security Measures	138
14 Future Scope	140
14.1 Limitations	140
14.2 Future Scope Of The Project	140
15 Bibliography	141
16 Glossary	141

1 Introduction

The name of my project is **Online Video Converter**.

As It's name says, it is a media converter application for Videos. It is a web application that can be accessed from any device having a modern operating system and capable of handling websites. It offers a clean and easy to use user interface that facilitates users to convert their videos.

Users can login to their account, upload new videos which will be saved on the web application as long as they like and can convert, download and stream them on the fly.

This web application, thus work as a cloud storage for the videos and a conversion tool at the same time. This application is also capable of streaming videos and therefore, users can also watch their videos when and wherever they want.

1.1 Abstract Of The Project

There are many times when we have to change the attributes of a video such as its resolution, it's format or compress it to decrease it's size. There are specialized software built to perform this task but these software are platform dependent. Therefore, a conversion tool built for one operating system cannot work on the other operating system.

Keeping this drawback in mind, I have developed a web based solution to this problem. As we all know, when an application is a "web application", it becomes platform agnostic. Since web can be accessed from any platform, therefore, this conversion tool can work on any platform that supports a modern web browser. Apart from being a platform independent conversion tool, this web application also work as a cloud storage and as a streaming platform for videos.

1.2 Objective Of The Project

This project aims to provide a platform agnostic application for the users to convert their videos, store them in a secure manner, and get additional information regarding their videos, and stream then when and wherever they want **on any device** of their choice.

Since there are numerous formats and resolution available for the videos, this application aims to support uploading and conversion of most commonly used video

formats such as .mp4, .avi, .flv and so on. The applications aims to provide a user friendly UI that can be easily understood.

1.2.1 Problems With Existing System

- Video conversion tool are often in the form of binaries complied to their specific platforms and therefore, there is not a single tool that can be used accross all platforms.
- Traditional video conversion solutions are often complex and hard to understand. Their user interface is full of an overwhelming amount of details and complexity.
- Traditional video conversion does not provide any cloud storage for videos. They can only be used offline and therefore, on slow computers such as handheld devices, there is no effecting and time friendly tool to convert videos.
- Traditional video conversion uses the client's resources for video conversion.
- Traditional video conversion does not provide any concept of streaming videos and cloud storage.

1.2.2 Proposed System

- Online Video Converter, being a web application, online media converter is platform agnostic. The same application can be used on any device whether it's a powerful computer or a handheld device such as a mobile phone.
- Online Video Converter is easy to use and understand. The **vuetify plugin**¹ used as a user interface provides a material design that is beautiful to look at, along with providing a smooth transition. The user interface is designed in such a way that the user can easily understand the application. Necessary tooltips are also provided wherever they are needed.
- Online Video Converter does provide cloud storage for videos and therefore, the videos once uploaded will be available to the user whenever they want.
- Online Video Converter uses the server's resources for video conversion and therefore, videos are converted quickly using a powerful server-side computer. Using this approach even large videos can be converted within minutes even on a small handheld device.

¹<https://vuetifyjs.com/>

- Online Video Converter also provides the facility of streaming service which enables the users to watch their videos once they are uploaded to the server.

1.3 Scope Of The Project

Online Video Converter is designed for the particular need of the company to carry out operations smoothly and effectively and will prove itself to be a powerful tool for users to convert their videos with ease. The project aims to be a platform-agnostic solution for users to convert, store, and stream their videos.

- The system aims to major video formats such as *.mp4, *.avi, *.mov, *.flv, and so on.
- The system aims to shift the load of video conversion from the client to the server.
- The system aims to be easy to operate.
- The system aims to satisfy the user requirement.
- The system aims to be delivered on schedule within the budget.
- The system aims to usable on any platform and on the most common screen sizes.

1.4 Modules Of The Project

1.4.1 Server Side Modules

- **Users**

The user's module on the server-side is REST² API³ that provides services for the frontend application to perform CRUD operations on the user's *collection*⁴. The modules support HTTP operations to POST, PATCH, and GET the data from the user's collection. However, DELETE requests are blocked, therefore, the frontend application cannot delete the user's account, and also all requests are only restricted to the current user therefore, users can't access the details or search other users on the server.

- **Uploads**

Uploads are videos that users upload on the server. The uploads' module on the server-side is a REST API that provides services for the frontend application to perform CRUD operations on the user's upload collection. The modules support HTTP operations to POST, GET, and DELETE data from the uploads' collection. However, PATCH requests are blocked, therefore, the frontend application cannot make changes to video information once they are uploaded on the server, and also all requests are only restricted to the current user therefore, users can't access the uploads of other users on the server.

- **Convert**

Convert are videos that users convert on the server using their original uploads. The converted videos are also available on the server as long as the users want. The 'convert' module on the server-side is a REST API that provides services for the frontend application to perform CRUD operations on the covert collection and also to request server to perform conversion. The modules support HTTP operations to POST, GET, and DELETE data from the convert collection. However, PATCH requests are blocked, therefore, the frontend application cannot make changes to converted video information once they are converted by the server, and also all requests are only restricted to the current user and therefore, users can't access the converted videos of other users on the server.

²Representational State Transfer - https://en.wikipedia.org/wiki/Representational_state_transfer

³Application Programming Interface - <https://en.wikipedia.org/wiki/API>

⁴<https://docs.mongodb.com/manual/core/databases-and-collections/>

- **Streams**

Streams service allows users to stream their uploaded videos. It does not support any database operations. The 'stream' module on the server-side is a REST API that provides services for the frontend application to perform GET operation on video files uploaded by the user. The modules support only HTTP GET requests to fetch video streaming data from the server. This data is then processed by the frontend application and played using video player of their choice.

- **Thumbs**

Thumbs service allows users to request thumbnails of their uploaded videos. The thumbnails provide visual assistance when the user is trimming a video. It does not support any database operations. The 'thumbs' module on the server-side is a REST API that provides services for the frontend application to perform GET operation on thumbnail files generated by the server when users upload their videos. The modules support only HTTP GET requests to fetch thumbnail data from the server. This data is then processed by the frontend application and displayed as a reference image to the user.

- **AuthManagement**

AuthManagement service manages the e-mails sent to the user. When ever the user registers on the web application, a confirmation email is sent to the user. And also if the user forgets his/her password or if he/she wants to change his/her password, a password reset e-mail is sent to the user to confirm his/her identity. All these operations are managed by the 'authmanagement' service. It does not support any database operations but it does add tokens to the user's collection in the background. The 'authmanagement' module on the server-side is a REST API that provides services for the frontend application to perform POST operations for various activities such as confirm registration,request password change e-mail, and change user's password. The modules support only HTTP POST requests.

- **Authentication**

The Authentication module manages the authentication of the user. It uses token-based authentication i.e., JWT⁵ (JSON Web Tokens) to authenticate users before they use any services. It also acts as a session management service. The 'authentication' module on the server-side is a REST API provides services for the frontend application to perform HTTP operations such as

⁵https://en.wikipedia.org/wiki/JSON_Web-Token

POST and DELETE to create token when logging in, get token after successful login, and delete token when users are logging out.

1.4.2 Client-Side Modules

- **App.vue**

App.vue is a container module for all other modules. It contains a header, navigation bar, footer, side-panel to quickly jump from one page to another.

- **Login.vue**

Login.vue is a frontend module that manages user login. It provides a user interface for the user to perform operations such as logging in and request password change e-mail in case he forgets his/her password.

- **SignUp.vue**

SignUp.vue is a frontend module that manages user sign up. It provides a user interface for the user to perform sign up operation. It validates all the data before sign up.

- **Account.vue**

Account.vue is a frontend module that manages the user's information. It provides a user interface for the user to perform changes in his information and requests a password e-mail. It validates all the data before changing it. Users can also view their profiles using this module.

- **Videos.vue**

Videos.vue is the main module. It is a frontend module that manages user's videos and it's conversion. It provides a user interface for the user to perform operations such as:

- Upload a new video.
- Stream a uploaded video and directly play it on the browser.
- View information about the uploaded video such as it's resolution, frame rate, bitrate, size and so on.
- Open the conversion dialog in order to perform conversion.
- Delete a uploaded video.

- Search a uploaded video by date or a search string or both.
- Change the view to either List View or Card View.
- **Video Conversion Dialog**

This dialog provides various parameters to tweak when a user is performing the conversion. These parameters include file format, video quality, trim duration, and so on. Users can initiate conversion using this dialog.

- **ConversionHistory.vue**

ConversionHistory.vue is also one of the main module. It is a frontend module that manages user's already converted videos. It provides a user interface for the user to perform operations such as:

- List all converted videos.
- Download a converted video.
- View information about the uploaded video such as it's resolution, frame rate, bitrate, size and so on.
- Delete a converted video.
- Search a uploaded video by date or a search string or both.
- Change the view to either List View or Card View.

- **Reset.vue**

Reset.vue is a frontend module that provides a user interface for the user to change their password. It validates the password before changing it.

- **Verify.vue**

Reset.vue is a frontend module that verifies user's e-mail after their signup. It is only useful for unverified users.

2 System Analysis

2.1 Software Requirement Specification (SRS)

The SRS is produced at the culmination of the analysis task. The function and performance allocated to software as a part of system engineering are refined by establishing a complete information description, a detailed functional and behavioral description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements.

Identification of need

- Traditional conversion tool are often in the form of binaries compiled to their specific platforms and therefore, there is not a single tool that can be used accross all platforms.
- Traditional video conversion solutions are often complex and hard to understand. Their user interface is full of an overwhelming amount of details and complexity.
- Traditional video conversion does not provide any cloud storage for videos. They can only be used offline and therefore, on slow computers such as handheld devices, there is no effecting and time friendly tool to convert videos.
- Traditional video conversion uses the client's resources for video conversion.
- Traditional video conversion does not provide any concept of streaming videos and cloud storage.

Proposed Requirements

- The proposed system should be platform agnostic. The same application can be used on any device whether it's a powerful computer or a handheld device such as a mobile phone.
- The proposed system should be is easy to use and understand. The user interface should be designed in such a way that the user can easily understand the application. Necessary tooltips should also be provided wherever they are needed.
- The proposed system should provide cloud storage for videos and therefore, the videos once uploaded will be available to the user whenever they want.

- The proposed system should use the server’s resources for video conversion and therefore, the videos can be converted quickly using a powerful server-side computer. Using this approach even large videos can be converted within minutes even on a small handheld device.
- The proposed system should also provide the facility of streaming service which enables the users to watch their videos once they are uploaded to the server.

Interface Requirements

- User Interface elements must be easy to understand.
- The user interface should be easy to learn. When the user uses the interface, they should know which element is used for which operation.
- The interface actions and elements should be consistent. When users press any button, required actions should be performed by the system along with the appropriate response to the user.
- The screen layout and accessibility should be clean and minimalistic.
- Understandability and vision of layout should be appealing and simple.

2.2 Risk Analysis

Uncertainty, which is constantly present in our daily lives, frequently impacts our decisions and actions. When we talk about risk, we normally mean the chance that some undesirable impact will occur. Hence, we normally seek to avoid or minimize risk. If there is a chance of rain, and we don’t want to get wet, we may choose to stay indoors “avoiding that risk” or we may take an umbrella to minimize the impact of rain upon us. Uncertainty can impact our decisions and actions in desirable as well as undesirable ways. In risk analysis, we usually focus on what can go wrong “the outcomes that represent loss or damage” although an effective analysis will also help us understand what can go right as well.

A risk assessment involves evaluating existing physical and environmental security and controls and accessing their adequacy relative to the potential threats of the organization. A system impact analysis involves identifying the critical system functions within the organization and determining the impact of not performing the business function beyond the maximum acceptable outage. Types of criteria that can be used to evaluate the impact include customer service, internal operations, legal/statutory, and financial.

A primary objective of business recovery planning is to protect the organization if all or part of its operations and/or computer services is rendered unusable. Each functional area of the organization should be analyzed to determine the potential risk and impact related to various disaster threats.

Regardless of the prevention techniques employed, possible threats that could arise inside or outside the organization need to be assessed. Although the exact nature of potential disasters or their resulting consequences is difficult to determine, it is beneficial to perform a comprehensive risk assessment of all threats that can realistically occur to the organization. Regardless of the type of threat, the goals of business recovery planning are to ensure the safety of users and the system.

Uncertainty can arise in several ways:

- The user's video might not be supported by the server.
- The server might be busy in a lot of conversion requests and may not be able to accept new requests.
- Unavailability of the internet.
- Unavailability of the resources.
- Data corruption.

3 Preliminary Investigation

A preliminary system study is the first stage of the system development life cycle. This is a brief investigation of the system under consideration and gives a clear picture of what the physical system is? In this stage, the need for the system is evaluated. The initial system study involves the preparation of a "Project Overview" which lists the Problem with the existing system, Objectives of the proposed system, Scope of the proposed System, Constraints, Expected benefits of the new system, etc. in the light of the user requirements. The main steps followed in this stage are:

- Problem identification and project initiation.
- Background analysis.
- Inference or findings (system proposal).
- Determine the size of the project.
- Assess the costs and benefits of approaches.
- Determine the technical feasibility of approaches.

4 Feasibility Study

After understanding the project, study, and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible – given unlimited resources and infinite time. This also includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on future upcoming requirements.

4.1 Economic Feasibility

This is an important aspect to be considered while developing a project. We decided the technology based on the minimum possible cost factor.

- All hardware and software cost has to be borne by the organization.
- Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome and the later on running cost for system

Here, in Online Video Converter; all elements are under defined economic constraints used for its development lie within the budget estimated for its development.

4.2 Technical Feasibility

This included the study of function, performance, and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS) and checked if everything was possible using a different type of frontend and backend platform.

For Online Video Converter, following technical needs of the system may include:

- The facility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to produce a certain volume of transactions at a particular speed.
- Facility to communicate data to a distinct location.

Here, following hardware and software requirements must be fulfilled:

Database Design	Mongo DB v4.0.4+
Frontend Design	Vue.js along with Vuetify and feathers-vuex
Backend Design	Node.js along with express.js and feathers.js

Specific software and hardware products can then be evaluated keeping in view with the logical needs. The “Online Video Converter” is technically feasible as all the software and hardware requirements are met by the organization.

4.3 Operational Feasibility

No doubt the proposed system is fully GUI based which is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with the new system. As far as our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

The “Online Video Converter” is found to be feasible operationally because it is designed in such an interactive manner that users need not take any special training for operating the website.

4.4 Cost Estimation Of The Project

Software cost comprises a small percentage of the overall computer-based system cost. There are several factors, which are considered, that can affect the ultimate cost of the software such as – human, technical, Hardware and Software availability, etc. The main point that was considered during the cost estimation of **Online Video Converter** was its sizing. In spite of complete software sizing, function point and approximate lines of code were also used to size each element of the Software and their costing. The cost estimation is done by me for Project also depends upon the baseline metrics collected from past projects and these were used in conjunction with estimation variables to develop cost and effort projections.

We have basically estimated this project mainly on two bases:

- **Effort Estimation**

This refers to the total man-hours required for the development of the project. It even includes the time required for doing documentation and user manual.

- **Hardware Required**

This includes the cost of the computer and the hardware cost required for the development of this project.

Tools and methods used in the development of this project are free to use for noncommercial purposes. So, the development cost is negligible. Basic COCOMO⁶(Constructive Cost Mode) computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of source lines of code (SLOC⁷, KLOC⁸).

COCOMO applies to three classes of software projects:

- **Organic projects**

Small teams with good experience working with less than rigid requirements.

- **Semi-Detached Projects**

Moderate sized teams with mixed experience working with a mixture of rigid and less than rigid requirements.

- **Embedded Projects**

Developed within a set of *tight* constraints. It is also combination of organic and semi-detached projects.

⁶<https://en.wikipedia.org/wiki/COCOMO>

⁷Source lines of code

⁸Thousand lines of code

The basic COCOMO equations takes the form:

$$\text{Effort Applied}(E) = a_b.(KLOC).b_b \text{ [man-month]}$$

$$\text{Development Time}(D) = c_b.(\text{Effort Applied}).d_b \text{ [months]}$$

$$\text{People Required}(P) = \text{Effort Applied} / \text{Development Time}$$

Where, KLOC is the estimated number of delivered lines (expressed in thousands) of code for project. The coefficients a_b , b_b , c_b and d_b are given in the following table⁹.

Basic COCOMO is good for a quick estimate of software costs. However, it does not account for differences in hardware constraints, personnel quality, and experience, use of modern tools and techniques, and so on.

Estimating The Cost Of The Project

Since this is an organic project, we'll take the respective values for the coefficients:

$$\text{Lines Of Code}(LOC) = 5042$$

Therefore,

$$KLOC = 5042/1000$$

$$KLOC = 5.042$$

Taking the values of the coefficients as:

a_b	a_b	a_b	a_b
2.4	1.05	2.5	0.38

• Effort Applied

$$\text{Effort Applied}(E) = a_b.(KLOC).b_b \text{ [man-month]}$$

$$E = 2.4(5.042)^{1.05}$$

$$E = 2.4(5.546679)$$

$$E = 13.120313 \text{ [man-month]}$$

⁹The values listed here are from the original analysis, with a modern reanalysis producing different values.

- **Development Time**

$$\text{Development Time}(D) = c_b \cdot (\text{Effort Applied}) \cdot d_b \text{ [months]}$$

$$D = 2.5(13.120312)^{0.38}$$

$$D = 2.5(2.65961)$$

$$D = 6.65 \text{ [months]}$$

- **People Required**

$$\text{People Required}(P) = \text{Effort Applied} / \text{Development Time}$$

$$P = 13.120313/6.65$$

$$P \approx 2$$

5 Technologies Used

5.1 Server Side Technology

node.js



Node.js is an open-source, cross-platform, JavaScript runtime environment (Framework) that executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user’s web browser. Consequently, Node.js represents a ”JavaScript everywhere” paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.

Though .js is the standard filename extension for JavaScript code, the name ”Node.js” doesn’t refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

express.js



ExpressJS is a web application framework that provides you with a simple API to build websites, web apps and back ends. With ExpressJS, you need not worry about low level protocols, processes, etc.

Express provides a minimal interface to build our applications. It provides us the tools that are required to build our app. It is flexible as there are numerous modules available on npm, which can be directly plugged into Express.

Express was developed by **TJ Holowaychuk** and is maintained by the Node.js foundation and numerous open source contributors.

mongoDB



MongoDB is a document-oriented NoSQL database used for high volume data storage. Instead of using tables and rows as in the traditional relational databases, MongoDB makes use of collections and documents. Documents consist of key-value pairs which are the basic unit of data in MongoDB. Collections contain sets of docu-

ments and function which is the equivalent of relational database tables. MongoDB is a database which came into light around the mid-2000s.

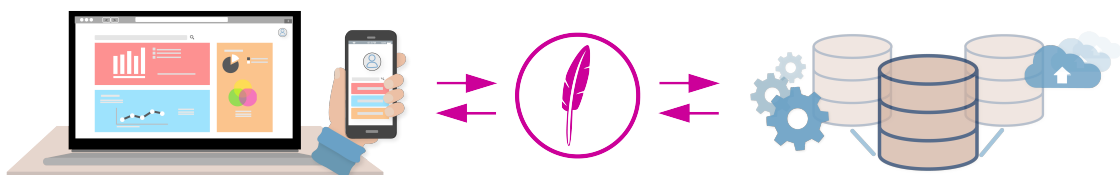
Feature Of MongoDB

- Each database contains collections which in turn contains documents. Each document can be different with a varying number of fields. The size and content of each document can be different from each other.
- The document structure is more in line with how developers construct their classes and objects in their respective programming languages. Developers will often say that their classes are not rows and columns but have a clear structure with key-value pairs.
- The rows (or documents as called in MongoDB) doesn't need to have a schema defined beforehand. Instead, the fields can be created on the fly.
- The data model available within MongoDB allows you to represent hierarchical relationships, to store arrays, and other more complex structures more easily.

feathers.js



Feathers is a lightweight web-framework for creating real-time applications and REST APIs using JavaScript or TypeScript. Feathers can interact with any backend technology, supports over a dozen databases and works with any frontend technology like React, VueJS, Angular, React Native, Android or iOS. At its core, Feathers is a set of tools and an architecture pattern that make it easy to create scalable REST APIs and real-time applications. You can build prototypes in minutes and production-ready apps in days.



ffmpeg



FFmpeg is a free and open-source software project consisting of a large suite of libraries and programs for handling video, audio, and other multimedia files and streams. At its core is the FFmpeg program itself, designed for command-line-based

processing of video and audio files, and widely used for format transcoding, basic editing (trimming and concatenation), video scaling, video post-production effects, and standards compliance (SMPTE, ITU).

FFmpeg includes libavcodec, an audio/video codec library used by many commercial and free software products, libavformat (Lavf), an audio/video container mux and demux library, and the core ffmpeg command-line program for transcoding multimedia files.

FFmpeg is part of the workflow of hundreds of other software projects, and its libraries are a core part of software media players such as VLC, and has been included in core processing for YouTube and iTunes. Codecs for the encoding and/or decoding of most audio and video file formats is included, making it highly useful for the transcoding of common and uncommon media files into a single common format.

The name of the project is inspired by the MPEG video standards group, together with "FF" for "fast forward". The logo uses a zigzag pattern that shows how MPEG video codecs handle entropy encoding.

5.2 Client Side Technology

vue.js



Vue (pronounced like view) is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries.

vuetsify.js



Vuetsify is the 1 component library for Vue.js and has been in active development since 2016. The goal of the project is to provide users with everything that is needed to build rich and engaging web applications using the Material Design specification. It accomplishes that with a consistent update cycle, Long-term Support (LTS) for previous versions, responsive community engagement, a vast ecosystem of resources and a dedication to quality components.

feathers-vuex



Feathers-Vuex is a first class integration of FeathersJS and Vuex. It implements many Redux best practices under the hood, eliminates a lot of boilerplate code with flexible data modeling, and still allows you to easily customize the Vuex store.

6 Hardware And Software Requirements

6.1 Requirement Specification

The system should have a browser that can handle a modern web applications. The detailed requirements are given in the next section which specifies the minimal requirement to run the product.

6.2 Tools / Platform / Software and Hardware Specifications

6.2.1 Software Requirements

Server

Operating System	Windows 7 Service Pack 1 Or Higher
Database	Mongo DB v4.0.4+
Server	Node.js (feathers.js + Express.js) along with ffmpeg

Client

Operating System	Windows 7+, Android, Any GUI Linux Distro, etc.
Web Browser	Google Chrome 60+, Edge, Mozilla Firefox, Safari, etc.
Display	800x600 or higher

Developer

Operating System	Windows 7 Service Pack 1 Or Higher
Database	Mongo DB v4.0.4+
Server	Node.js (feathers.js + Express.js) along with ffmpeg
Display	1024 x 768 32-Bit Color Depth or higher
IDE	Microsoft Visual Studio Code
Version Control	Git
Documentation	Latex

6.2.2 Hardware Requirements

Server

Processor	Quad Core Multi-threaded CPU or better
RAM	8 GB+
Hard Disk Space	80GB for OS + server and additional storage for cloud

Client

Processor	Any Processor Capable Of Handling Modern Website.
RAM	1 GB+
Display	800x600 or higher
Hard Disk Space	Storage for OS And Videos

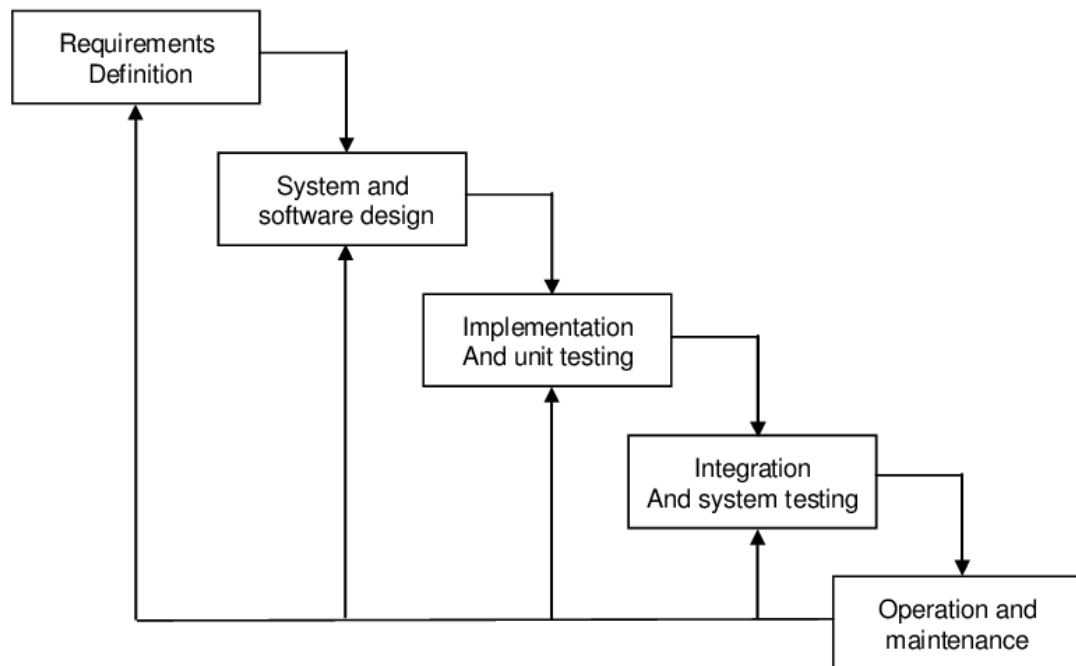
Developer

Processor	Quad Core Multi-threaded CPU or better
RAM	8GB+
Hard Disk Space	500GB for OS and server, IDEs and cloud testing
Display	1080p Display. More than one display preferred

7 System Design

To solve actual problems in an industry setting, a software engineer or a team of Engineers must incorporate a development strategy that encompasses the process, methods, and tools layers. This strategy is often referred to as a process model or a software engineering paradigm. A process model or a software engineering is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required.

In this project, **Waterfall Model** is used which involve steps given below:



In “The Waterfall” approach, the whole process of software development is divided into separate phases. The outcome of one phase acts as the input for the next phase sequentially. This means that any phase in the development process begins only if the previous phase is complete. The waterfall model is a sequential design process in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Production/Implementation, and Maintenance.

As the Waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a **Linear-Sequential Life Cycle Model**.

7.1 Sequential Phases in the Waterfall Model

- Requirements

The first phase involves understanding what needs to design and what is its function, purpose, etc. Here, the specifications of the input and output or the final product are studied and marked.

- **System Design**

The requirement specifications from the first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The software code to be written in the next stage is created now.

- **Implementation**

With inputs from system design, the system is first developed in small programs called units, which are integrated into the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

- **Integration and Testing**

All the units developed in the implementation phase are integrated into a system after testing of each unit. The software designed, needs to go through constant software testing to find out if there are any flaws or errors. Testing is done so that the client does not face any problem during the installation of the software.

- **Deployment of System**

Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market.

- **Maintenance**

This step occurs after installation, and involves making modifications to the system or an individual component to alter attributes or improve performance. These modifications arise either due to change requests initiated by the customer, or defects uncovered during live use of the system. The client is provided with regular maintenance and support for the developed software.

8 Project Planning

8.1 Gantt Chart

An elementary Gantt chart or timeline chart for the development plan is given below:

	December 19	January 20	February 20	March 20	April 20	May 20	June 20
<i>Requirement Specification And Analysis</i>	20 Dec - 31 Dec						
<i>Designing The System</i>		1 Jan - 31 Jan					
<i>Coding</i>			1 February - 31 March				
<i>Implementation</i>				1 Mar - 31 Mar			
<i>Unit Testing</i>					1 Apr - 30 Apr		
<i>Integration Testing</i>					1 Apr - 30 Apr		
<i>System Testing</i>					1 Apr - 30 Apr		
<i>Documentation</i>						1 May - 31 May	
<i>Maintenance</i>							1 June - Now

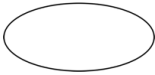



8.2 Data Flow Diagram

The data Flow diagram is a graphical representation of the flow of data throughout the information system. Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. On a DFD, data items flow from an external

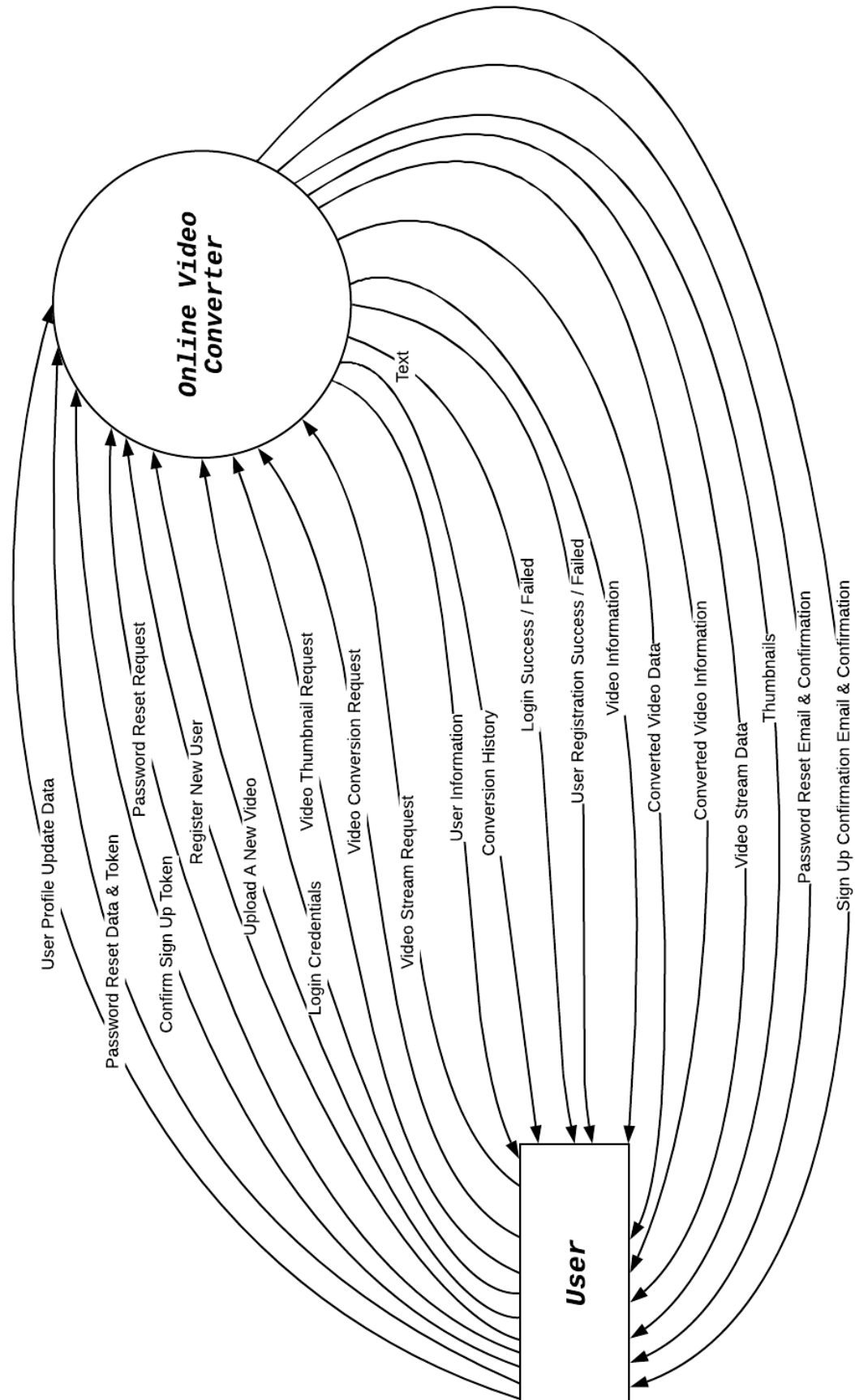
data source or an internal data store to an internal data store or an external data sink, via an internal process.

A DFD provides no information about the timing of processes, or about whether processes will operate in sequence or parallel. It is therefore quite different from a flowchart, which shows the flow of control through an algorithm, allowing a reader to determine what operations will be performed, in what order, and under what circumstances, but not what kinds of data will be input to and output from the system, nor where the data will come from and go to, nor where the data will be stored.

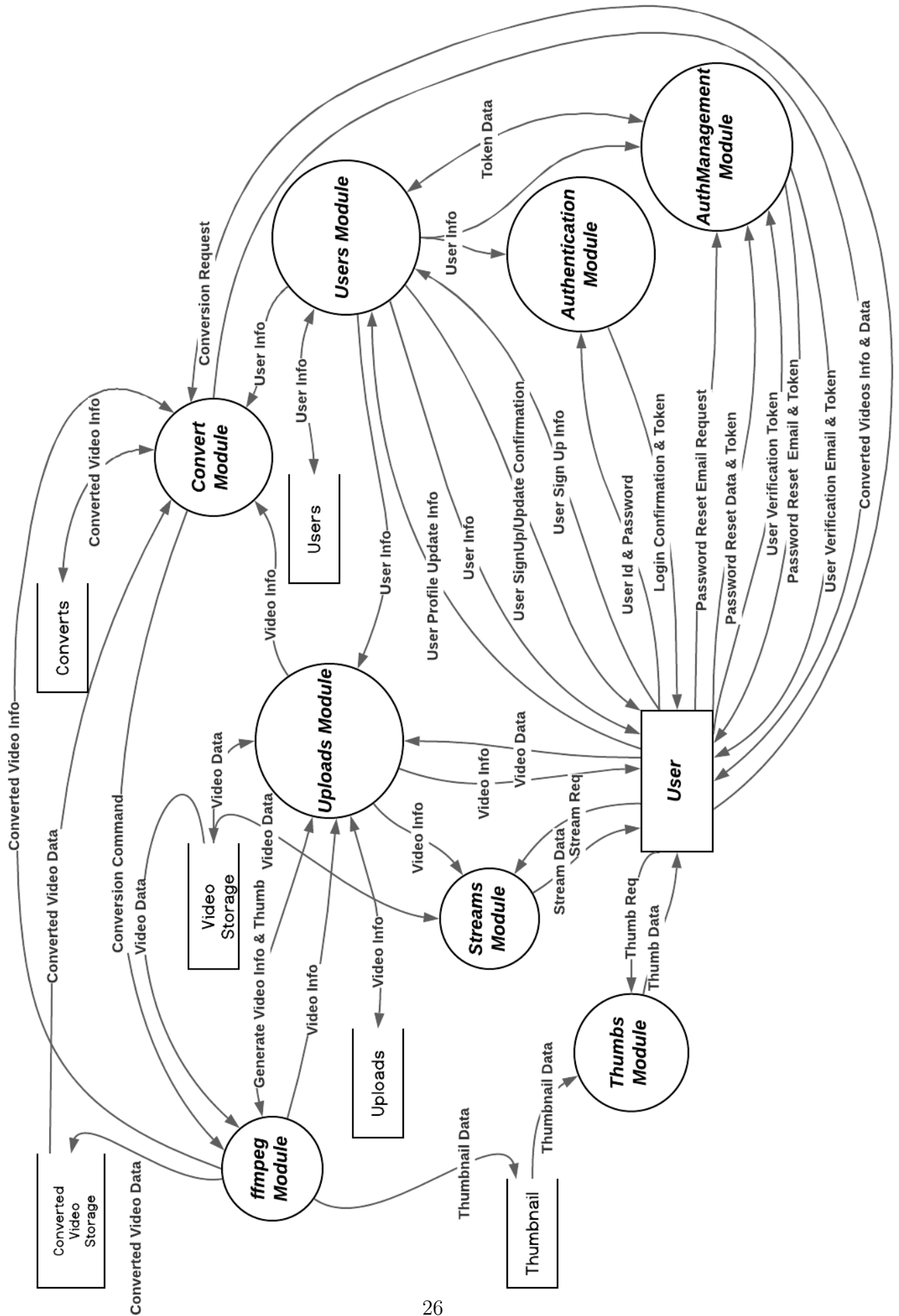
The following notations are used to represent the operational flow in a system -

Name	Notation	Role
Process		Transforms incoming data flow to output data flow
Store		Repositories of data in the system
Dataflow		Dataflow are pipelines through which packets of information flow
External Entity		External entities are objects outside the system, with which the system communicates.

8.2.1 Context Level Data Flow Diagram



8.2.2 Level-1 Data Flow Diagram



8.3 Entity-Relationship Diagram

An entity-relationship diagram describes interrelated things of interest in a specific domain of knowledge. A basic ER model consists of entity types and specifies relationships that can exist between entities.

An ER diagram usually consists of the following:

- **Entity**

Entities are represented through rectangles. Rectangles are named with the entity set they represent.

- **Attributes**

Attributes are the properties of entities. Attributes are represented through ellipses. Every ellipse represents one attribute and is directly connected to its entity (rectangle). If the attributes are *composite*, they are further divided into a tree-like structure. Every node is then connected to its attribute. That is, composite attributes are represented by ellipses that are connected with an ellipse.

- **Multivalued** attributes are depicted by double ellipses.
- **Derived** attributes are depicted by dashed ellipse.

- **Relationship**

Relationships are represented by a diamond-shaped box. The name of the relationship is written inside the diamond-box. All the entities (rectangles) participating in a relationship, are connected to it by a line.

- **Binary Relationship and Cardinality**

A relationship where two entities are participating is called a binary relationship. Cardinality is the number of instances of an entity from a relation that can be associated with the relation.

- * **One-To-One:** When only one instance of an entity is associated with the relationship, it is marked as '1:1'. The following image reflects that only one instance of each entity should be associated with the relationship. It depicts a one-to-one relationship.

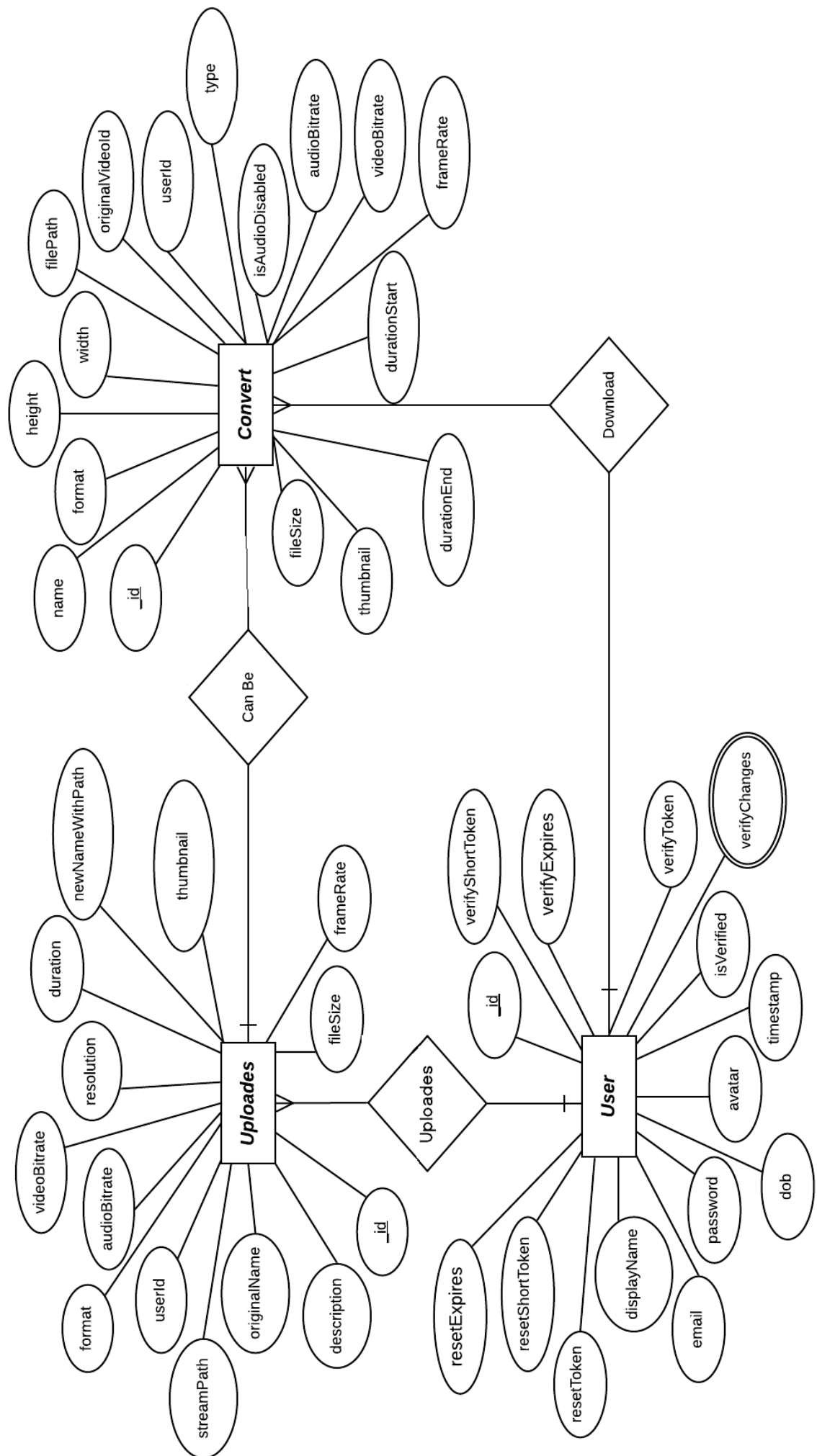


- * **Many-To-One:** When more than one instance of an entity is associated with the relationship, it is marked as 'N:1'. The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship. It depicts a many-to-one relationship.



- * **Many-To-Many:** The following image reflects that more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship. It depicts a many-to-many relationship.





9 Development And Coding

9.1 Directory Structure

Client-Side Directory Structure

```
/
├── public
│   ├── uploads
│   ├── favicon.ico
│   └── index.html
├── src
│   ├── assets
│   ├── components
│   │   ├── Dialog.vue
│   │   ├── Loading.vue
│   │   └── YesNoDialog.vue
│   ├── plugins
│   │   └── vuetyfy.js
│   ├── router
│   │   └── index.js
│   ├── store
│   │   ├── services
│   │   │   ├── authmanagement.js
│   │   │   ├── convert.js
│   │   │   ├── thumbs.js
│   │   │   ├── uploads.js
│   │   │   └── users.js
│   │   └── index.js
│   │   └── store.auth.js
│   └── views
│       ├── Account.vue
│       ├── ConversionHistory.vue
│       ├── Login.vue
│       ├── Reset.vue
│       ├── SignUp.vue
│       ├── texput.log
│       ├── Verify.vue
│       └── Videos.vue
├── App.vue
├── feathers-client.js
├── feathers-vuex.js
├── main.js
├── .editorconfig
├── .env
├── .gitignore
├── babel.config.js
├── package-lock.json
├── package.json
├── README.md
└── vue.config.js
```

Server-Side Directory Structure

```
/
├── config
│   ├── default.json
│   ├── production.json
│   └── test.json
├── public
│   ├── uploads
│   │   └── images
│   ├── favicon.ico
│   └── index.html
├── src
│   ├── hooks
│   │   ├── attachThumb.js
│   │   ├── findFiles.js
│   │   ├── processConvert.js
│   │   ├── processFile.js
│   │   └── sendVerificationEmail.js
│   ├── middleware
│   │   └── index.js
│   ├── models
│   │   ├── convert.model.js
│   │   ├── message.model.js
│   │   ├── uploads.model.js
│   │   └── users.model.js
│   ├── services
│   │   ├── authmanagement
│   │   │   ├── authmanagement.hooks.js
│   │   │   ├── authmanagement.service.js
│   │   │   └── notifier.js
│   │   ├── convert
│   │   │   ├── convert.class.js
│   │   │   ├── convert.hooks.js
│   │   │   └── convert.service.js
│   │   ├── message
│   │   │   ├── message.hooks.js
│   │   │   └── message.service.js
│   │   ├── streams
│   │   │   ├── streams.hooks.js
│   │   │   └── streams.service.js
│   │   ├── thumbs
│   │   │   ├── thumbs.class.js
│   │   │   ├── thumbs.hooks.js
│   │   │   └── thumbs.service.js
│   │   ├── uploads
│   │   │   ├── uploads.class.js
│   │   │   ├── uploads.hooks.js
│   │   │   └── uploads.service.js
│   │   └── users
│   │       ├── users.class.js
│   │       ├── users.hooks.js
│   │       └── users.service.js
│   └── index.js
├── app.hooks.js
├── app.js
├── authentication.js
├── channels.js
├── index.js
├── logger.js
├── mongodb.js
├── mongoose.js
├── test
│   ├── services
│   │   ├── authmanagement.test.js
│   │   ├── convert.test.js
│   │   ├── message.test.js
│   │   ├── streams.test.js
│   │   ├── thumbs.test.js
│   │   ├── uploads.test.js
│   │   └── users.test.js
│   ├── app.test.js
│   └── authentication.test.js
├── .env
├── .eslintrc.json
├── .gitignore
├── combined.log
├── error.log
├── package-lock.json
├── package.json
└── README.md
```

9.2 Client-Side Code

/vue.config.js

```
1 module.exports = {
2   transpileDependencies: [
3     'vuetify',
4     'feathers-vuex',
5   ],
6 };
```

/package.json

```
1 {
2   "name": "video-converter-client",
3   "version": "0.1.0",
4   "private": true,
5   "scripts": {
6     "serve": "vue-cli-service serve",
7     "build": "vue-cli-service build",
8     "lint": "vue-cli-service lint"
9   },
10  "dependencies": {
11    "@feathersjs/authentication-client": "^4.5.8",
12    "@feathersjs/feathers": "^4.5.8",
13    "@feathersjs/rest-client": "^4.5.8",
14    "@feathersjs/socketio-client": "^4.5.8",
15    "@vue/composition-api": "^0.3.4",
16    "axios": "^0.19.2",
17    "core-js": "^3.6.5",
18    "feathers-hooks-common": "^5.0.3",
19    "feathers-vuex": "^3.11.5",
20    "file-saver": "^2.0.2",
21    "js-file-download": "^0.4.12",
22    "mime-types": "^2.1.27",
23    "socket.io-client": "^2.3.0",
24    "vue": "^2.6.12",
25    "vue-router": "^3.4.3",
26    "vuetify": "^2.3.9",
27    "vuex": "^3.5.1"
28  },
29  "devDependencies": {
30    "@vue/cli-plugin-babel": "^4.5.4",
31    "@vue/cli-plugin-eslint": "^4.5.4",
32    "@vue/cli-plugin-router": "^4.5.4",
33    "@vue/cli-plugin-vuex": "^4.5.4",
34    "@vue/cli-service": "^4.5.4",
35    "@vue/eslint-config-airbnb": "^5.1.0",
36    "babel-eslint": "^10.1.0",
37    "eslint": "^6.7.2",
38    "eslint-plugin-import": "^2.22.0",
39    "eslint-plugin-vue": "^6.2.2",
40    "sass": "^1.26.10",
```

```

41     "sass-loader": "^8.0.0",
42     "vue-cli-plugin-vuetify": "^2.0.7",
43     "vue-template-compiler": "^2.6.12",
44     "vuetify-loader": "^1.6.0"
45 },
46 "eslintConfig": {
47   "root": true,
48   "env": {
49     "node": true
50   },
51   "extends": [
52     "plugin:vue/essential",
53     "@vue/airbnb"
54   ],
55   "parserOptions": {
56     "parser": "babel-eslint"
57   },
58   "rules": {
59     "indent": [
60       2,
61       "tab"
62     ],
63     "no-tabs": 0,
64     "linebreak-style": 0,
65     "no-underscore-dangle": 0
66   }
67 },
68 "browserslist": [
69   "> 1%",
70   "last 2 versions"
71 ]
72 }

```

/babel.config.js

```

1 module.exports = {
2   presets: [
3     '@vue/cli-plugin-babel/preset',
4   ],
5 };

```

/.env

```

1 # SERVER ADDRESS
2 VUE_APP_SERVER_ADDR=http://localhost:3030

```

/public/index.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Online Video Converter</title>
5     <meta charset="utf-8" />
6     <meta http-equiv="X-UA-Compatible" content="IE=edge" />

```

```

7     <meta name="viewport" content="width=device-width,initial-scale=1.0" />
8     <link rel="icon" href="<%= BASE_URL %>favicon.ico" />
9     <title><%= htmlWebpackPlugin.options.title %></title>
10    <link
11        rel="stylesheet"
12        href="https://fonts.googleapis.com/css?family=Roboto:100,300,400,500,700,900"
13    />
14    <link
15        rel="stylesheet"
16        href="https://cdn.jsdelivr.net/npm/@mdi/font@latest/css/materialdesignicons.min.css"
17    />
18 </head>
19 <body>
20     <noscript>
21         <strong>
22             >We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work
23             properly without JavaScript enabled. Please enable it to
24             continue.</strong>
25         >
26     </noscript>
27     <div id="app"></div>
28     <!-- built files will be auto injected -->
29 </body>
30 </html>

```

/src/main.js

```

1  import Vue from 'vue';
2  import VueCompositionApi from '@vue/composition-api';
3  import App from './App.vue';
4  import router from './router';
5  import store from './store';
6  import vuetify from './plugins/vuetify';
7
8  Vue.use(VueCompositionApi);
9
10 Vue.config.productionTip = false;
11
12 new Vue({
13     router,
14     store,
15     vuetify,
16     render: (h) => h(App),
17 }).$mount('#app');

```

/src/feathers-vuex.js

```

1  import feathersVuex from 'feathers-vuex';
2  import feathersClient from './feathers-client';
3
4  const {
5     makeServicePlugin,
6     makeAuthPlugin,
7     BaseModel,
8     models,

```

```

9   FeathersVuex,
10 } = feathersVuex(
11   feathersClient,
12   {
13     serverAlias: 'api',
14     idField: '_id',
15     whitelist: ['$regex', '$options'],
16   },
17 );
18
19 export {
20   makeAuthPlugin,
21   makeServicePlugin,
22   BaseModel,
23   models,
24   FeathersVuex,
25 };

```

/src/feathers-client.js

```

1  import feathers from '@feathersjs/feathers';
2  import auth from '@feathersjs/authentication-client';
3  import rest from '@feathersjs/rest-client';
4
5  const restClient = rest(process.env.VUE_APP_SERVER_ADDR);
6
7  const feathersClient = feathers()
8    .configure(restClient.fetch(window.fetch))
9    .configure(auth({ storage: window.localStorage }));
10 export default feathersClient;

```

/src/App.vue

```

1  <template>
2    <v-app>
3      <!-- Navigation Drawer -->
4      <v-navigation-drawer v-model="drawer" app clipped hide-overlay>
5        <!-- List View If Not Logged In -->
6        <v-list v-if="!isAuthenticated">
7
8          <!-- Login -->
9          <v-list-item link to="/login">
10            <v-list-item-action>
11              <v-icon>mdi-login</v-icon>
12            </v-list-item-action>
13            <v-list-item-content>
14              <v-list-item-title>Login</v-list-item-title>
15            </v-list-item-content>
16          </v-list-item>
17
18          <!-- Sign Up -->
19          <v-list-item link to="/signup">
20            <v-list-item-action>
21              <v-icon>mdi-account</v-icon>
22            </v-list-item-action>

```



```

23         <v-list-item-content>
24             <v-list-item-title>Sign Up</v-list-item-title>
25         </v-list-item-content>
26     </v-list-item>
27
28 </v-list>
29
30 <!-- List View If Logged In -->
31 <v-list v-if="isAuthenticated">
32     <!-- Profile -->
33     <v-list-item link to="/account">
34         <v-list-item-action>
35             <v-icon>mdi-face-profile</v-icon>
36         </v-list-item-action>
37         <v-list-item-content>
38             <v-list-item-title>Account</v-list-item-title>
39         </v-list-item-content>
40     </v-list-item>
41
42     <!-- Converted Files -->
43     <v-list-item link to="/videos">
44         <v-list-item-action>
45             <v-icon>mdi-video</v-icon>
46         </v-list-item-action>
47         <v-list-item-content>
48             <v-list-item-title>My Videos</v-list-item-title>
49         </v-list-item-content>
50     </v-list-item>
51
52 <!-- Converted Files -->
53     <v-list-item link to="/conversionhistory">
54         <v-list-item-action>
55             <v-icon>mdi-history</v-icon>
56         </v-list-item-action>
57         <v-list-item-content>
58             <v-list-item-title>Conversion History</v-list-item-title>
59         </v-list-item-content>
60     </v-list-item>
61
62 <!-- Convert New File -->
63     <v-list-item v-on:click="logout">
64         <v-list-item-action>
65             <v-icon>mdi-logout</v-icon>
66         </v-list-item-action>
67         <v-list-item-content>
68             <v-list-item-title>Log Out</v-list-item-title>
69         </v-list-item-content>
70     </v-list-item>
71
72 </v-list>
73 </v-navigation-drawer>
74
75 <!-- NavBar -->
76 <v-app-bar app clipped-left>
77     <v-app-bar-nav-icon @click.stop="drawer = !drawer" />

```

```

78     <v-toolbar-title>Online Video Converter</v-toolbar-title>
79 </v-app-bar>
80
81 <!-- Views Injection -->
82 <transition
83     name="fade"
84     mode="out-in"
85 >
86 <router-view />
87 </transition>
88
89 <!-- Footer -->
90 <v-footer app>
91     <span>&copy; 2020 Created By Bharat Singh Rajput</span>
92 </v-footer>
93 </v-app>
94 </template>
95
96 <script>
97 import { mapGetters, mapActions } from 'vuex';
98
99 export default {
100     name: 'app',
101     props: {
102         source: String,
103     },
104     data: () => ({
105         drawer: null,
106     }),
107     created() {
108         this.$vuetify.theme.dark = true;
109     },
110     computed: {
111         ...mapGetters('auth', ['isAuthenticated']),
112     },
113     methods: {
114         ...mapActions('auth', ['logout']),
115         logOut() {
116             this.logout().then(() => {
117                 this.$router.push('/login');
118                 this.$router.go();
119             }).catch(() => {
120                 this.$router.push('/login');
121                 this.$router.go();
122             });
123         },
124     },
125 };
126 </script>
127
128 <style scoped>
129 .fade-enter-active,
130 .fade-leave-active {
131     transition-duration: 0.3s;
132     transition-property: opacity;

```

```

133   transition-timing-function: ease;
134 }
135
136 .fade-enter,
137 .fade-leave-active {
138   opacity: 0
139 }
140 </style>

```

```
/src/components/Dialog.vue
```

```

1 <template>
2   <v-dialog
3     v-model="config.show"
4     :width="config.width || width"
5     :persistent="config.isPersistent || isPersistent">
6     <v-card>
7       <v-card-title
8         class="title lighten-1"
9         primary-title :class="config.color || defaults.color">
10        <v-icon left>{{config.icon || defaults.icon}}</v-icon>
11          {{config.title || title}}
12        </v-card-title>
13        <v-card-text>
14          <br />
15          {{config.message || message}}
16        </v-card-text>
17        <v-divider></v-divider>
18        <v-card-actions>
19          <v-spacer></v-spacer>
20          <v-btn
21            :color="config.buttonColor || defaults.buttonColor"
22            text
23            @click="onClick()"
24            :to="config.redirect === 'refresh' ? '' : config.redirect || redirect">
25            {{config.buttonText || buttonText}}
26          </v-btn>
27        </v-card-actions>
28      </v-card>
29    </v-dialog>
30 </template>
31
32 <script>
33 export default {
34   data: () => ({
35     title: 'Dialog Title',
36     show: false,
37     width: 400,
38     color: 'info',
39     icon: 'mdi-alert-circle',
40     message: 'Dialog Message',
41     redirect: '',
42     buttonText: 'OK',
43     buttonColor: 'primary',
44     isPersistent: false,

```

```

45     type: 'info',
46   }),
47   computed: {
48     defaults() {
49       switch (this.config.type ? this.config.type.toLowerCase() : 'info') {
50         case 'info':
51           return {
52             color: 'info',
53             icon: 'mdi-information',
54             buttonColor: 'info',
55           };
56         case 'error':
57           return {
58             color: 'error',
59             icon: 'mdi-alert-circle',
60             buttonColor: 'error',
61           };
62         default:
63           return {};
64       }
65     },
66   },
67   methods: {
68     onClick() {
69       this.config.show = false;
70       if (this.config.redirect === 'refresh') {
71         this.$router.go();
72       }
73     },
74   },
75   props: [
76     'config',
77   ],
78 };
79 </script>

```

/src/components/Loading.vue

```

1   <template>
2     <v-dialog v-model="config.show" width="400" :persistent="config.isPersistent" hide-overlay>
3       <v-card class="mx-auto" loading disabled>
4         <v-card-title>Hold on..</v-card-title>
5         <v-card-text>{{config.message}}</v-card-text>
6       </v-card>
7     </v-dialog>
8   </template>
9
10  <script>
11    export default {
12      props: [
13        'config',
14      ],
15    };
16  </script>

```

/src/components/YesNoDialog.vue

```

1 <template>
2   <v-dialog
3     v-model="config.show"
4     max-width="500px"
5     :persistent="config.isPersistent ? config.isPersistent : false"
6   >
7     <v-card>
8       <v-card-title>{{ config.title ? config.title : title }}</v-card-title>
9       <v-card-text>
10        {{ config.message ? config.message : message }}
11      </v-card-text>
12      <v-card-actions>
13        <v-spacer></v-spacer>
14        <v-btn
15          color="red"
16          v-on:click="config.onYes ? config.onYes[0](config.onYes[1]) : (show = false)"
17        >
18          {{ config.yesText ? config.yesText : yesText }}
19        </v-btn>
20        <v-btn
21          color="primary"
22          v-on:click="config.onNo ? config.onNo[0](config.onNo[1]) : (show = false)"
23        >
24          {{ config.noText ? config.noText : noText }}
25        </v-btn>
26      </v-card-actions>
27    </v-card>
28  </v-dialog>
29 </template>
30
31 <script>
32 export default {
33   name: 'YesNoDialog',
34   data: () => ({
35     show: false,
36     title: 'Dialog Title',
37     message: 'Dialog Message',
38     onYes: null,
39     onNo: null,
40     yesText: 'Yes',
41     noText: 'No',
42   }),
43   props: ['config'],
44 };
45 </script>

```

```
/src/plugins/vuetify.js
```

```

1 import Vue from 'vue';
2 import Vuetify from 'vuetify/lib';
3
4 Vue.use(Vuetify);
5
6 export default new Vuetify({});

```

```
/src/router/index.js
```

```

1  import Vue from 'vue';
2  import VueRouter from 'vue-router';
3  import Login from '@/views/Login.vue';
4  import SignUp from '@/views/SignUp.vue';
5  import Videos from '@/views/Videos.vue';
6  import Account from '@/views/Account.vue';
7  import Verify from '@/views/Verify.vue';
8  import Reset from '@/views/Reset.vue';
9  import ConversionHistory from '@/views/ConversionHistory.vue';
10 import Store from '@/store/index';
11
12 Vue.use(VueRouter);
13
14 const routes = [
15   {
16     path: '/',
17     name: 'home',
18     beforeEnter(to, from, next) {
19       Store.dispatch('auth/authenticate').then(() => {
20         next('/videos');
21       }).catch(() => {
22         next('/login');
23       });
24     },
25   },
26   {
27     path: '/verify',
28     name: 'verify',
29     component: Verify,
30   },
31   {
32     path: '/reset',
33     name: 'reset',
34     component: Reset,
35   },
36   {
37     path: '/signup',
38     name: 'signup',
39     component: SignUp,
40     beforeEnter(to, from, next) {
41       Store.dispatch('auth/authenticate').then(() => {
42         next('/videos');
43       }).catch(() => {
44         next();
45       });
46     },
47   },
48   {
49     path: '/login',
50     name: 'login',
51     component: Login,
52     beforeEnter(to, from, next) {
53       Store.dispatch('auth/authenticate').then(() => {
54         next('/videos');
55       }).catch(() => {

```

```

56     next();
57   });
58 },
59 },
60 {
61   path: '/videos',
62   name: 'videos',
63   component: Videos,
64   beforeEnter(to, from, next) {
65     if (!Store.getters['auth/isAuthenticated']) {
66       Store.dispatch('auth/authenticate').then(() => {
67         next();
68       }).catch(() => {
69         next('/login');
70       });
71     } else {
72       next();
73     }
74   },
75 },
76 {
77   path: '/account',
78   name: 'account',
79   component: Account,
80   beforeEnter(to, from, next) {
81     Store.dispatch('auth/authenticate').then(() => {
82       next();
83     }).catch(() => {
84       next('/login');
85     });
86   },
87 },
88 {
89   path: '/conversionhistory',
90   name: 'conversionhistory',
91   component: ConversionHistory,
92   beforeEnter(to, from, next) {
93     Store.dispatch('auth/authenticate').then(() => {
94       next();
95     }).catch(() => {
96       next('/login');
97     });
98   },
99 },
100 ];
101
102 const router = new VueRouter({
103   routes,
104   mode: 'history',
105 });
106
107 export default router;

```

```

/src/store/index.js

```

```

1 import Vue from 'vue';

```

```

2 import Vuex from 'vuex';
3 import { FeathersVuex } from '@/feathers-vuex';
4 import auth from './store.auth';
5
6 Vue.use(Vuex);
7 Vue.use(FeathersVuex);
8
9 const requireModule = require.context('./services', false, /\.js$/);
10
11 const servicePlugins = requireModule
12   .keys()
13   .map((modulePath) => requireModule(modulePath).default);
14
15 export default new Vuex.Store({
16   state: {},
17   mutations: {},
18   actions: {},
19   plugins: [...servicePlugins, auth],
20 });

```

/src/store/store.auth.js

```

1 import { makeAuthPlugin } from '@/feathers-vuex';
2
3 export default makeAuthPlugin({ userService: 'users' });

```

/src/store/services/authmanagement.js

```

1 // src/store/services/authmanagement.js
2 import { makeServicePlugin, BaseModel } from '@/feathers-vuex';
3 import feathersClient from '@/feathers-client';
4
5 class AuthManagement extends BaseModel {
6   /* eslint-disable */
7   constructor(data, options) {
8     super(data, options)
9   }
10
11   /* eslint-enable */
12
13   // Required for lFeathersVuex plugin to work after production transpile.
14   static modelName = 'AuthManagement';
15
16   // Define default properties here
17   static instanceDefaults() {
18     return {};
19   }
20 }
21
22 const servicePath = 'authManagement';
23 const servicePlugin = makeServicePlugin({
24   Model: AuthManagement,
25   service: feathersClient.service(servicePath),
26   servicePath,
27 });

```



```

28
29 // Setup the client-side Feathers hooks.
30 feathersClient.service(servicePath).hooks({
31   before: {
32     all: [],
33     find: [],
34     get: [],
35     create: [],
36     update: [],
37     patch: [],
38     remove: [],
39   },
40   after: {
41     all: [],
42     find: [],
43     get: [],
44     create: [],
45     update: [],
46     patch: [],
47     remove: [],
48   },
49   error: {
50     all: [],
51     find: [],
52     get: [],
53     create: [],
54     update: [],
55     patch: [],
56     remove: [],
57   },
58 });
59
60 export default servicePlugin;

```

```
/src/store/services/convert.js
```

```

1 // src/store/services/convert.js
2 import { makeServicePlugin, BaseModel } from '@feathers-vuex';
3 import feathersClient from '@feathers-client';
4
5 class Convert extends BaseModel {
6   /* eslint-disable */
7   constructor(data, options) {
8     super(data, options)
9   }
10
11   /* eslint-enable */
12
13   // Required for lFeathersVue plugin to work after production transpile.
14   static modelName = 'Convert';
15
16   // Define default properties here
17   static instanceDefaults() {
18     return {
19       name: '',

```

```

20     format: '',
21     width: 0,
22     height: 0,
23     originalVideoId: '',
24     type: '',
25     isAudioDisabled: false,
26     frameRate: 0,
27     durationStart: 0,
28     durationEnd: 0,
29     audioBitrate: 0,
30     videoBitrate: 0,
31   };
32 }
33 }
34
35 const servicePath = 'convert';
36 const servicePlugin = makeServicePlugin({
37   Model: Convert,
38   service: feathersClient.service(servicePath),
39   servicePath,
40 });
41
42 // Setup the client-side Feathers hooks.
43 feathersClient.service(servicePath).hooks({
44   before: {
45     all: [],
46     find: [],
47     get: [],
48     create: [],
49     update: [],
50     patch: [],
51     remove: [],
52   },
53   after: {
54     all: [],
55     find: [],
56     get: [],
57     create: [],
58     update: [],
59     patch: [],
60     remove: [],
61   },
62   error: {
63     all: [],
64     find: [],
65     get: [],
66     create: [],
67     update: [],
68     patch: [],
69     remove: [],
70   },
71 });
72
73 export default servicePlugin;

```

```
/src/store/services/thumbs.js
```

```

1  // src/store/services/thumbs.js
2  import { makeServicePlugin, BaseModel } from '@/feathers-vuex';
3  import feathersClient from '@/feathers-client';
4
5  class Thumbs extends BaseModel {
6    /* eslint-disable */
7    constructor(data, options) {
8      super(data, options)
9    }
10
11    /* eslint-enable */
12
13    // Required for lFeathersVue plugin to work after production transpile.
14    static modelName = 'Thumbs';
15
16    // Define default properties here
17    static instanceDefaults() {
18      return {};
19    }
20  }
21
22  const servicePath = 'thumbs';
23  const servicePlugin = makeServicePlugin({
24    Model: Thumbs,
25    service: feathersClient.service(servicePath),
26    servicePath,
27  });
28
29  // Setup the client-side Feathers hooks.
30  feathersClient.service(servicePath).hooks({
31    before: {
32      all: [],
33      find: [],
34      get: [],
35      create: [],
36      update: [],
37      patch: [],
38      remove: [],
39    },
40    after: {
41      all: [],
42      find: [],
43      get: [],
44      create: [],
45      update: [],
46      patch: [],
47      remove: [],
48    },
49    error: {
50      all: [],
51      find: [],
52      get: [],
53      create: [],
54      update: [],
55      patch: [],

```

```

56     remove: [],
57   },
58 });
59
60 export default servicePlugin;

```

/src/store/services/uploads.js

```

1  // src/store/services/uploads.js
2  import { makeServicePlugin, BaseModel } from '@/feathers-vuex';
3  import feathersClient from '@/feathers-client';
4
5  class Uploads extends BaseModel {
6    /* eslint-disable */
7    constructor(data, options) {
8      super(data, options)
9    }
10
11    /* eslint-enable */
12
13    // Required for lFeathersVuex plugin to work after production transpile.
14    static modelName = 'Uploads';
15
16    // Define default properties here
17    static instanceDefaults() {
18      return {
19        description: '',
20        originalName: '',
21        newNameWithPath: '',
22        format: '',
23        videoBitrate: 0,
24        audioBitrate: 0,
25        resolution: '',
26        duration: 0,
27        fps: 0,
28        fileSize: 0,
29        thumbnail: '',
30      };
31    }
32  }
33
34  const servicePath = 'uploads';
35  const servicePlugin = makeServicePlugin({
36    Model: Uploads,
37    service: feathersClient.service(servicePath),
38    servicePath,
39  });
40
41  // Setup the client-side Feathers hooks.
42  feathersClient.service(servicePath).hooks({
43    before: {
44      all: [],
45      find: [],
46      get: [],
47      create: [],

```

```

48     update: [],
49     patch: [],
50     remove: [],
51 },
52 after: {
53     all: [],
54     find: [],
55     get: [],
56     create: [],
57     update: [],
58     patch: [],
59     remove: [],
60 },
61 error: {
62     all: [],
63     find: [],
64     get: [],
65     create: [],
66     update: [],
67     patch: [],
68     remove: [],
69 },
70 });
71
72 export default servicePlugin;

```

```
/src/store/services/users.js
```

```

1  // src/store/services/users.js
2  import { makeServicePlugin, BaseModel } from '@/feathers-vuex';
3  import feathersClient from '@/feathers-client';
4
5  class User extends BaseModel {
6    /* eslint-disable */
7    constructor(data, options) {
8      super(data, options)
9    }
10
11    /* eslint-enable */
12
13    // Required for LFeathersVuex plugin to work after production transpile.
14    static modelName = 'User';
15
16    // Define default properties here
17    static instanceDefaults() {
18      return {
19        displayName: '',
20        email: '',
21        dob: new Date(1997, 8, 29).toISOString().substr(0, 10),
22        avatar: '',
23      };
24    }
25  }
26
27  const servicePath = 'users';

```

```

28  const servicePlugin = makeServicePlugin({
29    Model: User,
30    service: feathersClient.service(servicePath),
31    servicePath,
32  });
33
34  // Setup the client-side Feathers hooks.
35  feathersClient.service(servicePath).hooks({
36    before: {
37      all: [],
38      find: [],
39      get: [],
40      create: [],
41      update: [],
42      patch: [],
43      remove: [],
44    },
45    after: {
46      all: [],
47      find: [],
48      get: [],
49      create: [],
50      update: [],
51      patch: [],
52      remove: [],
53    },
54    error: {
55      all: [],
56      find: [],
57      get: [],
58      create: [],
59      update: [],
60      patch: [],
61      remove: [],
62    },
63  });
64
65  export default servicePlugin;

```

```
/src/views/Account.vue
```

```

1  <template>
2    <v-main>
3      <v-row align="center" justify="center" class="fill-height">
4        <v-col cols="8" md="4">
5          <v-carousel
6            hide-delimiters
7            :show-arrows="false"
8            :value="carouselVal"
9            height="50%"
10           fluid
11          >
12            <v-carousel-item>
13              <v-card>
14                <v-toolbar color="primary" dark>

```

```

15     <v-toolbar-title class="white--text">My Account</v-toolbar-title>
16     <v-spacer></v-spacer>
17     <v-tooltip right>
18         <template v-slot:activator="{ on }">
19             <v-btn icon large v-on="on">
20                 <v-icon>mdi-account</v-icon>
21             </v-btn>
22         </template>
23         <span>My Account</span>
24     </v-tooltip>
25 </v-toolbar>
26 <v-row justify="space-around" class="mt-4">
27     <v-avatar size="150">
28         
29     </v-avatar>
30 </v-row>
31 <v-row justify="space-around" class="mt-2">
32     <h2>{{ user.displayName }}</h2>
33 </v-row>
34 <v-row>
35     <v-spacer />
36     <v-col cols="12" md="10">
37         <v-simple-table class="ma-5">
38             <tbody>
39                 <tr>
40                     <td>Display Name</td>
41                     <td>{{ user.displayName }}</td>
42                 </tr>
43                 <tr>
44                     <td>Email</td>
45                     <td>{{ user.email }}</td>
46                 </tr>
47                 <tr>
48                     <td>Date Of Birth</td>
49                     <td>
50                         {{
51                             new Date(user.dob)
52                                 .toISOString()
53                                 .split(' ')
54                                 .slice(0, 4)
55                                 .join(' ')
56                         }}
57                     </td>
58                 </tr>
59                 <tr>
60                     <td>Date Of Joining</td>
61                     <td>
62                         {{
63                             new Date(user.createdAt)
64                                 .toISOString()
65                                 .split(' ')
66                                 .slice(0, 4)
67                                 .join(' ')
68                         }}
69                     </td>

```

```

70         </tr>
71     </tbody>
72 </v-simple-table>
73 </v-col>
74 <v-spacer />
75 </v-row>
76 <v-spacer />
77 <v-card-actions>
78     <v-spacer />
79     <v-btn v-on:click="setCarouselVal(1)" color="primary" class="ma-2">
80         <v-icon left>mdi-pen</v-icon>Edit Profile
81     </v-btn>
82 </v-card-actions>
83 </v-card>
84 </v-carousel-item>
85 <v-carousel-item>
86     <v-card>
87         <v-toolbar color="primary" dark>
88             <v-toolbar-title class="white--text">Edit Profile</v-toolbar-title>
89             <v-spacer></v-spacer>
90             <v-tooltip right>
91                 <template v-slot:activator="{ on }">
92                     <v-btn icon large v-on="on">
93                         <v-icon>mdi-pen</v-icon>
94                     </v-btn>
95                 </template>
96                 <span>Edit Profile</span>
97             </v-tooltip>
98         </v-toolbar>
99         <v-row justify="space-around" class="mt-4">
100             <v-avatar size="150">
101                 
105             </v-avatar>
106         </v-row>
107         <v-form v-model="valid" ref="form" class="mx-5">
108             <!-- Display Name -->
109             <v-text-field
110                 label="Name"
111                 name="name"
112                 prepend-icon="mdi-face-profile"
113                 clearable
114                 counter="30"
115                 :rules="notEmptyRules"
116                 v-model="editUser.displayName"
117                 hint="Your friendly name"
118                 maxlength="30"
119                 type="text"
120             />
121             <!-- Date Of Birth -->
122             <v-menu
123                 ref="menu"
124                 v-model="menu"

```



```

125         :close-on-content-click="false"
126         :return-value.sync="user.dob"
127         transition="scale-transition"
128         offset-y
129         min-width="290px"
130     >
131         <template v-slot:activator="{ on }">
132             <v-text-field
133                 v-model="editUser.dob"
134                 label="Date Of Birth"
135                 prepend-icon="mdi-calendar"
136                 readonly
137                 v-on="on"
138             ></v-text-field>
139         </template>
140         <v-date-picker v-model="editUser.dob" no-title scrollable>
141             <v-spacer />
142             <v-btn text color="primary" @click="$refs.menu.save(editUser.dob)">
143                 OK
144             </v-btn>
145         </v-date-picker>
146     </v-menu>
147
148     <!-- Avatar -->
149     <v-file-input
150         :rules="avatarRules"
151         accept="image/png, image/jpeg, image/bmp"
152         placeholder="Change avatar"
153         prepend-icon="mdi-camera"
154         label="Avatar"
155         v-model="avatarFile"
156         v-on:change="onFilePicked"
157     ></v-file-input>
158 </v-form>
159 <v-spacer />
160 <v-card-actions>
161     <v-btn color="error" class="ma-2" v-on:click="setCarouselVal(0)">
162         <v-icon size="30" left> mdi-keyboard-backspace </v-icon>
163     </v-btn>
164     <v-spacer />
165     <v-btn
166         color="deep-purple darken-3"
167         class="ma-2"
168         @click="changePasswordDlg = true"
169     >
170         <v-icon left>mdi-key</v-icon>Password
171     </v-btn>
172     <v-btn color="primary" class="ma-2" v-on:click="validate">
173         <v-icon left>mdi-content-save</v-icon>Save
174     </v-btn>
175 </v-card-actions>
176 </v-card>
177 </v-carousel-item>
178 <v-dialog v-model="changePasswordDlg" width="400px">
179     <v-card>

```

```

180         <v-card-title class="headline">Change Your Password</v-card-title>
181         <v-card-text>
182             Changing your password requires an e-mail verification. Click confirm to
183             generate a reset password e-mail.
184         </v-card-text>
185         <v-card-actions>
186             <v-spacer></v-spacer>
187             <v-btn color="error" text @click="changePasswordDlg = false"
188                 >Cancel</v-btn>
189             >
190             <v-btn color="primary" text @click="sendResetMail()">Confirm</v-btn>
191         </v-card-actions>
192     </v-card>
193 </v-dialog>
194 </v-carousel>
195 </v-col>
196 </v-row>
197 <Dialog :config="dConfig" />
198 <Loading :config="lConfig" v-if="isUpdatePending" />
199 </v-main>
200 </template>
201
202 <script>
203 import { mapGetters } from 'vuex';
204 import Dialog from '@components/Dialog.vue';
205 import Loading from '@components/Loading.vue';
206 // import axios from 'axios';
207
208 export default {
209     name: 'Account',
210     data: () => ({
211         editUser: {
212             displayName: 'displayName',
213             avatar: '',
214             dob: new Date(),
215         },
216         changePasswordDlg: false,
217         dConfig: {},
218         lConfig: {
219             isPersistent: true,
220             show: 'true',
221             message: 'Updating your profile!',
222         },
223         isUpdatePending: false,
224         avatarFile: null,
225         valid: true,
226         menu: false,
227         carouselVal: 0,
228         notEmptyRules: [(v) => (v && v.length) > 0 || 'Field cannot be empty'],
229         avatarRules: [
230             (v) => !v || v.size < 2000000 || 'Avatar size should be less than 2 MB!',
231         ],
232     }),
233     computed: {
234         ...mapGetters('auth', ['user']),

```

```

235 },
236 methods: {
237   sendResetMail() {
238     this.changePasswordDlg = false;
239     const { AuthManagement } = this.$FeathersVuex.api;
240     const data = {
241       action: 'sendResetPwd',
242       value: {
243         email: this.user.email,
244       },
245     };
246     const authRequest = new AuthManagement(data);
247     authRequest
248       .save()
249       .then(() => {
250         this.dConfig = {
251           show: true,
252           type: 'info',
253           title: 'Sucess',
254           message: 'Please check your email for a password reset link!',
255           isPersistent: true,
256           redirect: '/login',
257         };
258         this.carouselVal = 0;
259       })
260       .catch(() => {
261         this.dConfig = {
262           show: true,
263           title: 'An error occured',
264           message: 'Please check your email.',
265           buttonText: 'Retry',
266           isPersistent: true,
267           type: 'error',
268         };
269       });
270   },
271   validate() {
272     if (this.$refs.form.validate()) {
273       this.isUpdatePending = true;
274       this.user.displayName = this.editUser.displayName;
275       this.user.dob = this.editUser.dob;
276       if (this.avatarFile) {
277         this.user.avatar = this.editUser.avatar;
278       }
279       this.user.displayName = this.editUser.displayName;
280       delete this.user.isVerified;
281       this.user
282         .patch()
283         .then(() => {
284           this.avatarFile = null;
285           this.dConfig = {
286             show: true,
287             title: 'Success',
288             message: 'Your changes have been saved successfully.',
289             buttonText: 'OK',

```

```

290         isPersistent: true,
291         redirect: 'refresh',
292         type: 'info',
293     };
294 })
295     .catch(() => {
296         this.dConfig = {
297             show: true,
298             title: 'Error',
299             message: 'An error occurred while updating your details.',
300             buttonText: 'Retry',
301             isPersistent: true,
302             redirect: 'refresh',
303             type: 'error',
304         };
305     });
306 }
307 },
308 setCarouselVal(val) {
309     this.carouselVal = val;
310 },
311 decode(dat) {
312     try {
313         const b = Buffer.from(dat, 'base64');
314         return b.toString();
315     } catch {
316         return null;
317     }
318 },
319 onFilePicked(file) {
320     if (file) {
321         const fileReader = new FileReader();
322         fileReader.addEventListener('load', () => {
323             this.editUser.avatar = fileReader.result;
324         });
325         fileReader.readAsDataURL(file);
326     } else {
327         this.editUser.avatar = '';
328     }
329     this.avatarFile = file;
330 },
331 },
332 mounted() {
333     this.editUser.displayName = this.user.displayName;
334     this.editUser.dob = new Date(this.user.dob).toISOString().substring(0, 10);
335 },
336 components: {
337     Dialog,
338     Loading,
339 },
340 };
341 </script>

```

```
/src/views/ConversionHistory.vue
```

```
1 <template>
```

```

2   <v-main>
3     <v-container class="fill-height" fluid>
4       <v-row class="ma-0 pa-0 fill-height">
5         <v-col cols="12" sm="12" offset-sm="3" class="ma-0 pa-0 fill-height">
6           <v-card class="ma-0 pa-0 fill-height">
7             <v-toolbar color="primary" dark>
8               <v-toolbar-title class="white--text">Conversion History</v-toolbar-title>
9               <v-spacer />
10              <v-icon x-large>mdi-history</v-icon>
11            </v-toolbar>
12            <v-subheader>Converted Videos</v-subheader>
13            <v-row>
14              <v-text-field
15                placeholder="Search Video..."
16                v-model="filterString"
17                rounded
18                dense
19                filled
20                class="ml-5 my-2"
21                prepend-inner-icon="mdi-magnify"
22              />
23              <v-btn fab class="mx-5" @click="getVideos(generateParams())">
24                <v-icon>mdi-movie-search</v-icon>
25              </v-btn>
26              <v-menu
27                ref="menu"
28                v-model="menu"
29                :close-on-content-click="false"
30                :return-value.sync="queryDate"
31                transition="scale-transition"
32                offset-y
33                min-width="290px"
34              >
35                <template v-slot:activator="{ on }">
36                  <v-btn fab v-on="on" class="mx-2">
37                    <v-icon>mdi-calendar</v-icon>
38                  </v-btn>
39                </template>
40                <v-date-picker v-model="queryDate" no-title scrollable class="ma-2">
41                  <v-spacer></v-spacer>
42                  <v-btn
43                    text
44                    color="primary"
45                    @click="$refs.menu.save(queryDate) & getVideos(generateParams())"
46                    >OK</v-btn>
47                >
48              </v-date-picker>
49            </v-menu>
50            <v-tooltip bottom>
51              <template v-slot:activator="{ on, attrs }">
52                <v-btn fab v-on="on" class="mx-2 mr-5" @click="getVideos({})">
53                  <v-icon color="error" v-bind="attrs">mdi-restore</v-icon>
54                </v-btn>
55              </template>
56              <span>Clear Search Parameters</span>

```

```

57     </v-tooltip>
58 </v-row>
59 <div class="text-center" style="position: relative; bottom: 0">
60     <v-pagination
61         v-model="page"
62         :length="paginateLength == 0 ? 1 : paginateLength"
63         circle
64     ></v-pagination>
65 </div>
66 <v-list two-line v-if="viewMode == 'List'">
67     <v-list-item v-for="item in videos" :key="item._id" link>
68         <v-list-item-avatar>
69             <v-icon class="blue">mdi-video</v-icon>
70         </v-list-item-avatar>
71         <v-list-item-content>
72             <v-list-item-title>{{ item.name }}</v-list-item-title>
73             <v-list-item-subtitle>
74                 {{ new Date(item.createdAt).toUTCString() }}
75             </v-list-item-subtitle>
76         </v-list-item-content>
77         <v-btn icon ripple>
78             <v-icon color="grey lighten-1" @click="downloadVideo(item)"
79                 >mdi-download</v-icon>
80             >
81         </v-btn>
82         <v-btn icon ripple>
83             <v-icon color="grey lighten-1" @click="showInfo(item)"
84                 >mdi-information</v-icon>
85             >
86         </v-btn>
87         <v-btn icon ripple>
88             <v-icon color="grey lighten-1" @click="initDeleteVideo(item)"
89                 >mdi-delete</v-icon>
90             >
91         </v-btn>
92     </v-list-item>
93 </v-list>
94 <v-container fluid v-if="viewMode == 'Card'">
95     <v-row>
96         <v-card
97             v-for="item in videos"
98             :key="item._id"
99             max-width="336px"
100             color="primary"
101             elevation="24"
102             class="mx-5"
103         >
104             <v-img
105                 :src="'data:image/png;base64,' + item.thumbnail"
106                 class="white--text align-end"
107                 height="188px"
108                 width="336px"
109             >
110             <v-card-title v-text="item.name"></v-card-title>
111         </v-img>

```

```

112
113         <v-card-actions>
114             <v-spacer></v-spacer>
115             <v-btn icon @click="downloadVideo(item)">
116                 <v-icon>mdi-download</v-icon>
117             </v-btn>
118
119             <v-btn icon @click="showInfo(item)">
120                 <v-icon>mdi-information</v-icon>
121             </v-btn>
122
123             <v-btn icon @click="initDeleteVideo(item)">
124                 <v-icon>mdi-delete</v-icon>
125             </v-btn>
126         </v-card-actions>
127     </v-card>
128 </v-row>
129 </v-container>
130 <v-speed-dial
131     absolute
132     v-model="fab"
133     bottom
134     right
135     direction="top"
136     transition="slide-y-reverse-transition"
137 >
138     <template v-slot:activator>
139         <v-btn v-model="fab" color="blue darken-2" dark fab>
140             <v-icon v-if="fab">mdi-close</v-icon>
141             <v-icon v-else>mdi-view-split-vertical</v-icon>
142         </v-btn>
143     </template>
144     <v-btn fab dark small color="green" @click="viewMode = 'List'">
145         <v-icon>mdi-view-list</v-icon>
146     </v-btn>
147     <v-btn fab dark small color="indigo" @click="viewMode = 'Card'">
148         <v-icon>mdi-view-comfy</v-icon>
149     </v-btn>
150 </v-speed-dial>
151 </v-card>
152 </v-col>
153 </v-row>
154 </v-container>
155 <v-dialog v-model="showInfoDlg" max-width="500px">
156     <v-card>
157         <v-card-title>Video Information</v-card-title>
158         <v-card-text>
159             <v-simple-table>
160                 <thead>
161                     <tr>
162                         <th class="text-left">Attribute</th>
163                         <th class="text-left">Value</th>
164                     </tr>
165                 </thead>
166                 <tbody v-if="this.selectedVideo != null">

```

```

167         <tr>
168             <td>Name</td>
169             <td>{{ this.selectedVideo.name }}</td>
170         </tr>
171         <tr>
172             <td>Format</td>
173             <td>{{ this.selectedVideo.format }}</td>
174         </tr>
175         <tr>
176             <td>Resolution</td>
177             <td>{{ this.selectedVideo.height }}x{{ this.selectedVideo.width }}</td>
178         </tr>
179         <tr>
180             <td>Size</td>
181             <td>
182                 {{ parseFloat(this.selectedVideo.fileSize / 1000000.0).toFixed(2) }} MB
183             </td>
184         </tr>
185         <tr>
186             <td>Duration</td>
187             <td>{{ secondsToHms(this.selectedVideo.durationEnd) }}</td>
188         </tr>
189         <tr>
190             <td>Converted To</td>
191             <td>{{ this.selectedVideo.type }}</td>
192         </tr>
193         <tr>
194             <td>Frames Per Second</td>
195             <td>{{ this.selectedVideo.frameRate }}</td>
196         </tr>
197         <tr>
198             <td>Is Audio Disabled?</td>
199             <td>{{ this.selectedVideo.isAudioDisabled }}</td>
200         </tr>
201         <tr>
202             <td>Video Bitrate</td>
203             <td>{{ this.selectedVideo.videoBitrate }} kb/s</td>
204         </tr>
205         <tr>
206             <td>Audio Bitrate</td>
207             <td>{{ this.selectedVideo.audioBitrate }} kb/s</td>
208         </tr>
209     </tbody>
210 </v-simple-table>
211 </v-card-text>
212 <v-card-actions>
213     <v-spacer></v-spacer>
214     <v-btn color="primary" v-on:click="showInfoDlg = false">
215         Close
216     </v-btn>
217 </v-card-actions>
218 </v-card>
219 </v-dialog>
220 <YesNoDialog :config="yesNoConfig" />
221 <Dialog :config="dConfig" />

```



```

222     <Loading :config="lConfig" />
223   </v-main>
224 </template>
225
226 <script>
227 import YesNoDialog from '@components/YesNoDialog.vue';
228 import Dialog from '@components/Dialog.vue';
229 import Loading from '@components/Loading.vue';
230 import axios from 'axios';
231
232 export default {
233   data: () => ({
234     queryDate: new Date().toISOString().substr(0, 10),
235     queryDateSuccessor: new Date().toISOString().substr(0, 10),
236     viewMode: 'List',
237     menu: false,
238     page: 1,
239     fab: false,
240     filterString: '',
241     videos: [],
242     paginateLength: 0,
243     queryLimit: 10,
244     yesNoConfig: {},
245     dConfig: {},
246     lConfig: {
247       isPersistent: true,
248     },
249     selectedVideo: null,
250     showInfoDlg: false,
251   }),
252   mounted() {
253     this.getVideos();
254   },
255   methods: {
256     generateParams() {
257       const params = {};
258       if (this.queryDateSuccessor) {
259         params.createdAt = {
260           $gte: this.queryDate,
261           $lt: this.queryDateSuccessor,
262         };
263       }
264       if (this.filterString) {
265         params.originalName = {
266           $search: this.filterString,
267         };
268       }
269       return params;
270     },
271     decode(dat) {
272       try {
273         const b = Buffer.from(dat, 'base64');
274         return b.toString();
275       } catch {
276         return null;

```

```

277     }
278 },
279 downloadVideo(video) {
280     const token = window.localStorage['feathers-jwt'];
281     axios
282     .get(`${process.env.VUE_APP_SERVER_ADDR}/convert/${video._id}?download=true`, {
283         headers: {
284             Authorization: `Bearer ${token}`,
285         },
286         responseType: 'blob',
287     })
288     .then((response) => {
289         const url = window.URL.createObjectURL(new Blob([response.data]));
290         const link = document.createElement('a');
291         link.href = url;
292         link.setAttribute('download', `${video.name}.${video.format}`);
293         document.body.appendChild(link);
294         link.click();
295         link.remove();
296         this.dConfig = {
297             show: true,
298             title: 'Sucess',
299             message: 'Your download has been started successfully.',
300             buttonText: 'OK',
301             isPersistent: true,
302             type: 'info',
303         };
304     })
305     .catch(() => {
306         this.dConfig = {
307             show: true,
308             title: 'Error',
309             message: 'Some error has occured while downloading your video.',
310             buttonText: 'Retry',
311             isPersistent: true,
312             redirect: 'refresh',
313             type: 'error',
314         };
315     });
316 },
317 secondsToHms(value) {
318     return new Date(value * 1000).toISOString().substr(11, 8);
319 },
320 showInfo(video) {
321     this.selectedVideo = video;
322     this.showInfoDlg = true;
323 },
324 initDeleteVideo(video) {
325     this.yesNoConfig = {
326         show: true,
327         message: `Are you sure you want to delete ${video.name}?`,
328         title: 'Delete Video',
329         onYes: [this.deleteVideo, video],
330         onNo: [
331             () => {

```

```

332         this.yesNoConfig.show = false;
333     },
334     null,
335 ],
336     persistent: true,
337 };
338 },
339 async deleteVideo(video) {
340     this.yesNoConfig.show = false;
341     this.lConfig = {
342         show: 'true',
343         message: 'Deleting your video..',
344     };
345     const { Convert } = this.$FeathersVuex.api;
346     const result = await Convert.get(video._id);
347     result
348         .remove()
349         .then(() => {
350             this.dConfig = {
351                 show: true,
352                 title: 'Sucess',
353                 message: 'Video deleted successfully.',
354                 buttonText: 'OK',
355                 isPersistent: true,
356                 redirect: 'refresh',
357                 type: 'info',
358             };
359         })
360         .catch(() => {
361             this.dConfig = {
362                 show: true,
363                 title: 'Error',
364                 message: 'Some error has occured while deleting your video.',
365                 buttonText: 'Retry',
366                 isPersistent: true,
367                 redirect: 'refresh',
368                 type: 'error',
369             };
370         });
371 },
372 async setupPagination(params) {
373     const { Convert } = this.$FeathersVuex.api;
374     const convertParams = {
375         query: {
376             $sort: { createdAt: 1 },
377             $limit: 0,
378             $skip: this.page - 1,
379             ...params,
380         },
381     };
382     Convert.find(convertParams)
383         .then((result) => {
384             this.paginateLength = parseInt(result.total / this.queryLimit, 10);
385         })
386         .catch(() => {});

```

```

387     },
388     async getVideos(params) {
389         const { Convert } = this.$FeathersVuex.api;
390         const convertParams = {
391             query: {
392                 $sort: { createdAt: 1 },
393                 $limit: this.queryLimit,
394                 $skip: this.page - 1,
395                 ...params,
396             },
397         };
398         Convert.find(convertParams)
399             .then((result) => {
400                 this.videos = result.data;
401             })
402             .catch(() => {});
403         await this.setupPagination(params);
404     },
405 },
406 watch: {
407     queryDate(val) {
408         const date = new Date(val);
409         this.queryDateSuccessor = date.setDate(date.getDate() + 1);
410     },
411     page() {
412         this.getVideos([]);
413     },
414 },
415 components: {
416     YesNoDialog,
417     Dialog,
418     Loading,
419 },
420 name: 'ConversionHistory',
421 };
422 </script>

```

/src/views/Login.vue

```

1 <template>
2   <v-main>
3     <v-row align="center" justify="center" class="fill-height">
4       <v-col cols="8" md="4">
5         <v-carousel
6           hide-delimiters
7           :show-arrows="false"
8           :value="carouselVal"
9           height="500px"
10          fluid
11        >
12          <!-- Login Carousel -->
13          <v-carousel-item>
14            <v-card class="elevation-12">
15              <!-- Top Bar -->
16              <v-toolbar color="primary" dark flat>

```

```

17     <v-toolbar-title>Login to your account.</v-toolbar-title>
18     <v-spacer />
19
20     <!-- Login ToolTip -->
21     <v-tooltip right>
22         <template v-slot:activator="{ on }">
23             <v-btn icon large v-on="on">
24                 <v-icon>mdi-login</v-icon>
25             </v-btn>
26         </template>
27         <span>Login</span>
28     </v-tooltip>
29 </v-toolbar>
30
31 <!-- Card Content -->
32 <v-card-text>
33     <!-- Login Form -->
34     <v-form v-model="valid" ref="form">
35         <!-- Email -->
36         <v-text-field
37             label="Email"
38             name="email"
39             prepend-icon="mdi-email"
40             type="email"
41             hint="Your existing email"
42             v-model="user.email"
43             maxlength="30"
44             counter="30"
45             clearable
46             :rules="emailRules"
47         />
48
49         <!-- Password -->
50         <v-text-field
51             id="password"
52             label="Password"
53             name="password"
54             prepend-icon="mdi-lock"
55             :type="showPassword ? 'text' : 'password'"
56             hint="Choose a secret password"
57             maxlength="30"
58             v-model="user.password"
59             :rules="notEmptyRules"
60             counter="30"
61             :append-icon="showPassword ? 'mdi-eye' : 'mdi-eye-off'"
62             @click:append="showPassword = !showPassword"
63             clearable
64         />
65     </v-form>
66 </v-card-text>
67 <v-card-actions>
68     <v-spacer />
69     <v-btn color="error" class="ma-2" v-on:click="setCarouselVal(1)">
70         Forgot Password?
71     </v-btn>

```

```

72         <v-btn color="primary" class="ma-2" v-on:click="login">
73             <v-icon left>mdi-login</v-icon>Login
74         </v-btn>
75     </v-card-actions>
76 </v-card>
77 </v-carousel-item>
78 <!-- Forgot Password Carousel -->
79 <v-carousel-item>
80     <v-card class="elevation-12">
81         <!-- Top Bar -->
82         <v-toolbar color="primary" dark flat>
83             <v-toolbar-title>Recover Your Password.</v-toolbar-title>
84             <v-spacer />
85
86         <!-- Login ToolTip -->
87         <v-tooltip right>
88             <template v-slot:activator="{ on }">
89                 <v-btn icon large v-on="on">
90                     <v-icon>mdi-key</v-icon>
91                 </v-btn>
92             </template>
93             <span>Recover Your Password</span>
94         </v-tooltip>
95     </v-toolbar>
96
97     <!-- Card Content -->
98     <v-card-text>
99         <!-- Login Form -->
100        <v-form v-model="valid" ref="formRecovery">
101            <!-- Email -->
102            <v-text-field
103                label="Email"
104                name="email"
105                prepend-icon="mdi-email"
106                type="email"
107                hint="Your existing email"
108                v-model="user.email"
109                maxlength="30"
110                counter="30"
111                clearable
112                :rules="emailRules"
113            />
114        </v-form>
115    </v-card-text>
116    <v-card-actions>
117        <v-btn color="error" class="ma-2" v-on:click="setCarouselVal(0)">
118            <v-icon size="30" left> mdi-keyboard-backspace </v-icon>
119        </v-btn>
120        <v-spacer />
121        <v-btn color="primary" class="ma-2" v-on:click="sendResetMail">
122            <v-icon left>mdi-mail</v-icon>Send Reset Mail
123        </v-btn>
124    </v-card-actions>
125 </v-card>
126 </v-carousel-item>

```

```

127         </v-carousel>
128         <Dialog :config="dConfig" />
129         <Loading :config="lConfig" v-if="isAuthenticatePending" />
130     </v-col>
131 </v-row>
132 </v-main>
133 </template>
134
135 <script>
136 import { mapState, mapActions } from 'vuex';
137 import Dialog from '@components/Dialog.vue';
138 import Loading from '@components/Loading.vue';
139
140 export default {
141     name: 'Login',
142     data: () => ({
143         carouselVal: 0,
144         dConfig: {},
145         lConfig: {
146             isPersistent: true,
147             show: 'true',
148             message: 'Loggin in...',
149         },
150         user: {
151             email: '',
152             password: '',
153         },
154         showPassword: false,
155         valid: false,
156         notEmptyRules: [(v) => v.length > 0 || 'Field cannot be empty'],
157         emailRules: [(v) => /.+@.+.test(v) || 'E-mail must be valid'],
158     }),
159     props: {
160         source: String,
161     },
162     components: {
163         Dialog,
164         Loading,
165     },
166     methods: {
167         sendResetMail() {
168             const { AuthManagement } = this.$FeathersVuex.api;
169             const data = {
170                 action: 'sendResetPwd',
171                 value: {
172                     email: this.user.email,
173                 },
174             };
175             const authRequest = new AuthManagement(data);
176             authRequest
177                 .save()
178                 .then(() => {
179                     this.dConfig = {
180                         show: true,
181                         type: 'info',

```

```

182         title: 'Sucess',
183         message: 'Please check your email for a password reset link!',
184         isPersistent: true,
185         redirect: '/login',
186     };
187     this.carouselVal = 0;
188 })
189 .catch(() => {
190     this.dConfig = {
191         show: true,
192         title: 'An error occured',
193         message: 'Please check your email.',
194         buttonText: 'Retry',
195         isPersistent: true,
196         type: 'error',
197     };
198 });
199 },
200 setCarouselVal(val) {
201     this.carouselVal = val;
202 },
203 login() {
204     if (this.$refs.form.validate()) {
205         this.authenticate({
206             strategy: 'local',
207             ...this.user,
208         })
209         .then((data) => {
210             if (data.user.isVerified === false) {
211                 const er = new Error();
212                 er.code = 400;
213                 throw er;
214             }
215             this.dConfig = {
216                 show: true,
217                 type: 'info',
218                 title: 'Sucess',
219                 message: 'Sucessfully logged in!',
220                 isPersistent: true,
221                 redirect: '/videos',
222             };
223         })
224         .catch((error) => {
225             if (error.code === 401) {
226                 this.dConfig = {
227                     show: true,
228                     title: 'An error occured',
229                     message: 'User not found. Please check your email and password.',
230                     buttonText: 'Retry',
231                     isPersistent: true,
232                     type: 'error',
233                 };
234             } else if (error.code === 403) {
235                 this.dConfig = {
236                     show: true,

```



```

237         title: 'An error occurred',
238         message:
239             'It looks like your email is not verified yet. Please verify your email in o
240         buttonText: 'Retry',
241         isPersistent: true,
242         type: 'error',
243     };
244     this.logout();
245 } else {
246     this.dConfig = {
247         show: true,
248         title: 'An error occurred',
249         message: 'Unknown server error has occurred.',
250         buttonText: 'Retry',
251         isPersistent: true,
252         type: 'error',
253     };
254     this.logout();
255 }
256 });
257 }
258 },
259 ...mapActions('auth', ['authenticate', 'logout']),
260 },
261 computed: {
262     ...mapState('auth', ['isAuthenticatePending']),
263 },
264 };
265 </script>

```

/src/views/Reset.vue

```

1 <template>
2 <v-main>
3 <v-row align="center" justify="center" class="fill-height">
4 <v-col cols="12" sm="8" md="4">
5 <v-card class="elevation-12">
6 <!-- Top Bar -->
7 <v-toolbar color="primary" dark flat>
8 <v-toolbar-title>Recover Your Password.</v-toolbar-title>
9 <v-spacer />
10
11 <!-- Login ToolTip -->
12 <v-tooltip right>
13 <template v-slot:activator="{ on }">
14 <v-btn icon large v-on="on">
15 <v-icon>mdi-key</v-icon>
16 </v-btn>
17 </template>
18 <span>Recover Your Password</span>
19 </v-tooltip>
20 </v-toolbar>
21
22 <!-- Card Content -->
23 <v-card-text>

```

```

24     <!-- Recovery Form -->
25     <v-form v-model="valid" ref="formRecovery">
26         <!-- Password -->
27         <v-text-field
28             id="password"
29             label="Password"
30             name="password"
31             prepend-icon="mdi-lock"
32             :type="showPassword ? 'text' : 'password'"
33             hint="Choose a secret password"
34             maxlength="30"
35             v-model="password"
36             :rules="passwordRules"
37             counter="30"
38             :append-icon="showPassword ? 'mdi-eye' : 'mdi-eye-off'"
39             @click:append="showPassword = !showPassword"
40             clearable
41         />
42
43         <!-- Confirm Password -->
44         <v-text-field
45             id="confirmPassword"
46             label="Confirm Password"
47             name="confirmPassword"
48             hint="Re-enter the password"
49             prepend-icon="mdi-refresh"
50             :type="showConfirmPassword ? 'text' : 'password'"
51             v-model="confirmPassword"
52             :rules="confirmPasswordRules"
53             maxlength="30"
54             counter="30"
55             clearable
56             :append-icon="showConfirmPassword ? 'mdi-eye' : 'mdi-eye-off'"
57             @click:append="showConfirmPassword = !showConfirmPassword"
58         />
59     </v-form>
60 </v-card-text>
61 <v-card-actions>
62     <v-spacer />
63     <v-btn color="primary" v-on:click="beginReset" class="ma-2">
64         <v-icon left>mdi-key</v-icon>Reset Password
65     </v-btn>
66 </v-card-actions>
67 </v-card>
68 </v-col>
69 </v-row>
70 <Dialog :config="dConfig" />
71 <Loading :config="lConfig" v-if="isResetPending" />
72 </v-main>
73 </template>
74
75 <script>
76 import Dialog from '@components/Dialog.vue';
77 import Loading from '@components/Loading.vue';
78

```

```

79 export default {
80   name: 'Reset',
81   data: (vm) => ({
82     valid: false,
83     password: 'Abc@1234',
84     confirmPassword: 'Abc@1234',
85     showPassword: false,
86     showConfirmPassword: false,
87     passwordRules: [
88       (v) => v.length >= 8 || 'Password length must be greater than 8 characters',
89       /* eslint-disable */
90       (v) =>
91         /^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*{8,})/.test(v) ||
92         'Password must contain a nummber, a lowercase alphabet and an uppercase alphabet',
93       /* eslint-enable */
94     ],
95     confirmPasswordRules: [(v) => v === vm.password || 'Passwords does not match'],
96     isResetPending: false,
97     dConfig: {},
98     lConfig: {
99       isPersistent: true,
100       show: 'true',
101       message: 'Please wait while we change your password!',
102     },
103   }),
104   methods: {
105     beginReset() {
106       const url = new URL(window.location.href);
107       const token = url.searchParams.get('token');
108       const resetData = {
109         action: 'resetPwdLong',
110         value: {
111           token,
112           password: this.password,
113         },
114       };
115       const { AuthManagement } = this.$FeathersVuex.api;
116       const authRequest = new AuthManagement(resetData);
117       authRequest
118         .save()
119         .then(() => {
120           this.dConfig = {
121             show: true,
122             title: 'Sucess',
123             message: 'Password changed successfully!',
124             buttonText: 'Continue',
125             isPersistent: true,
126             redirect: '/login',
127             type: 'info',
128           };
129         })
130         .catch(() => {
131           this.dConfig = {
132             show: true,
133             title: 'Error',

```

```

134         message: 'Some error has occurred while changing your password.',
135         buttonText: 'OK',
136         isPersistent: true,
137         redirect: '/login',
138         type: 'error',
139     };
140 });
141 },
142 },
143 components: {
144     Dialog,
145     Loading,
146 },
147 };
148 </script>

```

/src/views/SignUp.vue

```

1 <template>
2   <v-main>
3     <v-row align="center" justify="center" class="fill-height">
4       <v-col cols="12" sm="8" md="4">
5         <v-card class="elevation-12">
6           <!-- Top Bar -->
7           <v-toolbar color="primary" dark flat>
8             <v-toolbar-title>Create a new account.</v-toolbar-title>
9             <v-spacer />
10
11           <!-- Login ToolTip -->
12           <v-tooltip right>
13             <template v-slot:activator="{ on }">
14               <v-btn icon large v-on="on">
15                 <v-icon>mdi-account</v-icon>
16               </v-btn>
17             </template>
18             <span>Sign Up</span>
19           </v-tooltip>
20         </v-toolbar>
21
22         <!-- Card Content -->
23         <v-card-text>
24           <v-row justify="space-around">
25             <v-avatar color="indigo" size="150">
26               <v-icon size="100" dark v-if="user.avatar === ''"
27                 >mdi-account-circle</v-icon>
28             >
29             
30           </v-avatar>
31         </v-row>
32
33         <!-- Sign Form -->
34         <v-form v-model="valid" ref="form">
35           <!-- Display Name -->
36           <v-text-field
37             label="Name"

```

```

38         name="name"
39         prepend-icon="mdi-face-profile"
40         clearable
41         counter="30"
42         :rules="notEmptyRules"
43         v-model="user.displayName"
44         hint="Your friendly name"
45         maxlength="30"
46         type="text"
47     />
48
49     <!-- Email -->
50     <v-text-field
51         label="Email"
52         name="email"
53         prepend-icon="mdi-email"
54         type="email"
55         hint="Your existing email"
56         v-model="user.email"
57         maxlength="30"
58         counter="30"
59         clearable
60         :rules="emailRules"
61     />
62
63     <!-- Password -->
64     <v-text-field
65         id="password"
66         label="Password"
67         name="password"
68         prepend-icon="mdi-lock"
69         :type="showPassword ? 'text' : 'password'"
70         hint="Choose a secret password"
71         maxlength="30"
72         v-model="user.password"
73         :rules="passwordRules"
74         counter="30"
75         :append-icon="showPassword ? 'mdi-eye' : 'mdi-eye-off'"
76         @click:append="showPassword = !showPassword"
77         clearable
78     />
79
80     <!-- Confirm Password -->
81     <v-text-field
82         id="confirmPassword"
83         label="Confirm Password"
84         name="confirmPassword"
85         hint="Re-enter the password"
86         prepend-icon="mdi-refresh"
87         :type="showConfirmPassword ? 'text' : 'password'"
88         v-model="user.confirmPassword"
89         :rules="confirmPasswordRules"
90         maxlength="30"
91         counter="30"
92         clearable

```

```

93         :append-icon="showConfirmPassword ? 'mdi-eye' : 'mdi-eye-off'"
94         @click:append="showConfirmPassword = !showConfirmPassword"
95     />
96
97     <!-- Date Of Birth -->
98     <v-menu
99         ref="menu"
100         v-model="menu"
101         :close-on-content-click="false"
102         :return-value.sync="user.dob"
103         transition="scale-transition"
104         offset-y
105         min-width="290px"
106     >
107         <template v-slot:activator="{ on }">
108             <v-text-field
109                 v-model="user.dob"
110                 label="Date Of Birth"
111                 prepend-icon="mdi-calendar"
112                 readonly
113                 v-on="on"
114             ></v-text-field>
115         </template>
116         <v-date-picker v-model="user.dob" no-title scrollable>
117             <v-spacer></v-spacer>
118             <v-btn text color="primary" @click="$refs.menu.save(user.dob)"
119                 >OK</v-btn>
120         >
121     </v-date-picker>
122 </v-menu>
123
124     <!-- Avatar -->
125     <v-file-input
126         :rules="avatarRules"
127         accept="image/png, image/jpeg, image/bmp"
128         placeholder="Pick an avatar"
129         prepend-icon="mdi-camera"
130         label="Avatar"
131         v-model="avatarFile"
132         v-on:change="onFilePicked"
133     ></v-file-input>
134 </v-form>
135 </v-card-text>
136
137 <v-card-actions>
138     <v-spacer />
139     <v-btn color="primary" v-on:click="validate" class="ma-2">
140         <v-icon left>mdi-account-plus-outline</v-icon>Sign Up
141     </v-btn>
142 </v-card-actions>
143 </v-card>
144 </v-col>
145 </v-row>
146 <Dialog :config="dConfig" />
147 <Loading :config="lConfig" v-if="isCreatePending" />

```

```

148     </v-main>
149 </template>
150
151 <script>
152 import Dialog from '@components/Dialog.vue';
153 import Loading from '@components/Loading.vue';
154 import { mapState } from 'vuex';
155
156 export default {
157   name: 'SignUp',
158   data: (vm) => ({
159     dConfig: {},
160     lConfig: {
161       isPersistent: true,
162       show: 'true',
163       message: 'Signing you up!',
164     },
165     valid: false,
166     avatarFile: null,
167     user: {
168       displayName: '',
169       email: '',
170       password: '',
171       confirmPassword: '',
172       dob: new Date().toISOString().substr(0, 10),
173       avatar: '',
174     },
175     showPassword: false,
176     showConfirmPassword: false,
177     menu: false,
178     avatarRules: [
179       (v) => !!v || 'Please choose an avatar',
180       (v) => !v || v.size < 2000000 || 'Avatar size should be less than 2 MB!',
181     ],
182     notEmptyRules: [(v) => v.length > 0 || 'Field cannot be empty'],
183     emailRules: [(v) => /.+@.+.test(v) || 'E-mail must be valid'],
184     passwordRules: [
185       (v) => v.length >= 8 || 'Password length must be greater than 8 characters',
186       /* eslint-disable */
187       (v) =>
188         /^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*{8,})/.test(v) ||
189         'Password must contain a nummber, a lowercase alphabet and an uppercase alphabet',
190       /* eslint-enable */
191     ],
192     confirmPasswordRules: [(v) => v === vm.user.password || 'Passwords does not match'],
193   }),
194   props: {
195     source: String,
196   },
197   methods: {
198     validate() {
199       if (this.$refs.form.validate()) {
200         const { User } = this.$FeathersVuex.api;
201         const newUser = new User(this.user);
202         newUser

```

```

203     .save()
204     .then(() => {
205         this.dConfig = {
206             show: true,
207             title: 'Sucess',
208             message:
209                 'Sign up sucessful! Please Check Your Email For The Verification Link.',
210             buttonText: 'Log In',
211             isPersistent: true,
212             redirect: '/login',
213             type: 'info',
214         };
215     })
216     .catch((error) => {
217         if (error.code === 409) {
218             this.dConfig = {
219                 show: true,
220                 title: 'An error occured',
221                 message: 'An account with the same e-mail already exists.',
222                 isPersistent: true,
223                 type: 'error',
224             };
225         } else {
226             this.dConfig = {
227                 show: true,
228                 title: 'An error occured',
229                 message: 'A server error has occured.',
230                 isPersistent: true,
231                 type: 'error',
232             };
233         }
234     });
235 }
236 },
237 onFilePicked(file) {
238     if (file) {
239         const fileReader = new FileReader();
240         fileReader.addEventListener('load', () => {
241             this.user.avatar = fileReader.result;
242         });
243         fileReader.readAsDataURL(file);
244     } else {
245         this.user.avatar = '';
246     }
247 },
248 },
249 components: {
250     Dialog,
251     Loading,
252 },
253 computed: {
254     ...mapState('users', ['isCreatePending']),
255 },
256 };
257 </script>

```



```
1 <template>
2   <v-card>
3     <Dialog :config="dConfig" />
4     <Loading :config="lConfig" v-if="isVerificationPending" />
5   </v-card>
6 </template>
7
8 <script>
9 import Dialog from '@components/Dialog.vue';
10 import Loading from '@components/Loading.vue';
11
12 export default {
13   name: 'Verify',
14   methods: {
15     beginVerify() {
16       const { AuthManagement } = this.$FeathersVuex.api;
17       const url = new URL(window.location.href);
18       const token = url.searchParams.get('token');
19       const verificationData = {
20         action: 'verifySignupLong',
21         value: token,
22       };
23       const authRequest = new AuthManagement(verificationData);
24       authRequest
25         .save()
26         .then(() => {
27           this.isVerificationPending = false;
28           this.dConfig = {
29             show: true,
30             title: 'Success',
31             message: 'Verification Successful! Please Login To Continue.',
32             buttonText: 'Log In',
33             isPersistent: true,
34             redirect: '/login',
35             type: 'info',
36           };
37         })
38         .catch(() => {
39           this.dConfig = {
40             show: true,
41             title: 'Error',
42             message: 'Some Error Has Occured In The Verification Process!.',
43             buttonText: 'OK',
44             isPersistent: true,
45             redirect: '/login',
46             type: 'error',
47           };
48         });
49     },
50   },
51   beforeMount() {
52     this.beginVerify();
53   },
54   data: () => ({
```

```

55     isVerificationPending: true,
56     dConfig: {},
57     lConfig: {
58         isPersistent: true,
59         show: 'true',
60         message: 'Please Wait While We Verify You!',
61     },
62 },
63 components: {
64     Dialog,
65     Loading,
66 },
67 };
68 </script>

```

/src/views/Videos.vue

```

1  <template>
2    <v-main>
3      <v-container class="fill-height" fluid>
4        <v-row class="ma-0 pa-0 fill-height">
5          <v-col cols="12" sm="12" offset-sm="3" class="ma-0 pa-0 fill-height">
6            <v-card class="ma-0 pa-0 fill-height">
7              <v-toolbar color="primary" dark>
8                <v-toolbar-title class="white--text">My Videos</v-toolbar-title>
9                <v-spacer />
10               <v-icon x-large>mdi-video</v-icon>
11               <template v-slot:extension>
12                 <v-btn
13                   fab
14                   color="red accent-2"
15                   bottom
16                   left
17                   absolute
18                   @click="dialog = !dialog"
19                 >
20                   <v-icon>mdi-plus</v-icon>
21                 </v-btn>
22               </template>
23             </v-toolbar>
24             <v-row class="mt-8">
25               <v-text-field
26                 placeholder="Search Video..."
27                 v-model="filterString"
28                 rounded
29                 dense
30                 filled
31                 class="ml-5 my-2"
32                 prepend-inner-icon="mdi-magnify"
33               />
34               <v-btn fab class="mx-5" @click="getVideos(generateParams())">
35                 <v-icon>mdi-movie-search</v-icon>
36               </v-btn>
37               <v-menu
38                 ref="menu"

```

```

39     v-model="menu"
40     :close-on-content-click="false"
41     :return-value.sync="queryDate"
42     transition="scale-transition"
43     offset-y
44     min-width="290px"
45   >
46     <template v-slot:activator="{ on }">
47       <v-btn fab v-on="on" class="mx-2">
48         <v-icon>mdi-calendar</v-icon>
49       </v-btn>
50     </template>
51     <v-date-picker v-model="queryDate" no-title scrollable class="ma-2">
52       <v-spacer></v-spacer>
53       <v-btn
54         text
55         color="primary"
56         @click="$refs.menu.save(queryDate) & getVideos(generateParams())"
57         >OK</v-btn>
58     >
59   </v-date-picker>
60 </v-menu>
61 <v-tooltip bottom>
62   <template v-slot:activator="{ on, attrs }">
63     <v-btn fab v-on="on" class="mx-2 mr-5" @click="getVideos({})">
64       <v-icon color="error" v-bind="attrs">mdi-restore</v-icon>
65     </v-btn>
66   </template>
67   <span>Clear Search Parameters</span>
68 </v-tooltip>
69 </v-row>
70 <div class="text-center" style="position: relative; bottom: 0">
71   <v-pagination
72     v-model="page"
73     :length="paginateLength == 0 ? 1 : paginateLength"
74     circle
75   ></v-pagination>
76 </div>
77 <v-list two-line v-if="viewMode == 'List'">
78   <v-list-item v-for="item in videos" :key="item._id">
79     <v-list-item-avatar>
80       <v-icon class="blue">mdi-video</v-icon>
81     </v-list-item-avatar>
82     <v-list-item-content>
83       <v-list-item-title>{{ item.originalName }}</v-list-item-title>
84       <v-list-item-subtitle>
85         {{ new Date(item.createdAt).toUTCString() }}
86       </v-list-item-subtitle>
87     </v-list-item-content>
88     <v-btn icon ripple>
89       <v-icon color="grey lighten-1" @click="play(item)">mdi-play</v-icon>
90     </v-btn>
91     <v-btn icon ripple>
92       <v-icon color="grey lighten-1" @click="showInfo(item)">
93       >mdi-information</v-icon>

```

```

94         >
95     </v-btn>
96     <v-btn icon ripple>
97         <v-icon color="grey lighten-1" @click="initConvert(item)"
98             >mdi-refresh</v-icon>
99         >
100     </v-btn>
101     <v-btn icon ripple>
102         <v-icon color="grey lighten-1" @click="confirmDeleteVideo(item)">
103             mdi-delete
104         </v-icon>
105     </v-btn>
106 </v-list-item>
107 </v-list>
108 <v-container fluid v-if="viewMode == 'Card'">
109     <v-row>
110         <v-card
111             v-for="item in videos"
112             :key="item._id"
113             max-width="336px"
114             color="primary"
115             elevation="24"
116             class="mx-5"
117         >
118             <v-img
119                 :src="'data:image/png;base64,' + item.thumbnail"
120                 class="white--text align-end"
121                 height="188px"
122                 width="336px"
123             >
124                 <v-card-title v-text="item.originalName"></v-card-title>
125             </v-img>
126
127             <v-card-actions>
128                 <v-spacer></v-spacer>
129                 <v-btn icon @click="play(item)">
130                     <v-icon>mdi-play</v-icon>
131                 </v-btn>
132
133                 <v-btn icon @click="showInfo(item)">
134                     <v-icon>mdi-information</v-icon>
135                 </v-btn>
136                 <v-btn icon @click="initConvert(item)">
137                     <v-icon>mdi-refresh</v-icon>
138                 </v-btn>
139                 <v-btn icon @click="confirmDeleteVideo(item)">
140                     <v-icon>mdi-delete</v-icon>
141                 </v-btn>
142             </v-card-actions>
143         </v-card>
144     </v-row>
145 </v-container>
146 <v-dialog v-model="isUploading" max-width="400px" persistent>
147     <v-card loading class="text-center">
148         <v-card-title>Uploading Your Video...</v-card-title>

```

```

149     <v-card-text>
150         <v-divider></v-divider>
151         <br />
152         <v-progress-circular
153             :rotate="360"
154             :size="100"
155             :width="15"
156             :value="uploadProgress"
157             color="primary"
158             >{{ uploadProgress }}</v-progress-circular>
159     >
160 </v-card-text>
161 </v-card>
162 </v-dialog>
163 <Dialog :config="dConfig" />
164 <Loading :config="lConfig" />
165 <v-dialog v-model="dialog" max-width="500px">
166     <v-card>
167         <v-card-title>Upload A New Video</v-card-title>
168         <v-card-text>
169             <v-form v-model="valid" ref="uploadForm">
170                 <v-file-input
171                     :rules="videoRules"
172                     show-size
173                     counter
174                     label="Select a video to be converted"
175                     prepend-icon="mdi-video"
176                     accept="video/*"
177                     v-model="video.files"
178                 ></v-file-input>
179                 <p class="caption grey--text text--darken-1">
180                     This video size must be less than 500MB
181                 </p>
182                 <p class="caption grey--text text--darken-1">
183                     Currently *.mp4, *.avi, *.flv, *.ogm, *.wmv, *.mpg, *.ogv, *.mov,
184                     *.asx, *.mpeg and *.m4v are supported.
185                 </p>
186                 <v-text-field
187                     label="Description"
188                     name="description"
189                     prepend-icon="mdi-text"
190                     clearable
191                     counter="100"
192                     :rules="notEmptyRules"
193                     v-model="video.description"
194                     hint="A short description of your video"
195                     maxlength="100"
196                     type="text"
197                 />
198             </v-form>
199         </v-card-text>
200         <v-card-actions>
201             <v-spacer></v-spacer>
202             <v-btn color="primary" @click="validate()">
203                 <v-icon left>mdi-upload</v-icon>Upload

```

```

204         </v-btn>
205     </v-card-actions>
206 </v-card>
207 </v-dialog>
208 <YesNoDialog :config="yesNoConfig" />
209 <v-dialog v-model="playVideo" max-width="50%">
210     <video :src="videoData" controls :type="videoMime" autoplay></video>
211 </v-dialog>
212 <v-dialog v-model="showInfoDlg" max-width="500px">
213     <v-card>
214         <v-card-title>Video Information</v-card-title>
215         <v-card-text>
216             <v-simple-table>
217                 <thead>
218                     <tr>
219                         <th class="text-left">Attribute</th>
220                         <th class="text-left">Value</th>
221                     </tr>
222                 </thead>
223                 <tbody v-if="this.selectedVideo != null">
224                     <tr>
225                         <td>Name</td>
226                         <td>{{ this.selectedVideo.originalName }}</td>
227                     </tr>
228                     <tr>
229                         <td>Description</td>
230                         <td>{{ this.selectedVideo.description }}</td>
231                     </tr>
232                     <tr>
233                         <td>Format</td>
234                         <td>{{ this.selectedVideo.format }}</td>
235                     </tr>
236                     <tr>
237                         <td>Size</td>
238                         <td>
239                             {{
240                                 parseFloat(this.selectedVideo.fileSize / 1000000.0).toFixed(2)
241                             }}
242                             MB
243                         </td>
244                     </tr>
245                     <tr>
246                         <td>Duration</td>
247                         <td>{{ this.selectedVideo.duration }}</td>
248                     </tr>
249                     <tr>
250                         <td>Resolution</td>
251                         <td>{{ this.selectedVideo.resolution }}</td>
252                     </tr>
253                     <tr>
254                         <td>Frames Per Second</td>
255                         <td>{{ this.selectedVideo.frameRate }}</td>
256                     </tr>
257                     <tr>
258                         <td>Video Bitrate</td>

```

```

259         <td>{{ this.selectedVideo.videoBitrate }} kb/s</td>
260     </tr>
261     <tr>
262         <td>Audio Bitrate</td>
263         <td>{{ this.selectedVideo.audioBitrate }} kb/s</td>
264     </tr>
265 </tbody>
266 </v-simple-table>
267 </v-card-text>
268 <v-card-actions>
269     <v-spacer></v-spacer>
270     <v-btn class="ma-2" color="primary" v-on:click="showInfoDlg = false">
271         Close
272     </v-btn>
273 </v-card-actions>
274 </v-card>
275 </v-dialog>
276 <v-dialog v-model="convertVideo" max-width="50%">
277     <v-card>
278         <v-card-title>Convert Video</v-card-title>
279         <v-card-text>
280             <v-form v-model="valid" ref="convertForm">
281                 <v-row>
282                     <v-col>
283                         <v-text-field
284                             label="Name"
285                             name="name"
286                             prepend-icon="mdi-rename-box"
287                             type="text"
288                             hint="Name of the converted file"
289                             v-model="convert.name"
290                             maxlength="30"
291                             counter="30"
292                             clearable
293                             :rules="notEmptyRules"
294                         />
295                     <section v-if="convert.type === 'video'">
296                         <span>Resolution</span>
297                         <v-radio-group v-model="resType">
298                             <v-row>
299                                 <v-col>
300                                     <v-radio label="Standard Resolution" value="standard" />
301                                 </v-col>
302                                 <v-col>
303                                     <v-radio label="Custom Resolution" value="custom" />
304                                 </v-col>
305                             </v-row>
306                         </v-radio-group>
307                         <v-row v-if="resType === 'custom'">
308                             <v-col>
309                                 <v-text-field
310                                     label="Width"
311                                     clearable
312                                     suffix="px"
313                                     v-model="convert.width"

```

```

314         :rules="dimRules"
315     ></v-text-field>
316 </v-col>
317 <v-col>
318     <v-text-field
319         label="Height"
320         clearable
321         suffix="px"
322         v-model="convert.height"
323         :rules="dimRules"
324     ></v-text-field>
325 </v-col>
326 </v-row>
327 <v-row v-if="resType === 'standard'">
328     <v-col cols="5">
329         <v-select
330             v-model="aspectRatio"
331             :rules="notEmptyRules"
332             :items="aspectRatios"
333             label="Select aspect ratio"
334         ></v-select>
335     </v-col>
336     <v-col>
337         <v-select
338             v-model="selectedStandardResolution"
339             :rules="notEmptyRules"
340             :items="standardResolutions"
341             label="Select a standard resolution"
342         ></v-select>
343     </v-col>
344 </v-row>
345 <p>Frame Rate</p>
346 <v-slider
347     v-model="convert.frameRate"
348     class="mt-10"
349     track-color="grey"
350     min="1"
351     thumb-label="always"
352     :max="selectedVideo ? selectedVideo.frameRate : 100"
353 >
354 </v-slider>
355 <v-checkbox
356     label="Disable Audio"
357     v-model="convert.isAudioDisabled"
358     :disabled="
359         selectedVideo ? selectedVideo.audioBitrate == 0 : false
360     "
361 ></v-checkbox>
362 </section>
363 <div>
364     <p>Audio Quality</p>
365     <v-slider
366         v-model="convert.audioQuality"
367         class="mt-10"
368         track-color="grey"

```



```

369         min="1"
370         thumb-label="always"
371         :max="100"
372         :disabled="
373             selectedVideo ? selectedVideo.audioBitrate == 0 : false
374         "
375     >
376     </v-slider>
377 </div>
378 </v-col>
379 <v-col>
380 <p>Convert To</p>
381 <v-radio-group v-model="convert.type">
382     <v-row>
383         <v-col>
384             <v-radio label="Video" value="video"></v-radio>
385         </v-col>
386         <v-col>
387             <v-radio label="Audio" value="audio"></v-radio>
388         </v-col>
389     </v-row>
390     <v-spacer />
391     <v-select
392         v-model="convert.format"
393         :rules="notEmptyRules"
394         :items="
395             convert.type === 'video' ? videoFormats : audioFormats
396         "
397         label="Select the output file format"
398     ></v-select>
399 </v-radio-group>
400 <p>Duration</p>
401 <v-row>
402     <v-col>
403         <v-card>
404             
409         </v-card>
410     </v-col>
411     <v-col> </v-col>
412     <v-col>
413         <v-card>
414             
419         </v-card>
420     </v-col>
421 </v-row>
422 <v-row>
423     <v-col>

```

```

424         <v-range-slider
425             v-model="convert.range"
426             :max="
427                 selectedVideo ? hmsToSeconds(selectedVideo.duration) : 0
428             "
429             :min="0"
430             hide-details
431             @change="getThumb()"
432             class="align-center"
433         >
434             <template v-slot:append>
435                 <span>{{ secondsToHms(convert.range[1]) }}</span>
436             </template>
437             <template v-slot:prepend>
438                 <span>{{ secondsToHms(convert.range[0]) }}</span>
439             </template>
440         </v-range-slider>
441     </v-col>
442 </v-row>
443 <div v-if="convert.type === 'video'">
444     <p>Video Quality</p>
445     <v-slider
446         v-model="convert.videoQuality"
447         class="mt-10"
448         track-color="grey"
449         min="1"
450         thumb-label="always"
451         :max="100"
452     >
453     </v-slider>
454 </div>
455 </v-col>
456 </v-row>
457 </v-form>
458 </v-card-text>
459 <v-card-actions>
460     <v-spacer></v-spacer>
461     <v-btn color="primary" @click="beginConvert()">
462         <v-icon left>mdi-refresh</v-icon>Convert
463     </v-btn>
464 </v-card-actions>
465 </v-card>
466 </v-dialog>
467 <v-speed-dial
468     absolute
469     v-model="fab"
470     bottom
471     right
472     direction="top"
473     transition="slide-y-reverse-transition"
474 >
475     <template v-slot:activator>
476         <v-btn v-model="fab" color="blue darken-2" dark fab>
477             <v-icon v-if="fab">mdi-close</v-icon>
478             <v-icon v-else>mdi-view-split-vertical</v-icon>

```

```

479         </v-btn>
480     </template>
481     <v-btn fab dark small color="green" @click="viewMode = 'List'">
482         <v-icon>mdi-view-list</v-icon>
483     </v-btn>
484     <v-btn fab dark small color="indigo" @click="viewMode = 'Card'">
485         <v-icon>mdi-view-comfy</v-icon>
486     </v-btn>
487 </v-speed-dial>
488 </v-card>
489 </v-col>
490 </v-row>
491 </v-container>
492 </v-main>
493 </template>
494 <script>
495 import axios from 'axios';
496 import Dialog from '@/components/Dialog.vue';
497 import Loading from '@/components/Loading.vue';
498 import YesNoDialog from '@/components/YesNoDialog.vue';
499 import mime from 'mime-types';
500
501 export default {
502   watch: {
503     playVideo(val) {
504       if (val === false) {
505         this.videoData = '';
506       }
507     },
508     queryDate(val) {
509       const date = new Date(val);
510       this.queryDateSuccessor = date.setDate(date.getDate() + 1);
511     },
512     page() {
513       this.getVideos([]);
514     },
515     selectedStandardResolution(value) {
516       const splitted = value.toString().split('x');
517       this.convert.height = parseInt(splitted[0], 10);
518       this.convert.width = parseInt(splitted[1], 10);
519     },
520     selectedVideo(value) {
521       this.convert.frameRate = value.frameRate;
522       this.convert.range[1] = this.hmsToSeconds(value.duration);
523     },
524   },
525   name: 'Videos',
526   data: () => ({
527     videos: [],
528     viewMode: 'List',
529     menu: false,
530     queryDateSuccessor: null,
531     queryDate: new Date().toISOString().substr(0, 10),
532     paginateLength: 0,
533     queryLimit: 10,

```

```

534     page: 1,
535     yesNoConfig: {},
536     fab: false,
537
538     videoMime: '',
539     thumbStart: null,
540     thumbEnd: null,
541     videoData: null,
542     playVideo: false,
543     filterString: '',
544     selectedStandardResolution: '',
545     testConfig: {
546       show: true,
547     },
548     selectedVideo: null,
549     resType: 'standard',
550     lConfig: {
551       isPersistent: true,
552       show: false,
553       message: 'Processing your video!',
554     },
555     videoFormats: ['avi', 'mp4', 'flv', 'webm', 'mov'],
556     audioFormats: ['mp3', 'wav', 'oga'],
557     convert: {
558       videoQuality: 100,
559       audioQuality: 100,
560       name: '',
561       format: '',
562       width: 0,
563       height: 0,
564       upload: null,
565       type: 'video',
566       isAudioDisabled: false,
567       frameRate: 10,
568       range: [null, null],
569     },
570     dConfig: {
571       show: false,
572       color: '',
573       title: '',
574       icon: '',
575       message: '',
576       buttonText: '',
577     },
578     uploadProgress: 0,
579     dialog: false,
580     convertVideo: false,
581     deleteVideoDlg: false,
582     video: {
583       files: null,
584       description: '',
585     },
586     notEmptyRules: [(v) => !!v || 'Field cannot be empty.'],
587     videoRules: [
588       (v) => !!v || 'Please choose a video',

```

```

589     (v) => !v || v.size < 500000000 || 'Video size should be less than 500 MB!',
590 ],
591 dimRules: [
592     (v) => !!v || 'Field cannot be empty.',
593     (v) => !v || Number.isInteger(Number.parseInt(v, 10)) || 'Invalid Dimension',
594 ],
595 valid: false,
596 isUploading: false,
597 isConverting: true,
598 showInfoDlg: false,
599 standardResolutions43: [
600     '640x480',
601     '800x600',
602     '960x720',
603     '1024x768',
604     '1280x960',
605     '1400x1050',
606     '1440x1080',
607     '1600x1200',
608     '1856x1392',
609     '1920x1440',
610     '2048x1536',
611 ],
612 standardResolutions1610: [
613     '1280x800',
614     '1440x900',
615     '1680x1050',
616     '1920x1200',
617     '2560x1600',
618 ],
619 standardResolutions169: [
620     '1024x576',
621     '1152x648',
622     '1280x720',
623     '1366x768',
624     '1600x900',
625     '1920x1080',
626     '2560x1440',
627     '3840x2160',
628     '7680x4320',
629 ],
630 aspectRatios: ['4:3', '16:10', '16:9'],
631 aspectRatio: '',
632 }),
633 mounted() {
634     this.getVideos();
635 },
636 computed: {
637     standardResolutions() {
638         switch (this.aspectRatio) {
639             /* eslint-disable */
640             case '4:3':
641                 return this.standardResolutions43;
642             case '16:10':
643                 return this.standardResolutions1610;

```

```

644         case '16:9':
645             return this.standardResolutions169;
646         default:
647             return [];
648         /* eslint-enable */
649     }
650 },
651 },
652 methods: {
653     generateParams() {
654         const params = {};
655         if (this.queryDateSuccessor) {
656             params.createdAt = {
657                 $gte: this.queryDate,
658                 $lt: this.queryDateSuccessor,
659             };
660         }
661         if (this.filterString) {
662             params.originalName = {
663                 $search: this.filterString,
664             };
665         }
666         return params;
667     },
668     async setupPagination(params) {
669         const { Uploads } = this.$FeathersVuex.api;
670         const uploadParams = {
671             query: {
672                 $sort: { createdAt: 1 },
673                 $limit: 0,
674                 $skip: this.page - 1,
675                 ...params,
676             },
677         };
678         Uploads.find(uploadParams)
679             .then((result) => {
680                 this.paginateLength = parseInt(result.total / this.queryLimit, 10);
681             })
682             .catch(() => {});
683     },
684     async getVideos(params) {
685         const { Uploads } = this.$FeathersVuex.api;
686         const uploadParams = {
687             query: {
688                 $sort: { createdAt: 1 },
689                 $limit: this.queryLimit,
690                 $skip: this.page - 1,
691                 ...params,
692             },
693         };
694         Uploads.find(uploadParams)
695             .then((result) => {
696                 this.videos = result.data;
697             })
698             .catch(() => {});

```

```

699     await this.setupPagination(params);
700   },
701   play(video) {
702     this.selectedVideo = video;
703     this.playVideo = true;
704     this.videoData = `${process.env.VUE_APP_SERVER_ADDR}/streams/${video._id}`;
705     this.videoMime = mime.lookup(this.selectedVideo.format);
706   },
707   getThumb() {
708     const { range } = this.convert;
709     const defVal = [0, this.hmsToSeconds(this.selectedVideo.duration)];
710     const [start, end] = range[0] != null ? range : defVal;
711     const factor = this.hmsToSeconds(this.selectedVideo.duration) / 30 + 1;
712     let startThumbId = parseInt(start / factor, 10);
713     let endThumbId = parseInt(end / factor, 10);
714     if (startThumbId <= 0) {
715       startThumbId = 1;
716     } else if (startThumbId > 30) {
717       startThumbId = 30;
718     }
719     if (endThumbId <= 0) {
720       endThumbId = 1;
721     } else if (endThumbId > 30) {
722       endThumbId = 30;
723     }
724     const { Thumbs } = this.$FeathersVuex.api;
725     Thumbs.get(this.selectedVideo._id, {
726       query: {
727         thumbId: startThumbId,
728       },
729     }).then((result) => {
730       this.thumbStart = result.data;
731     });
732     Thumbs.get(this.selectedVideo._id, {
733       query: {
734         thumbId: endThumbId,
735       },
736     }).then((result) => {
737       this.thumbEnd = result.data;
738     });
739   },
740   hmsToSeconds(value) {
741     const a = value.split(':');
742     const seconds = +a[0] * 60 * 60 + +a[1] * 60 + +a[2];
743     return seconds;
744   },
745   secondsToHms(value) {
746     return new Date(value * 1000).toISOString().substr(11, 8);
747   },
748   showInfo(video) {
749     this.showInfoDlg = true;
750     this.selectedVideo = video;
751   },
752   confirmDeleteVideo(video) {
753     this.yesNoConfig = {

```

```

754     show: true,
755     message: `Are you sure you want to delete ${video.originalName}?`,
756     title: 'Delete Video',
757     onYes: [this.deleteVideo, video],
758     onNo: [
759       () => {
760         this.yesNoConfig.show = false;
761       },
762       null,
763     ],
764     persistent: true,
765   };
766 },
767 initConvert(video) {
768   this.selectedVideo = video;
769   this.convert.name = `${video.originalName}_Converted`;
770   this.convertVideo = true;
771   this.convert.upload = video;
772   this.getThumb();
773 },
774 async deleteVideo(video) {
775   this.yesNoConfig.show = false;
776   this.lConfig = {
777     show: 'true',
778     message: 'Deleting your video..',
779   };
780   const { Uploads } = this.$FeathersVuex.api;
781   const result = await Uploads.get(video._id);
782   result
783     .remove()
784     .then(() => {
785       this.dConfig = {
786         show: true,
787         title: 'Sucess',
788         message: 'Video deleted successfully.',
789         buttonText: 'OK',
790         isPersistent: true,
791         redirect: 'refresh',
792         type: 'info',
793       };
794     })
795     .catch(() => {
796       this.dConfig = {
797         show: true,
798         title: 'Error',
799         message: 'Some error has occured while deleting your video.',
800         buttonText: 'Retry',
801         isPersistent: true,
802         redirect: 'refresh',
803         type: 'error',
804       };
805     });
806 },
807 beginConvert() {
808   if (this.$refs.convertForm.validate()) {

```



```

809     const token = window.localStorage['feathers-jwt'];
810     axios
811       .post(`${process.env.VUE_APP_SERVER_ADDR}/convert`, this.convert, {
812         headers: {
813           Authorization: `Bearer ${token}`,
814           'Content-Type': 'application/json',
815         },
816       })
817       .then((res) => {
818         axios
819           .get(
820             `${process.env.VUE_APP_SERVER_ADDR}/convert/${res.data._id}?download=true`,
821             {
822               headers: {
823                 Authorization: `Bearer ${token}`,
824               },
825               responseType: 'blob',
826             },
827           )
828           .then((response) => {
829             const url = window.URL.createObjectURL(new Blob([response.data]));
830             const link = document.createElement('a');
831             link.href = url;
832             link.setAttribute(
833               'download',
834               `${this.convert.name}.${this.convert.format}`,
835             );
836             document.body.appendChild(link);
837             link.click();
838             link.remove();
839           });
840         this.lConfig.show = false;
841         this.dConfig = {
842           show: true,
843           color: 'info',
844           title: 'Sucess',
845           icon: 'mdi-information',
846           message: 'Video Converted Successully!',
847           buttonText: 'OK',
848           isPersistent: true,
849         };
850       })
851       .catch(() => {
852         this.dConfig = {
853           show: true,
854           color: 'red',
855           title: 'An error occured',
856           icon: 'mdi-alert',
857           message: 'An error has occured while converting your video.',
858           buttonText: 'OK',
859           isPersistent: true,
860         };
861         this.lConfig.show = false;
862         this.convertVideo = false;
863       });

```

```

864         this.convertVideo = false;
865         this.lConfig.show = true;
866     }
867 },
868 validate() {
869     if (this.$refs.uploadForm.validate()) {
870         const token = window.localStorage['feathers-jwt'];
871         const formData = new FormData();
872         formData.append('files', this.video.files, this.video.files.name);
873         formData.append('description', this.video.description);
874         axios
875             .post(`${process.env.VUE_APP_SERVER_ADDR}/uploads`, formData, {
876                 headers: {
877                     Authorization: `Bearer ${token}`,
878                 },
879                 onUploadProgress: (e) => {
880                     if (e.lengthComputable) {
881                         this.uploadProgress = parseInt((e.loaded / e.total) * 100, 10);
882                     }
883                 },
884             })
885             .then(() => {
886                 this.dConfig = {
887                     show: true,
888                     color: 'info',
889                     title: 'Sucess',
890                     icon: 'mdi-information',
891                     message: 'Video Sucessfully Uploaded!',
892                     buttonText: 'Continue',
893                     isPersistent: true,
894                     redirect: 'refresh',
895                 };
896                 this.dialog = false;
897                 this.isUploading = false;
898                 this.video.files = null;
899                 this.video.description = null;
900             })
901             .catch(() => {
902                 this.dConfig = {
903                     show: true,
904                     color: 'red',
905                     title: 'An error occured',
906                     icon: 'mdi-alert',
907                     message: 'An error has occured while uploading your video.',
908                     buttonText: 'OK',
909                     isPersistent: true,
910                 };
911                 this.isUploading = false;
912             });
913         this.isUploading = true;
914     }
915 },
916 },
917 components: {
918     Dialog,

```

```

919     Loading,
920     YesNoDialog,
921   },
922 };
923 </script>

```

9.3 Server-Side Code

/package.json

```

1  {
2    "name": "video-converter-server",
3    "description": "",
4    "version": "0.0.0",
5    "homepage": "",
6    "main": "src",
7    "keywords": [
8      "feathers"
9    ],
10   "author": {
11     "name": "Bharat Singh Rajput",
12     "email": "bharatsr1997@gmail.com"
13   },
14   "contributors": [],
15   "bugs": {},
16   "directories": {
17     "lib": "src",
18     "test": "test/",
19     "config": "config/"
20   },
21   "engines": {
22     "node": "^12.0.0",
23     "npm": ">= 3.0.0"
24   },
25   "scripts": {
26     "test": "npm run eslint && npm run jest",
27     "eslint": "eslint src/. test/. --config .eslintrc.json",
28     "dev": "nodemon src/",
29     "start": "node src/",
30     "jest": "jest --forceExit"
31   },
32   "dependencies": {
33     "@feathersjs/authentication": "^4.5.8",
34     "@feathersjs/authentication-local": "^4.5.8",
35     "@feathersjs/configuration": "^4.5.8",
36     "@feathersjs/errors": "^4.5.8",
37     "@feathersjs/express": "^4.5.8",
38     "@feathersjs/feathers": "^4.5.8",
39     "@feathersjs/transport-commons": "^4.5.8",
40     "compression": "^1.7.4",
41     "cors": "^2.8.5",

```

```

42     "dotenv": "^8.2.0",
43     "feathers-authentication": "^1.3.1",
44     "feathers-authentication-hooks": "^1.0.2",
45     "feathers-authentication-management": "^3.0.0",
46     "feathers-blob": "^2.2.0",
47     "feathers-hooks": "^2.1.2",
48     "feathers-hooks-common": "^5.0.3",
49     "feathers-mailer": "^3.0.1",
50     "feathers-mongoose": "^8.3.1",
51     "ffmpeg": "0.0.4",
52     "file-saver": "^2.0.2",
53     "fluent-ffmpeg": "^2.1.2",
54     "helmet": "^3.23.3",
55     "jade": "^1.11.0",
56     "mime-db": "^1.44.0",
57     "mime-lookup": "0.0.2",
58     "mime-types": "^2.1.27",
59     "mongodb": "^3.6.0",
60     "mongodb-core": "^3.2.7",
61     "mongoose": "^5.10.0",
62     "multer": "^1.4.2",
63     "nodemailer-smtp-transport": "^2.7.4",
64     "serve-favicon": "^2.5.0",
65     "winston": "^3.3.3"
66   },
67   "devDependencies": {
68     "axios": "^0.19.2",
69     "eslint": "^7.7.0",
70     "eslint-config-airbnb": "^18.2.0",
71     "eslint-config-airbnb-base": "^14.2.0",
72     "eslint-plugin-import": "^2.22.0",
73     "eslint-plugin-jsx-a11y": "^6.3.1",
74     "eslint-plugin-react": "^7.20.6",
75     "jest": "^25.5.4",
76     "nodemon": "^2.0.4"
77   }
78 }

```

```
/src/app.js
```

```

1  const path = require('path');
2  const favicon = require('serve-favicon');
3  const compress = require('compression');
4  const helmet = require('helmet');
5  const cors = require('cors');
6  const feathers = require('@feathersjs/feathers');
7  const configuration = require('@feathersjs/configuration');
8  const express = require('@feathersjs/express');
9  const logger = require('./logger');
10 require('dotenv').config({ path: './env' });
11
12 const middleware = require('./middleware');
13 const services = require('./services');
14 const appHooks = require('./app.hooks');
15 const channels = require('./channels');

```

```

16
17 const authentication = require('./authentication');
18
19 const mongoose = require('./mongoose');
20
21 const mongodb = require('./mongodb');
22
23 const app = express(feathers());
24
25 // Load app configuration
26 app.configure(configuration());
27 // Enable security, CORS, compression, favicon and body parsing
28 app.use(helmet());
29 app.use(cors());
30 app.use(compress());
31 // Handle request upto 5MB
32 app.use(express.json({
33   limit: '5mb',
34 }));
35 app.use(express.urlencoded({ extended: true }));
36
37 app.use(favicon(path.join(app.get('public'), 'favicon.ico')));
38 // Host the public folder
39 app.use('/', express.static(app.get('public')));
40
41 // Set up Plugins and providers
42 app.configure(express.rest());
43
44 app.configure(mongoose);
45
46 app.configure(mongodb);
47
48 // Configure other middleware (see `middleware/index.js`)
49 app.configure(middleware);
50 app.configure(authentication);
51 // Set up our services (see `services/index.js`)
52 app.configure(services);
53 // Set up event channels (see channels.js)
54 app.configure(channels);
55
56 // Configure a middleware for 404s and the error handler
57 app.use(express.notFound());
58 app.use(express.errorHandler({ logger }));
59
60 app.hooks(appHooks);
61
62 module.exports = app;

```

/src/app.hooks.js

```

1 // Application hooks that run for every service
2
3 module.exports = {
4   before: {
5     all: [],

```

```

6     find: [],
7     get: [],
8     create: [],
9     update: [],
10    patch: [],
11    remove: [],
12  },
13
14  after: {
15    all: [],
16    find: [],
17    get: [],
18    create: [],
19    update: [],
20    patch: [],
21    remove: [],
22  },
23
24  error: {
25    all: [],
26    find: [],
27    get: [],
28    create: [],
29    update: [],
30    patch: [],
31    remove: [],
32  },
33 };

```

/src/authentication.js

```

1  const { AuthenticationService, JWTStrategy } = require('@feathersjs/authentication');
2  const { LocalStrategy } = require('@feathersjs/authentication-local');
3
4  module.exports = (app) => {
5    const authentication = new AuthenticationService(app);
6
7    authentication.register('jwt', new JWTStrategy());
8    authentication.register('local', new LocalStrategy());
9
10   app.use('/authentication', authentication);
11 };

```

/src/channels.js

```

1  // Changed Unnamed Function To Anonymous
2  module.exports = (app) => {
3    if (typeof app.channel !== 'function') {
4      // If no real-time functionality has been configured just return
5      return;
6    }
7
8    app.on('connection', (connection) => {
9      // On a new real-time connection, add it to the anonymous channel
10     app.channel('anonymous').join(connection);

```

```

11  });
12
13  app.on('login', (authResult, { connection }) => {
14    // connection can be undefined if there is no
15    // real-time connection, e.g. when logging in via REST
16    if (connection) {
17      // Obtain the logged in user from the connection
18      // const user = connection.user;
19
20      // The connection is no longer anonymous, remove it
21      app.channel('anonymous').leave(connection);
22
23      // Add it to the authenticated user channel
24      app.channel('authenticated').join(connection);
25
26      // Channels can be named anything and joined on any condition
27
28      // E.g. to send real-time events only to admins use
29      // if(user.isAdmin) { app.channel('admins').join(connection); }
30
31      // If the user has joined e.g. chat rooms
32      // if(Array.isArray(user.rooms)) {
33      //   user.rooms.forEach(room => app.channel(`rooms/${room.id}`).join(channel));
34      // }
35
36      // Easily organize users by email and userid for things like messaging
37      // app.channel(`emails/${user.email}`).join(channel);
38      // app.channel(`userIds/${user.id}`).join(channel);
39    }
40  });
41
42  // eslint-disable-next-line no-unused-vars
43  app.publish((data, hook) => {
44    // Here you can add event publishers to channels set up in `channels.js`
45    // To publish only for a specific event use `app.publish(eventname, () => {})`
46
47    // eslint-disable-next-line
48    console.log('Publishing all events to all authenticated users.');

```

/src/index.js

```
1  /* eslint-disable no-console */
2
3  const logger = require("./logger");
4  const app = require("./app");
5
6  const port = app.get("port");
7  const server = app.listen(port);
8
9  process.on("unhandledRejection", (reason, p) =>
10    logger.error("Unhandled Rejection at: Promise ", p, reason)
11  );
12
13  server.on("listening", () =>
14    logger.info(
15      "Feathers application started on http://%s:%d",
16      app.get("host"),
17      port
18    )
19  );
```

/src/logger.js

```
1  const { createLogger, format, transports } = require('winston');
2
3  // Configure the Winston logger.
4  // For the complete documentation see https://github.com/winstonjs/winston
5
6  const logger = createLogger({
7    // To see more detailed errors, change this to 'debug'
8    level: 'info',
9    format: format.combine(
10      format.splat(),
11      format.simple(),
12    ),
13    transports: [
14      new transports.Console(),
15      new transports.File({
16        filename: 'error.log',
17        level: 'error',
18      }),
19      new transports.File({
20        filename: 'combined.log',
21      }),
22    ],
23  });
24
25  module.exports = logger;
```

/src/mongodb.js

```
1  const { MongoClient } = require('mongodb');
2
```



```

3 // Changed Unnamed Funtion To Anonymous
4 module.exports = (app) => {
5   const connection = app.get('mongodb');
6   const database = connection.substr(connection.lastIndexOf('/') + 1);
7   const mongoClient = MongoClient.connect(connection, {
8     useNewUrlParser: true,
9     useUnifiedTopology: true,
10  })
11    .then((client) => client.db(database));
12
13   app.set('mongoClient', mongoClient);
14 };

```

/src/mongoose.js

```

1 const mongoose = require('mongoose');
2 const logger = require('./logger');
3
4 // Changed Unnamed Funtion To Anonymous
5 module.exports = (app) => {
6   mongoose.connect(
7     app.get('mongodb'),
8     {
9       useCreateIndex: true,
10      useNewUrlParser: true,
11      useUnifiedTopology: true,
12    },
13  ).catch((err) => {
14    logger.error(err);
15    process.exit(1);
16  });
17
18   mongoose.Promise = global.Promise;
19
20   app.set('mongooseClient', mongoose);
21 };

```

/src/hooks/attachThumb.js

```

1 const fs = require("fs");
2
3 module.exports = async (context) => {
4   function base64Encode(file) {
5     const bitmap = fs.readFileSync(file);
6     return Buffer.from(bitmap).toString("base64");
7   }
8   const result = await context.app.service("uploads").get(context.id);
9   // eslint-disable-next-line
10  if (context.params.user._id == result.userId) {
11    context.result.data = base64Encode(
12      `${result.newNameWithPath}_thumb\\tn_${context.params.query.thumbId}.png`
13    );
14  } else {
15    throw new Error(
16      "Requested thumb does not belong to a video uploaded by the current user."
17    );
18  }
19 };

```

```

17     );
18   }
19   return context;
20 };

```

/src/hooks/findFiles.js

```

1  // Return Only Those Videos That Belong To The Requesting User
2
3  module.exports = (context) => {
4    // eslint-disable-next-line
5    context.result.data = context.result.data.filter(
6      (video) => context.params.user._id == video.userId
7    );
8    return context;
9  };

```

/src/hooks/processConvert.js

```

1  // Convert The Video And Add Additional Attributes
2  const ffmpeg = require("fluent-ffmpeg");
3  const fs = require("fs");
4  const logger = require("../logger");
5
6  module.exports = async (context) => {
7    function getFileSizeInBytes(filename) {
8      const stats = fs.statSync(filename);
9      const fileSizeInBytes = stats.size;
10     return fileSizeInBytes;
11   }
12
13   // Converted File Name
14   // eslint-disable-next-line
15   const convertsPath = `public/uploads/${context.params.user._id}/converts`;
16   const newFilePath = `${convertsPath}/${Date.now()}${context.data.upload.originalName.replace(
17     / /g,
18     ""
19   )}_Converted.${context.data.format}`;
20   if (!fs.existsSync(convertsPath)) {
21     fs.mkdirSync(convertsPath);
22   }
23
24   // eslint-disable-next-line no-unused-vars
25   const result = await new Promise((resolve, reject) => {
26     const command = ffmpeg(context.data.upload.newNameWithPath)
27       .on("progress", (progress) => {
28         logger.info(`[ffmpeg] ${JSON.stringify(progress)}`);
29       })
30       .on("error", (err) => {
31         logger.error(`[ffmpeg] error: ${err.message}`);
32         reject(err);
33       })
34       .on("end", () => {
35         logger.info("[ffmpeg] finished");
36         resolve();

```

```

37     });
38     command.format(context.data.format);
39     command.fps(context.data.frameRate);
40     command.seek(context.data.range[0]);
41     command.duration(context.data.range[1] - context.data.range[0]);
42     if (context.data.type === "video") {
43         command.size(`${context.data.height}x${context.data.width}`);
44     }
45     if (context.data.isAudioDisabled) {
46         command.noAudio();
47     }
48     if (context.data.videoQuality < 100) {
49         const { videoBitrate } = context.data.upload;
50         const newBitrate = parseInt(
51             (videoBitrate * context.data.videoQuality) / 100,
52             10
53         );
54         command.videoBitrate(newBitrate);
55         context.data.videoBitrate = newBitrate;
56     } else {
57         context.data.videoBitrate = context.data.upload.videoBitrate;
58     }
59     if (context.data.audioQuality < 100) {
60         const { audioBitrate } = context.data.upload;
61         const newBitrate = parseInt(
62             (audioBitrate * context.data.audioQuality) / 100,
63             10
64         );
65         command.audioBitrate(newBitrate);
66         context.data.audioBitrate = newBitrate;
67     } else {
68         context.data.audioBitrate = context.data.upload.audioBitrate;
69     }
70     command.save(newFilePath);
71 });
72
73 // Add Attributes
74 context.data.userId = context.data.upload.userId;
75 // eslint-disable-next-line
76 context.data.originalVideoId = context.data.upload._id;
77 context.data.filePath = newFilePath;
78 context.data.fileSize = await getFileSizeInBytes(newFilePath);
79 context.data.name = `${context.data.name}.${context.data.format}`;
80 [context.data.durationStart, context.data.durationEnd] = context.data.range;
81
82 // Save Thumbnail
83 const ssResult = await new Promise((resolve, reject) => {
84     const imgDir = "public/uploads/images/";
85     if (context.data.type === "video") {
86         const resultThumb = ffmpeg(context.data.filePath)
87             .takeScreenshots(
88                 {
89                     count: 1,
90                     size: "336x188",
91                 },

```

```

92     imgDir,
93     (err) => {
94         logger.error(`[ffmpeg] error: ${err.message}`);
95         reject();
96     }
97 )
98 .on("end", () => {
99     const bitmap = fs.readFileSync(`${imgDir}/tn.png`);
100     context.data.thumbnail = Buffer.from(bitmap).toString("base64");
101     const delConfirm = fs.unlinkSync(`${imgDir}/tn.png`);
102     resolve();
103 });
104 } else {
105     const bitmap = fs.readFileSync(`${imgDir}/music.png`);
106     context.data.thumbnail = Buffer.from(bitmap).toString("base64");
107 }
108 });
109
110 return context;
111 };

```

/src/hooks/processFile.js

```

1  // Add Additional Video Info To Uploaded Video
2  const fs = require("fs");
3  const util = require("util");
4  const ffmpeg = require("fluent-ffmpeg");
5  // Promisify exec()
6  const exec = util.promisify(require("child_process").exec);
7  const logger = require("../logger");
8
9  module.exports = async (context) => {
10     function getFileSizeInBytes(filename) {
11         const stats = fs.statSync(filename);
12         const fileSizeInBytes = stats.size;
13         return fileSizeInBytes;
14     }
15
16     // Unfortunately exec() Output Is Piped To Standard Error
17     const { stderr } = await exec(`ffprobe ${context.data[0].newNameWithPath}`);
18     // Apply Regex ffprobe Output To Retrieve Additional Information
19     const durationRegex = /Duration: (.*?)\s/;
20     const videoBitrateRegex = /Stream.+Video.+, (.*?)kb\s/;
21     const audioBitrateRegex = /Stream.+Audio.+, (.*?)kb\s/;
22     const formatRegex = /Input.*?\.(.*)'\s/;
23     const resolutionRegex = /Stream #\d:\d.*, (\d+x\d+)/;
24     const fpsRegex = /Stream #\d:\d.*, (\d+\.\d+ fps)/;
25
26     // Add Attributes
27     const duration = durationRegex.exec(stderr);
28     const videoBitrate = videoBitrateRegex.exec(stderr);
29     const audioBitrate = audioBitrateRegex.exec(stderr);
30     const format = formatRegex.exec(stderr);
31     const resolution = resolutionRegex.exec(stderr);
32     const frameRate = fpsRegex.exec(stderr);

```

```

33
34 context.data[0].format = format ? format[1] : "Unidentified";
35 context.data[0].videoBitrate = videoBitrate
36   ? parseInt(videoBitrate[1], 10)
37   : 0;
38 context.data[0].audioBitrate = audioBitrate
39   ? parseInt(audioBitrate[1], 10)
40   : 0;
41
42 if (context.data[0].format !== "mp4") {
43   await new Promise((resolve, reject) => {
44     const command = ffmpeg(context.data[0].newNameWithPath)
45       .on("progress", (progress) => {
46         logger.info(`[ffmpeg] ${JSON.stringify(progress)}`);
47       })
48       .on("error", (err) => {
49         logger.error(`[ffmpeg] error: ${err.message}`);
50         reject(err);
51       })
52       .on("end", () => {
53         logger.info("[ffmpeg] finished");
54         resolve();
55       });
56     command.format("mp4");
57     const streamPath = `${context.data[0].newNameWithPath}_.mp4`;
58     command.save(streamPath);
59     context.data[0].streamPath = streamPath;
60   });
61   // eslint-disable-next-line
62   const { stderr } = await exec(`ffprobe ${context.data[0].streamPath}`);
63   const frameRateStream = fpsRegex.exec(stderr);
64   context.data[0].frameRate = frameRateStream
65     ? parseInt(frameRateStream[1], 10)
66     : 0;
67   if (context.data[0].videoBitrate === 0) {
68     // eslint-disable-next-line
69     const videoBitrate = videoBitrateRegex.exec(stderr);
70     context.data[0].videoBitrate = videoBitrate
71       ? parseInt(videoBitrate[1], 10)
72       : 0;
73   }
74   if (context.data[0].audioBitrate === 0) {
75     // eslint-disable-next-line
76     const audioBitrate = audioBitrateRegex.exec(stderr);
77     context.data[0].audioBitrate = audioBitrate
78       ? parseInt(audioBitrate[1], 10)
79       : 0;
80   }
81 } else {
82   context.data[0].streamPath = context.data[0].newNameWithPath;
83   context.data[0].frameRate = frameRate ? parseInt(frameRate[1], 10) : 0;
84 }
85
86 // eslint-disable-next-line
87 context.data[0].userId = context.params.user._id;

```

```

88   context.data[0].duration = duration ? duration[1] : 0;
89   context.data[0].resolution = resolution ? resolution[1] : "Unidentified";
90   context.data[0].fileSize = getFileSizeInBytes(
91     context.data[0].newNameWithPath
92   );
93   const ssDir = `${context.data[0].newNameWithPath}_thumb`;
94   fs.mkdirSync(ssDir);
95   await ffmpeg(context.data[0].newNameWithPath).takeScreenshots(
96     {
97       count: 30,
98       size: "336x188",
99     },
100    ssDir,
101    (err) => {
102      logger.error(`[ffmepeg] error: ${err.message}`);
103    }
104  );
105  // Save Thumbnail
106  const ssResult = await new Promise((resolve, reject) => {
107    const imgDir = "public/uploads/images/";
108    ffmpeg(context.data[0].newNameWithPath).takeScreenshots(
109      {
110        count: 1,
111        size: "336x188",
112      },
113      imgDir,
114      (err) => {
115        logger.error(`[ffmepeg] error: ${err.message}`);
116        reject();
117      }
118    ).on('end', () => {
119      const bitmap = fs.readFileSync(`${imgDir}/tn.png`);
120      context.data[0].thumbnail = Buffer.from(bitmap).toString("base64");
121      // eslint-disable-next-line
122      const removeResult = fs.unlinkSync(`${imgDir}/tn.png`);
123      resolve();
124    })
125  });
126
127
128   return context;
129 };

```

```
/src/hooks/sendVerificationEmail.js
```

```

1  const accountService = require("../services/authmanagement/notifier");
2
3  module.exports = (context) => {
4    if (!context.params.provider) {
5      return context;
6    }
7    const user = context.result;
8    if (process.env.GMAIL && context.data && context.data.email && user) {
9      accountService(context.app).notifier("resendVerifySignup", user);
10     return context;

```

```

11   }
12   return context;
13 };

```

```
/src/middleware/index.js
```

```

1  // Changed Unnamed Funtion To Anonymous
2  // eslint-disable-next-line no-unused-vars
3  module.exports = (app) => {
4    // Add your custom middleware here. Remember that in Express, the order matters
5  };

```

```
/src/models/convert.model.js
```

```

1  // convert-model.js - A mongoose model
2
3  // Changed Unnamed Funtion To Anonymous
4  module.exports = (app) => {
5    const modelName = "convert";
6    const mongooseClient = app.get("mongooseClient");
7    const { Schema } = mongooseClient;
8
9    const schema = new Schema(
10     {
11       name: { type: String, require: true },
12       format: { type: String, required: true },
13       height: { type: Number, required: true },
14       width: { type: Number, required: true },
15       filePath: { type: String, required: true },
16       originalVideoId: { type: String, required: true },
17       userId: { type: String, required: true },
18       type: { type: String, required: true },
19       isAudioDisabled: { type: String, required: true },
20       frameRate: { type: Number, required: true },
21       durationStart: { type: Number, required: true },
22       durationEnd: { type: Number, required: true },
23       fileSize: { type: Number, required: true },
24       thumbnail: { type: String, required: true },
25       audioBitrate: { type: Number, required: true },
26       videoBitrate: { type: Number, required: true },
27     },
28     {
29       timestamps: true,
30     }
31   );
32
33   if (mongooseClient.modelNames().includes(modelName)) {
34     mongooseClient.deleteModel(modelName);
35   }
36   return mongooseClient.model(modelName, schema);
37 };

```

```
/src/models/uploads.model.js
```

```

1 // uploads-model.js - A mongoose model
2
3 // Changed Unnamed Function To Anonymous
4 module.exports = (app) => {
5   const modelName = "uploads";
6   const mongooseClient = app.get("mongooseClient");
7   const { Schema } = mongooseClient;
8
9   const schema = new Schema(
10     {
11       description: { type: String, required: true },
12       originalName: { type: String, required: true },
13       newNameWithPath: { type: String, required: true },
14       streamPath: { type: String, required: true },
15       userId: { type: String, required: true },
16
17       format: { type: String, required: true },
18       audioBitrate: { type: Number, required: true },
19       videoBitrate: { type: Number, required: true },
20       resolution: { type: String, required: true },
21       duration: { type: String, required: true },
22       frameRate: { type: Number, required: true },
23       fileSize: { type: Number, required: true },
24       thumbnail: { type: String, require: true },
25     },
26     {
27       timestamps: true,
28     }
29   );
30
31   if (mongooseClient.modelNames().includes(modelName)) {
32     mongooseClient.deleteModel(modelName);
33   }
34   return mongooseClient.model(modelName, schema);
35 };

```

/src/models/users.model.js

```

1 // users-model.js - A mongoose model
2
3 // Changed Unnamed Function To Anonymous
4 module.exports = (app) => {
5   const modelName = "users";
6   const mongooseClient = app.get("mongooseClient");
7
8   const schema = new mongooseClient.Schema(
9     {
10       displayName: { type: String, required: true },
11       email: {
12         type: String,
13         unique: true,
14         lowercase: true,
15         required: true,
16       },
17       password: { type: String, required: true },

```



```

18     dob: { type: Date, required: true },
19     avatar: { type: String, required: true },
20
21     isVerified: { type: Boolean },
22     verifyToken: { type: String },
23     verifyShortToken: { type: String },
24     verifyExpires: { type: Date },
25     verifyChanges: { type: Object },
26     resetToken: { type: String },
27     resetShortToken: { type: String },
28     resetExpires: { type: Date },
29   },
30   {
31     timestamps: true,
32   }
33 );
34
35 if (mongooseClient.modelNames().includes(modelName)) {
36   mongooseClient.deleteModel(modelName);
37 }
38
39 return mongooseClient.model(modelName, schema);
40 };

```

/src/services/index.js

```

1  const users = require('./users/users.service.js');
2  const uploads = require('./uploads/uploads.service.js');
3  const convert = require('./convert/convert.service.js');
4  const message = require('./message/message.service.js');
5  const authmanagement = require('./authmanagement/authmanagement.service.js');
6  const streams = require('./streams/streams.service.js');
7  const thumbs = require('./thumbs/thumbs.service.js');
8
9  // Changed Unnamed Funtion To Anonymous
10 // eslint-disable-next-line no-unused-vars
11 module.exports = (app) => {
12   app.configure(users);
13   app.configure(uploads);
14   app.configure(convert);
15   app.configure(message);
16   app.configure(authmanagement);
17   app.configure(streams);
18   app.configure(thumbs);
19 };

```

/src/services/authManagement/authmanagement.hooks.js

```

1  // const hooks = require('feathers-hooks');
2  const auth = require("@feathersjs/authentication").hooks;
3  const { iff } = require("feathers-hooks-common");
4
5  const isAction = (...args) => (hook) => args.includes(hook.data.action);
6
7  module.exports = {

```

```

8   before: {
9     all: [],
10    find: [],
11    get: [],
12    create: [
13      iff(isAction("passwordChange, identityChange"), auth.authenticate("jwt")),
14    ],
15    update: [],
16    patch: [],
17    remove: [],
18  },
19
20  after: {
21    all: [],
22    find: [],
23    get: [],
24    create: [],
25    update: [],
26    patch: [],
27    remove: [],
28  },
29
30  error: {
31    all: [],
32    find: [],
33    get: [],
34    create: [],
35    update: [],
36    patch: [],
37    remove: [],
38  },
39 };

```

```
/src/services/authManagement/authmanagement.service.js
```

```

1  // Initializes the `authmanagement` service on path `/authmanagement`
2  const authManagement = require("feathers-authentication-management");
3  const hooks = require("../authmanagement.hooks");
4  const notifier = require("../notifier");
5
6  // Changed Unnamed Funtion To Anonymous
7  module.exports = (app) => {
8    // Initialize our service with any options it requires
9    app.configure(authManagement(notifier(app)));
10
11    // Get our initialized service so that we can register hooks
12    const service = app.service("authManagement");
13
14    service.hooks(hooks);
15  };

```

```
/src/services/authManagement/notifier.js
```

```

1  const logger = require("../../logger");
2

```

```

3  // Changed Unnamed Funtion To Anonymous
4  module.exports = (app) => {
5    function getLink(type, hash) {
6      return `${process.env.FRONTEND_ADDR}/${type}?token=${hash}`;
7    }
8
9    function sendMail(email) {
10     return app
11       .service("message")
12       .create(email)
13       .then((result) => {
14         logger.info(result);
15       })
16       .catch((err) => {
17         logger.error(err);
18       });
19   }
20
21   return {
22     notifier: (type, user) => {
23       let tokenLink;
24       let email;
25       switch (type) {
26         // Email Verification On SignUp
27         case "resendVerifySignup":
28           tokenLink = getLink("verify", user.verifyToken);
29           email = {
30             from: process.env.GMAIL,
31             to: user.email,
32             subject: "Please Verify Your Email!",
33             html: tokenLink,
34           };
35           return sendMail(email);
36
37         // Email Thanking For Verifcation
38         case "verifySignup":
39           email = {
40             from: process.env.GMAIL,
41             to: user.email,
42             subject: "Confirm Signup",
43             html:
44               "Thank You For Verifying Your Email! Have Fun Using Online Video Converter!",
45           };
46           return sendMail(email);
47
48         case "sendResetPwd":
49           tokenLink = getLink("reset", user.resetToken);
50           email = {
51             from: process.env.GMAIL,
52             to: user.email,
53             subject: "Reset Your Online Video Converter Password!",
54             html: tokenLink,
55           };
56           return sendMail(email);
57         default:

```

```

58         break;
59     }
60     return null;
61 },
62 };
63 };

```

```

/src/services/convert/convert.hooks.js

```

```

1  const { authenticate } = require("@feathersjs/authentication").hooks;
2  const { setField } = require("feathers-authentication-hooks");
3  const { disallow } = require("feathers-hooks-common");
4  const processConvert = require("../hooks/processConvert");
5
6  /* eslint-disable */
7  function searchRegex(context) {
8      const { query } = context.params;
9      for (const field in query) {
10         if (query[field].$search && field.indexOf("$") == -1) {
11             query[field] = { $regex: new RegExp(query[field].$search) };
12         }
13     }
14     context.params.query = query;
15     return context;
16 }
17 /* eslint-enable */
18
19 // Restrict Reading To Only Record Belonging To The Querying User
20 const readRestrict = [
21     setField({
22         from: "params.user._id",
23         as: "params.query.userId",
24     }),
25 ];
26
27 module.exports = {
28     before: {
29         all: [authenticate("jwt")],
30         find: [searchRegex, ...readRestrict],
31         get: [...readRestrict],
32         create: [processConvert],
33         update: [disallow("external")],
34         patch: [disallow("external")],
35         remove: [...readRestrict],
36     },
37
38     after: {
39         all: [],
40         find: [],
41         get: [],
42         create: [],
43         update: [],
44         patch: [],
45         remove: [],
46     },

```

```

47
48   error: {
49     all: [],
50     find: [],
51     get: [],
52     create: [],
53     update: [],
54     patch: [],
55     remove: [],
56   },
57 };

```

```

/src/services/convert/convert.service.js

```

```

1  // Initializes the `convert` service on path `/convert`
2
3  const fs = require("fs");
4  const MimeLookup = require("mime-lookup");
5  const mimeDb = require("mime-db");
6  const logger = require("../../logger");
7
8  const mime = new MimeLookup(mimeDb);
9  const { Convert } = require("./convert.class");
10 const createModel = require("../../models/convert.model");
11 const hooks = require("./convert.hooks");
12
13 // Changed Unnamed Funtion To Anonymous
14 module.exports = (app) => {
15   const options = {
16     Model: createModel(app),
17     paginate: app.get("paginate"),
18   };
19
20   // Initialize our service with any options it requires
21   app.use(
22     "/convert",
23     async (req, res, next) => {
24       const { method } = req;
25       if (method === "GET" && req.query.download) {
26         // Fetch The Converted Video Using Query URL
27         app
28           .service("convert")
29           .find({
30             query: {
31               // eslint-disable-next-line
32               _id: req.params.__feathersId,
33             },
34           })
35           .then((result) => {
36             // Send The Converted Video To The Client
37             // TODO: If Not Found Send Error
38             const newFilePath = result.data[0].filePath;
39             const file = fs.createReadStream(newFilePath);
40             const stat = fs.statSync(newFilePath);
41             res.status(200);

```

```

42         res.setHeader("Content-Length", stat.size);
43         res.setHeader("Content-Type", mime.lookup(newFilePath));
44         res.setHeader(
45             "Content-Disposition",
46             `attachment; filename=${newFilePath}`
47         );
48         file.pipe(res);
49     })
50     .catch(() => {
51         next();
52     });
53     } else if (method === "DELETE") {
54         // eslint-disable-next-line
55         await app
56             .service("convert")
57             .get(req.params.__feathersId)
58             .then((result) => {
59                 fs.unlinkSync(result.filePath);
60             })
61             .catch((err) => {
62                 logger.error(err.message);
63             });
64         next();
65     } else {
66         next();
67     }
68 },
69 new Convert(options, app)
70 );
71
72 // Get our initialized service so that we can register hooks
73 const service = app.service("convert");
74
75 service.hooks(hooks);
76 };

```

```
/src/services/convert/convert.class.js
```

```

1  const { Service } = require("feathers-mongoose");
2
3  exports.Convert = class Convert extends Service {};

```

```
/src/services/message/message.hooks.js
```

```

1  const common = require("feathers-hooks-common");
2
3  module.exports = {
4    before: {
5      all: [common.disallow("external")],
6      find: [],
7      get: [],
8      create: [],
9      update: [],
10     patch: [],
11     remove: [],

```

```

12 },
13
14 after: {
15   all: [],
16   find: [],
17   get: [],
18   create: [],
19   update: [],
20   patch: [],
21   remove: [],
22 },
23
24 error: {
25   all: [],
26   find: [],
27   get: [],
28   create: [],
29   update: [],
30   patch: [],
31   remove: [],
32 },
33 };

```

```
/src/services/message/message.service.js
```

```

1  // Initializes the `message` service on path `/message`
2  const Mailer = require("feathers-mailer");
3  const smtpTransport = require("nodemailer-smtp-transport");
4  const hooks = require("./message.hooks");
5
6  module.exports = (app) => {
7    // Initialize our service with any options it requires
8    app.use(
9      "/message",
10     new Mailer(
11       smtpTransport({
12         service: "Gmail",
13         auth: {
14           user: process.env.GMAIL,
15           pass: process.env.GMAIL_PASSWORD,
16         },
17       })
18     )
19   );
20
21   // Get our initialized service so that we can register hooks
22   const service = app.service("message");
23
24   service.hooks(hooks);
25 };

```

```
/src/services/streams/streams.hooks.js
```

```

1  const { disallow } = require('feathers-hooks-common');
2

```

```

3  const { authenticate } = require('@feathersjs/authentication').hooks;
4
5  module.exports = {
6    before: {
7      all: [authenticate('jwt')],
8      find: [],
9      get: [],
10     create: [],
11     update: [],
12     patch: [],
13     remove: [],
14   },
15
16   after: {
17     all: [],
18     find: [],
19     get: [],
20     create: [],
21     update: [],
22     patch: [],
23     remove: [],
24   },
25
26   error: {
27     all: [],
28     find: [],
29     get: [],
30     create: [],
31     update: [],
32     patch: [],
33     remove: [],
34   },
35 };

```

```
/src/services/streams/streams.service.js
```

```

1  // Initializes the `streams` service on path `/streams`
2  const fs = require('fs');
3  const mime = require('mime-types');
4  const hooks = require('./streams.hooks');
5  const logger = require('../../logger');
6
7  module.exports = (app) => {
8    // Initialize our service with any options it requires
9    // eslint-disable-next-line
10   app.use('/streams', async (req, res, next) => {
11     const { method } = req;
12     if (method === 'GET') {
13       // eslint-disable-next-line
14       await app.service('uploads').get(req.url.substr(1, req.url.length - 1))
15         .then((result) => {
16           let path;
17           if (result.format !== 'mp4') {
18             path = result.streamPath;
19           } else {

```



```

20     path = result.newNameWithPath;
21 }
22 const stat = fs.statSync(path);
23 const fileSize = stat.size;
24 const { range } = req.headers;
25 if (range) {
26     const parts = range.replace(/bytes=/, '').split('-');
27     const start = parseInt(parts[0], 10);
28     const end = parts[1] ? parseInt(parts[1], 10) : fileSize - 1;
29     const chunkSize = (end - start) + 1;
30     const file = fs.createReadStream(path, { start, end });
31     const head = {
32         'Content-Range': `bytes ${start}-${end}/${fileSize}`,
33         'Accept-Ranges': 'bytes',
34         'Content-Length': chunkSize,
35         'Content-Type': mime.lookup(result.newNameWithPath),
36     };
37     res.writeHead(206, head);
38     file.pipe(res);
39 } else {
40     const head = {
41         'Content-Length': fileSize,
42         'Content-Type': mime.lookup(result.newNameWithPath),
43     };
44     res.writeHead(200, head);
45     fs.createReadStream(path).pipe(res);
46 }
47 })
48 .catch((err) => {
49     logger.error(err);
50 });
51 }
52 });
53
54 // Get our initialized service so that we can register hooks
55 // const service = app.service('streams');
56
57 // service.hooks(hooks);
58 };

```

```
/src/services/thumbs/thumbs.hooks.js
```

```

1  const { authenticate } = require('@feathersjs/authentication').hooks;
2  const attachThumb = require('../hooks/attachThumb');
3
4  module.exports = {
5    before: {
6      all: [authenticate('jwt')],
7      find: [],
8      get: [],
9      create: [],
10     update: [],
11     patch: [],
12     remove: [],
13   },

```

```

14
15   after: {
16     all: [],
17     find: [],
18     get: [attachThumb],
19     create: [],
20     update: [],
21     patch: [],
22     remove: [],
23   },
24
25   error: {
26     all: [],
27     find: [],
28     get: [],
29     create: [],
30     update: [],
31     patch: [],
32     remove: [],
33   },
34 };

```

```
/src/services/thumbs/thumbs.service.js
```

```

1  // Initializes the `thumbs` service on path `/thumbs`
2  const { Thumbs } = require('./thumbs.class');
3  const hooks = require('./thumbs.hooks');
4
5  // Changed Unnamed Funtion To Anonymous
6  module.exports = (app) => {
7    const options = {
8      paginate: app.get('paginate'),
9    };
10
11    // Initialize our service with any options it requires
12    app.use('thumbs', new Thumbs(options, app));
13
14    // Get our initialized service so that we can register hooks
15    const service = app.service('thumbs');
16
17    service.hooks(hooks);
18  };

```

```
/src/services/thumbs/thumbs.class.js
```

```

1  /* eslint-disable class-methods-use-this, no-unused-vars */
2
3  exports.Thumbs = class Thumbs {
4    constructor(options) {
5      this.options = options || {};
6    }
7
8    async find(params) {
9      return [];
10    }

```

```

11
12   async get(id, params) {
13     return {
14       id, text: `A new message with ID: ${id}!`,
15     };
16   }
17
18   async create(data, params) {
19     if (Array.isArray(data)) {
20       return Promise.all(data.map((current) => this.create(current, params)));
21     }
22
23     return data;
24   }
25
26   async update(id, data, params) {
27     return data;
28   }
29
30   async patch(id, data, params) {
31     return data;
32   }
33
34   async remove(id, params) {
35     return { id };
36   }
37 };

```

```
/src/services/uploads/uploads.hooks.js
```

```

1  const { authenticate } = require("@feathersjs/authentication").hooks;
2  const { setField } = require("feathers-authentication-hooks");
3  const { disallow } = require("feathers-hooks-common");
4
5  const processFile = require("../hooks/processFile");
6
7  // Restrict Reading To Only Record Belonging To The Querying User
8  const readRestrict = [
9    setField({
10      from: "params.user._id",
11      as: "params.query.userId",
12    }),
13  ];
14
15  /* eslint-disable */
16  function searchRegex(context) {
17    const { query } = context.params;
18    for (const field in query) {
19      if (query[field].$search && field.indexOf("$") == -1) {
20        query[field] = { $regex: new RegExp(query[field].$search) };
21      }
22    }
23    context.params.query = query;
24    return context;
25  }

```

```

26  /* eslint-enable */
27
28  module.exports = {
29    before: {
30      all: [authenticate("jwt")],
31      find: [...readRestrict, searchRegex],
32      get: [...readRestrict],
33      create: [processFile],
34      update: [disallow("external")],
35      patch: [disallow("external")],
36      remove: [...readRestrict],
37    },
38
39    after: {
40      all: [],
41      find: [],
42      get: [],
43      create: [],
44      update: [],
45      patch: [],
46      remove: [],
47    },
48
49    error: {
50      all: [],
51      find: [],
52      get: [],
53      create: [],
54      update: [],
55      patch: [],
56      remove: [],
57    },
58  };

```

```
/src/services/uploads/uploads.service.js
```

```

1  const multer = require("multer");
2  const fs = require("fs");
3  const { authenticate } = require("@feathersjs/express");
4  const logger = require("../..//logger");
5  const { Uploads } = require("../uploads.class");
6  const createModel = require("../..//models/uploads.model");
7  const hooks = require("../uploads.hooks");
8
9  const storage = multer.diskStorage({
10    // eslint-disable-next-line
11    destination: (req, file, cb) => {
12      // eslint-disable-next-line
13      const path = `public/uploads/${req.user._id}`;
14      if (!fs.existsSync(path)) {
15        fs.mkdirSync(path);
16      }
17      return cb(null, path);
18    },
19    filename: (req, file, cb) =>

```

```

20     cb(null, `${Date.now()}-${file.originalname.replace(/ /g, "")}`),
21   });
22
23   const upload = multer({
24     storage,
25     limits: {
26       fileSize: 5e8,
27       fileSize: 5e8,
28     },
29   });
30
31   // Changed Unnamed Funtion To Anonymous
32   module.exports = (app) => {
33     const options = {
34       Model: createModel(app),
35       paginate: app.get("paginate"),
36       multi: true,
37     };
38
39     app.use(
40       "/uploads",
41       authenticate("jwt"),
42       upload.array("files"),
43       (req, res, next) => {
44         const { method } = req;
45         if (method === "POST" || method === "PATCH") {
46           req.feathers.files = req.files;
47           const body = [];
48           const files = Object.values(req.files);
49           files.forEach((file) => {
50             body.push({
51               description: req.body.description,
52               originalName: req.body.originalname,
53               newNameWithPath: file.path,
54             });
55             req.body = method === "POST" ? body : body[0];
56           });
57           next();
58         } else if (method === "DELETE") {
59           app
60             .service("uploads")
61             // eslint-disable-next-line
62             .get(req.params.__feathersId)
63             .then((result) => {
64               if (result.format !== "mp4") {
65                 fs.unlinkSync(result.streamPath);
66               }
67               fs.unlinkSync(result.newNameWithPath);
68               fs.rmdirSync(`${result.newNameWithPath}_thumb`, {
69                 recursive: true,
70               });
71             });
72           next();
73         }
74         .catch((err) => {
75           logger.error(err.message);

```

```

75         });
76     } else {
77         next();
78     }
79 },
80 (req, res, next) => {
81     const { method } = req;
82     if (method === "POST" || method === "PATCH") {
83         req.body[0].originalName = req.files[0].originalname;
84         req.body[0].newNameWithPath = req.files[0].path;
85     }
86     next();
87 },
88 new Uploads(options, app)
89 );
90 const service = app.service("uploads");
91 service.hooks(hooks);
92 };

```

```
/src/services/uploads/uploads.class.js
```

```

1  const { Service } = require("feathers-mongoose");
2
3  exports.Uploads = class Uploads extends Service {};

```

```
/src/services/users/users.hooks.js
```

```

1  const { authenticate } = require('@feathersjs/authentication').hooks;
2  const { hashPassword, protect } = require('@feathersjs/authentication-local').hooks;
3  const verifyHooks = require('feathers-authentication-management').hooks;
4
5  const {
6    iff, isProvider, preventChanges, disallow,
7  } = require('feathers-hooks-common');
8
9  const { setField } = require('feathers-authentication-hooks');
10 const { Forbidden } = require('@feathersjs/errors');
11 const sendVerificationEmail = require('../hooks/sendVerificationEmail');
12
13 // Restrict Reading To Only Record Belonging To The Querying User
14 const readRestrict = [
15   setField({
16     from: 'params.user._id',
17     as: 'params.query._id',
18   }),
19 ];
20
21 module.exports = {
22   before: {
23     all: [],
24     find: [
25       authenticate('jwt'),
26       ...readRestrict,
27     ],
28     get: [

```

```

29     authenticate('jwt'),
30     ...readRestrict,
31 ],
32 create: [
33     hashPassword('password'),
34     verifyHooks.addVerification(),
35 ],
36 update: [
37     authenticate('jwt'),
38     disallow('external'),
39 ],
40 patch: [
41     authenticate('jwt'),
42     ...readRestrict,
43     iff(isProvider('external'), preventChanges(
44         true,
45         'isVerified',
46         'verifyToken',
47         'verifyShortToken',
48         'verifyExpires',
49         'verifyChanges',
50         'resetToken',
51         'resetShortToken',
52         'resetExpires',
53     )),
54 ],
55 remove: [
56     authenticate('jwt'),
57     disallow('external'),
58 ],
59 },
60
61 after: {
62     all: [
63         // Make sure the password field is never sent to the client
64         // Always must be the last hook
65         protect('password'),
66     ],
67     find: [
68         disallow('external'),
69     ],
70     get: [
71         verifyHooks.removeVerification(),
72         iff(isProvider('external'), (context) => {
73             if (!context.result.isVerified) {
74                 throw new Forbidden('User not verified.');
75             }
76             return context;
77         }),
78     ],
79     create: [
80         sendVerificationEmail,
81         verifyHooks.removeVerification(),
82     ],
83     update: [],

```

```

84     patch: [
85         verifyHooks.removeVerification(),
86     ],
87     remove: [],
88 },
89
90 error: {
91     all: [],
92     find: [],
93     get: [],
94     create: [],
95     update: [],
96     patch: [],
97     remove: [],
98 },
99 };

```

```
/src/services/users/users.service.js
```

```

1  // Initializes the `users` service on path `/users`
2  const { Users } = require('./users.class');
3  const createModel = require('../../models/users.model');
4  const hooks = require('./users.hooks');
5
6  // Changed Unnamed Funtion To Anonymous
7  module.exports = (app) => {
8      const options = {
9          Model: createModel(app),
10         paginate: app.get('paginate'),
11     };
12
13     // Initialize our service with any options it requires
14     app.use('/users/', new Users(options, app));
15
16     // Get our initialized service so that we can register hooks
17     const service = app.service('/users/');
18
19     service.hooks(hooks);
20 };

```

```
/src/services/users/users.class.js
```

```

1  const { Service } = require('feathers-mongoose');
2
3  exports.Users = class Users extends Service {};

```

```
/test/app.test.js
```

```

1  /* eslint-disable no-undef */
2
3  const axios = require('axios');
4  const url = require('url');
5  const app = require('../src/app');
6

```



```

7  const port = app.get('port') || 8998;
8  const getUrl = (pathname) => url.format({
9    hostname: app.get('host') || 'localhost',
10   protocol: 'http',
11   port,
12   pathname,
13 });
14
15 describe('Feathers application tests (with jest)', () => {
16   let server;
17
18   beforeAll((done) => {
19     server = app.listen(port);
20     server.once('listening', () => done());
21   });
22
23   afterAll((done) => {
24     server.close(done);
25   });
26
27   it('starts and shows the index page', async () => {
28     expect.assertions(1);
29
30     const { data } = await axios.get(getUrl());
31
32     expect(data.indexOf('<html lang="en">')).not.toBe(-1);
33   });
34
35   describe('404', () => {
36     it('shows a 404 HTML page', async () => {
37       expect.assertions(2);
38       try {
39         await axios.get(getUrl('path/to/nowhere'), {
40           headers: {
41             Accept: 'text/html',
42           },
43         });
44       } catch (error) {
45         const { response } = error;
46
47         expect(response.status).toBe(404);
48         expect(response.data.indexOf('<html>')).not.toBe(-1);
49       }
50     });
51
52     it('shows a 404 JSON error without stack trace', async () => {
53       expect.assertions(4);
54
55       try {
56         await axios.get(getUrl('path/to/nowhere'));
57       } catch (error) {
58         const { response } = error;
59
60         expect(response.status).toBe(404);
61         expect(response.data.code).toBe(404);

```

```

62         expect(response.data.message).toBe('Page not found');
63         expect(response.data.name).toBe('NotFound');
64     }
65 });
66 });
67 });

```

/test/authentication.test.js

```

1  /* eslint-disable no-undef */
2
3  const app = require('../src/app');
4
5  describe('authentication', () => {
6      it('registered the authentication service', () => {
7          expect(app.service('authentication')).toBeTruthy();
8      });
9
10     describe('local strategy', () => {
11         const userInfo = {
12             email: 'someone@example.com',
13             password: 'supersecret',
14         };
15
16         beforeAll(async () => {
17             try {
18                 await app.service('users').create(userInfo);
19             } catch (error) {
20                 // Do nothing, it just means the user already exists and can be tested
21             }
22         });
23
24         it('authenticates user and creates accessToken', async () => {
25             const { user, accessToken } = await app.service('authentication').create({
26                 strategy: 'local',
27                 ...userInfo,
28             });
29
30             expect(accessToken).toBeTruthy();
31             expect(user).toBeTruthy();
32         });
33     });
34 });

```

/test/services/authmanagement.test.js

```

1  /* eslint-disable no-undef */
2
3  const app = require('../../src/app');
4
5  describe('\authmanagement\ service', () => {
6      it('registered the service', () => {
7          const service = app.service('authmanagement');
8          expect(service).toBeTruthy();
9      });
10 });

```

/test/services/convert.test.js

```
1  /* eslint-disable no-undef */
2
3  const app = require('../../src/app');
4
5  describe('\convert\ service', () => {
6    it('registered the service', () => {
7      const service = app.service('convert');
8      expect(service).toBeTruthy();
9    });
10 });
```

/test/services/message.test.js

```
1  /* eslint-disable no-undef */
2
3  const app = require('../../src/app');
4
5  describe('\message\ service', () => {
6    it('registered the service', () => {
7      const service = app.service('message');
8      expect(service).toBeTruthy();
9    });
10 });
```

/test/services/streams.test.js

```
1  /* eslint-disable no-undef */
2
3  const app = require('../../src/app');
4
5  describe('\streams\ service', () => {
6    it('registered the service', () => {
7      const service = app.service('streams');
8      expect(service).toBeTruthy();
9    });
10 });
```

/test/services/thumbs.test.js

```
1  /* eslint-disable no-undef */
2
3  const app = require('../../src/app');
4
5  describe('\thumbs\ service', () => {
6    it('registered the service', () => {
7      const service = app.service('thumbs');
8      expect(service).toBeTruthy();
9    });
10 });
```

/test/services/uploads.test.js

```

1  /* eslint-disable no-undef */
2
3  const app = require('../../src/app');
4
5  describe('\`uploads\` service', () => {
6    it('registered the service', () => {
7      const service = app.service('uploads');
8      expect(service).toBeTruthy();
9    });
10 });

```

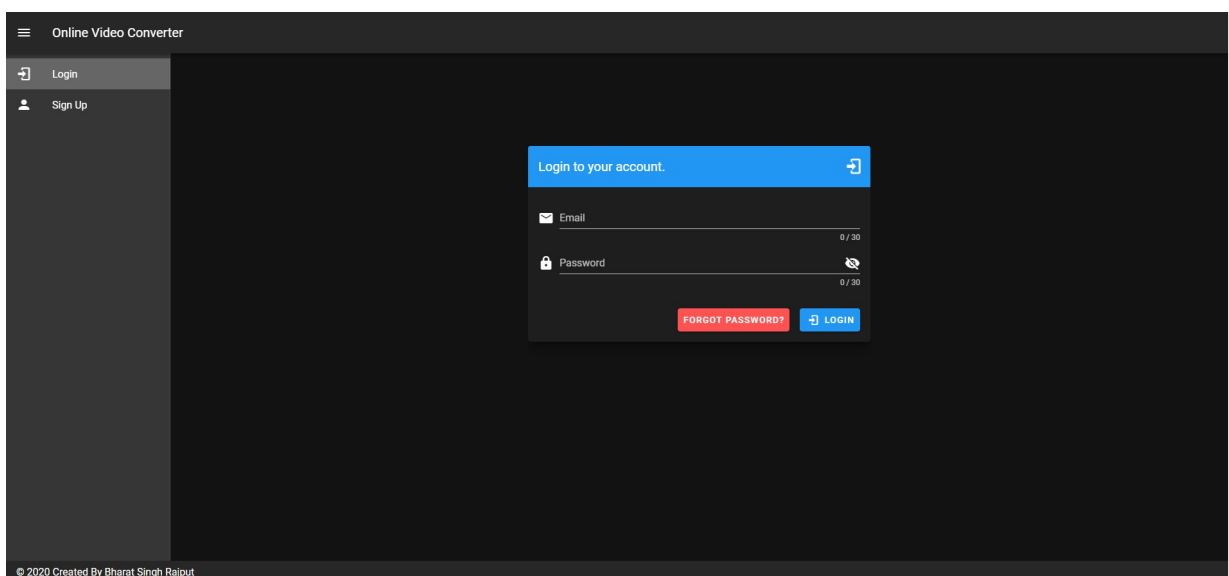
/test/services/users.test.js

```

1  /* eslint-disable no-undef */
2
3  const app = require('../../src/app');
4
5  describe('\`users\` service', () => {
6    it('registered the service', () => {
7      const service = app.service('users');
8      expect(service).toBeTruthy();
9    });
10 });

```

10 Screenshots



Login Page

Online Video Converter

Login

Sign Up

Create a new account.

Name 0 / 30

Email 0 / 30

Password 0 / 30

Confirm Password 0 / 30

Date Of Birth 0 / 30

2020-08-29

Avatar

Pick an avatar

SIGN UP

© 2020 Created By Bharat Singh Rajput

Sign Up Page

Online Video Converter

Login

Sign Up

Recover Your Password.

Email 0 / 30

SEND RESET MAIL

© 2020 Created By Bharat Singh Rajput

Forgot Password Page

Online Video Converter

Login

Sign Up

Create a new account.

Name 19 / 30

Bharat Singh Rajput

Email 15 / 30

An error occurred

An account with the same e-mail already exists.

OK

Confirm Password 10 / 30

1998-08-29

Date Of Birth 10 / 30

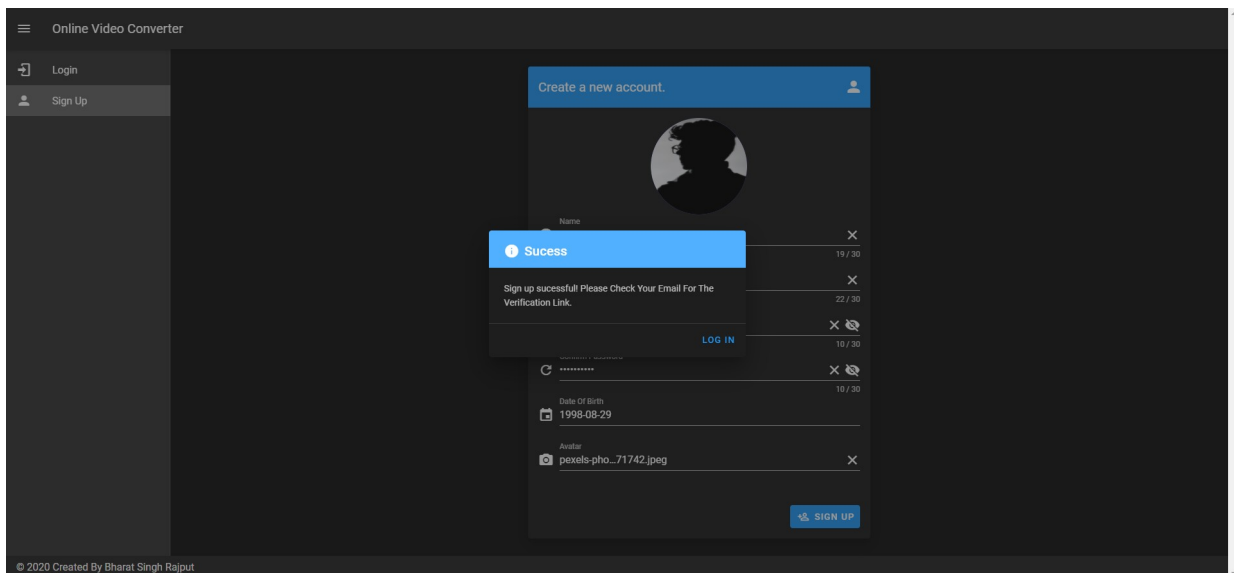
Avatar

pexels-pho_71742.jpeg

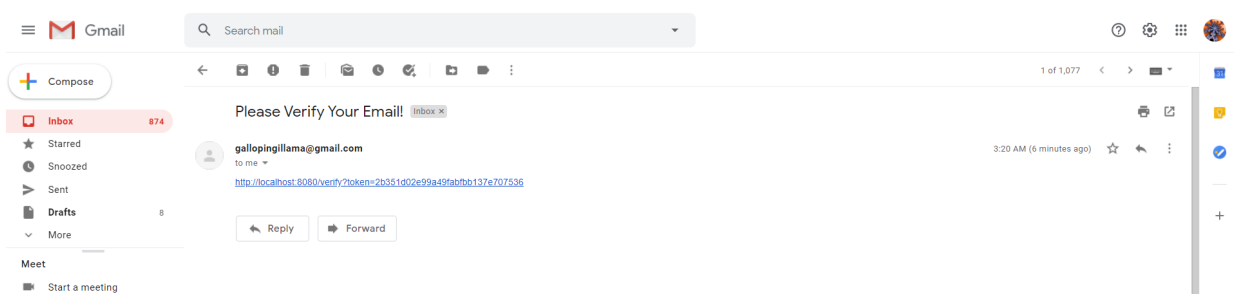
SIGN UP

© 2020 Created By Bharat Singh Rajput

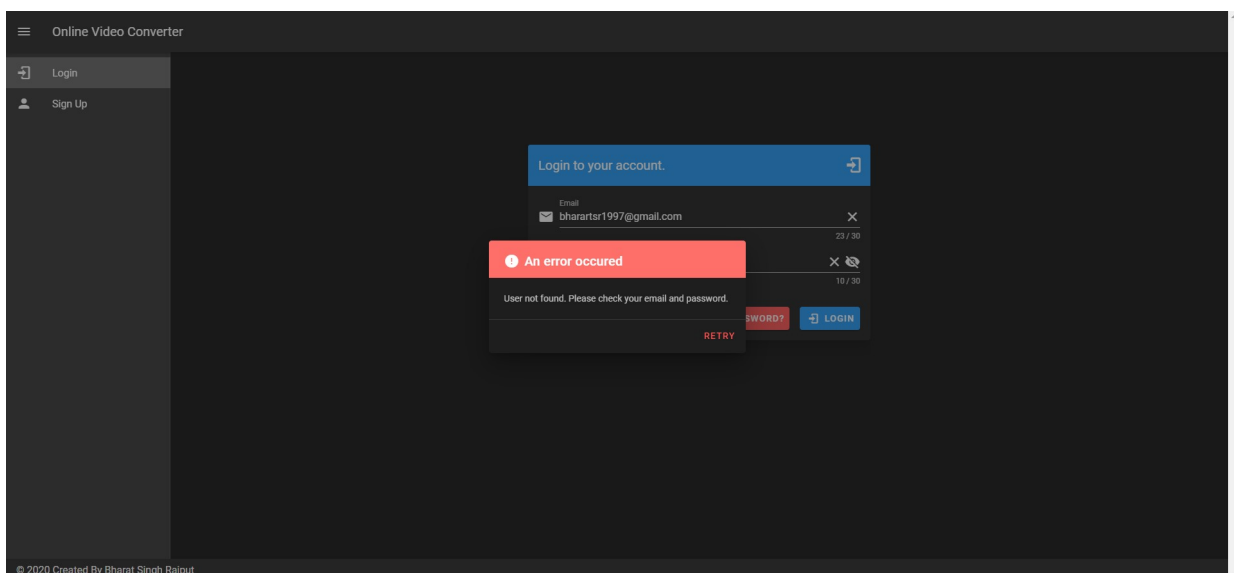
Sign Up Validation



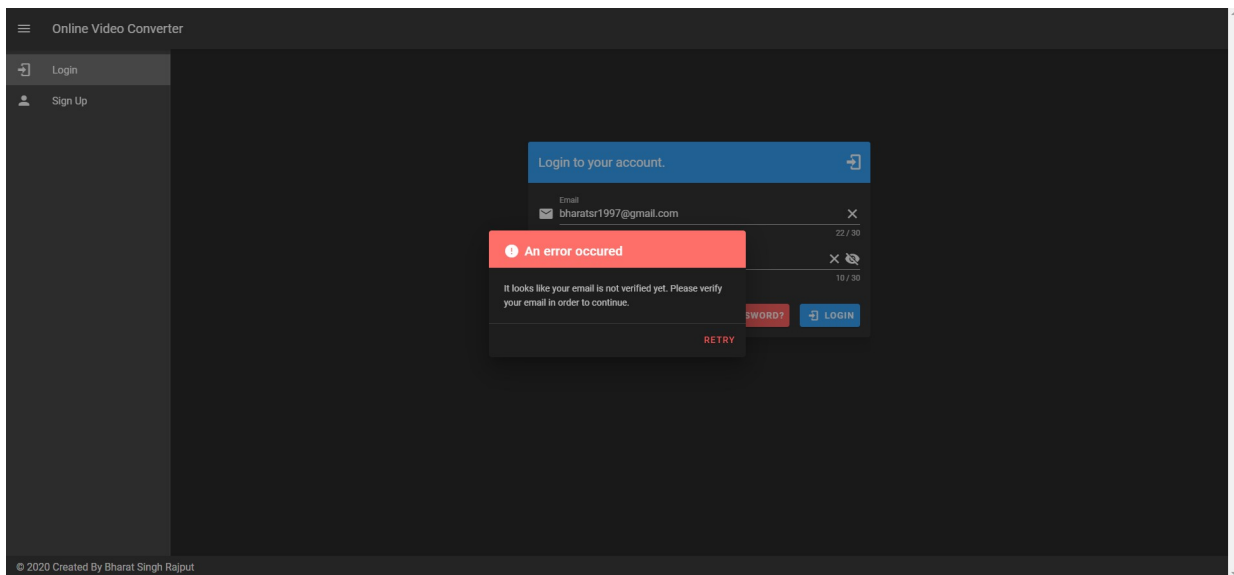
Sign Up Confirmation



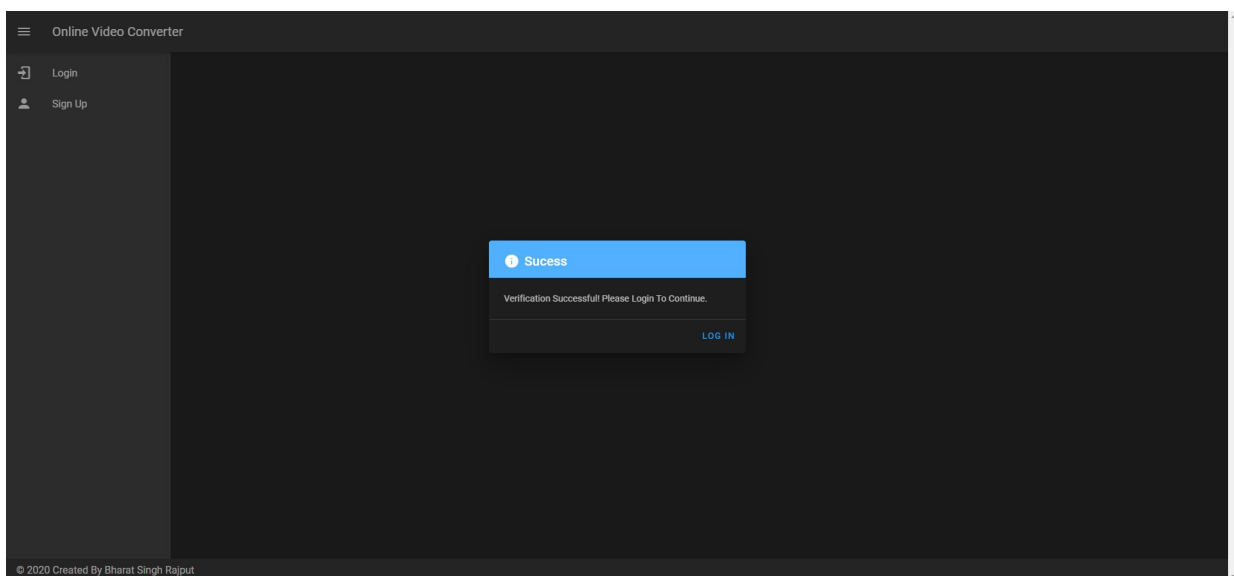
Sign Up Confirmation



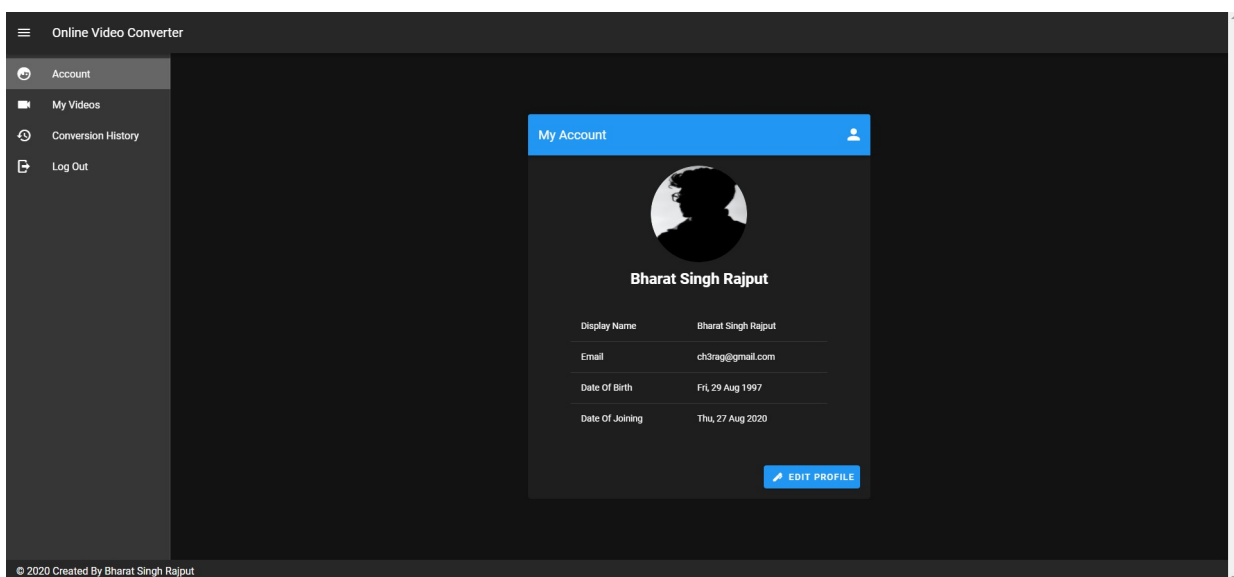
Log In Validation



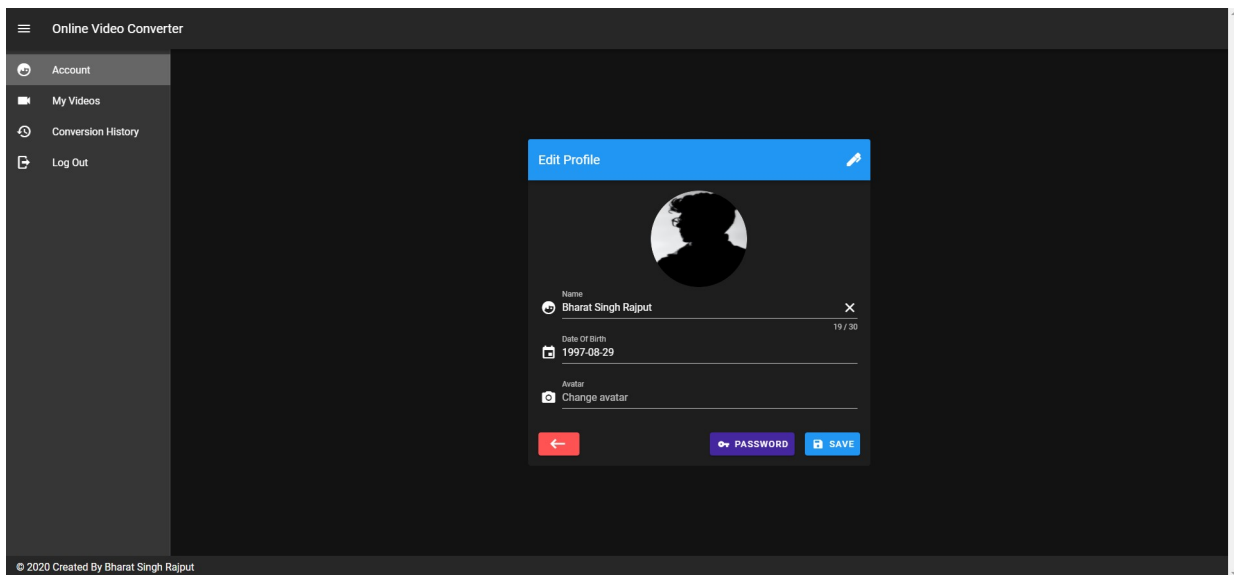
Log In Without Verification



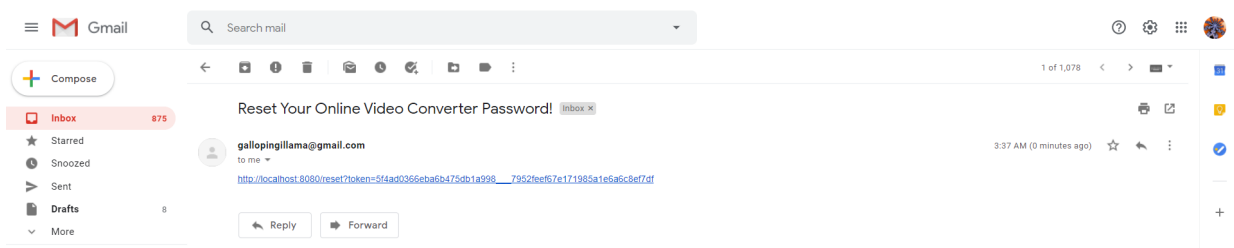
User Verification



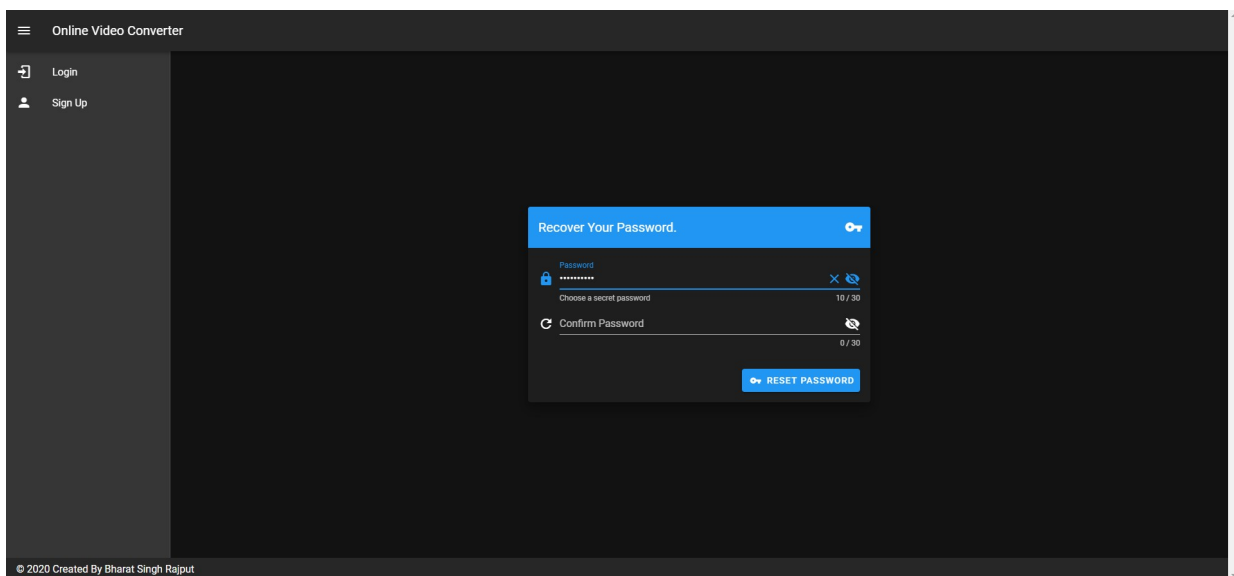
User Account



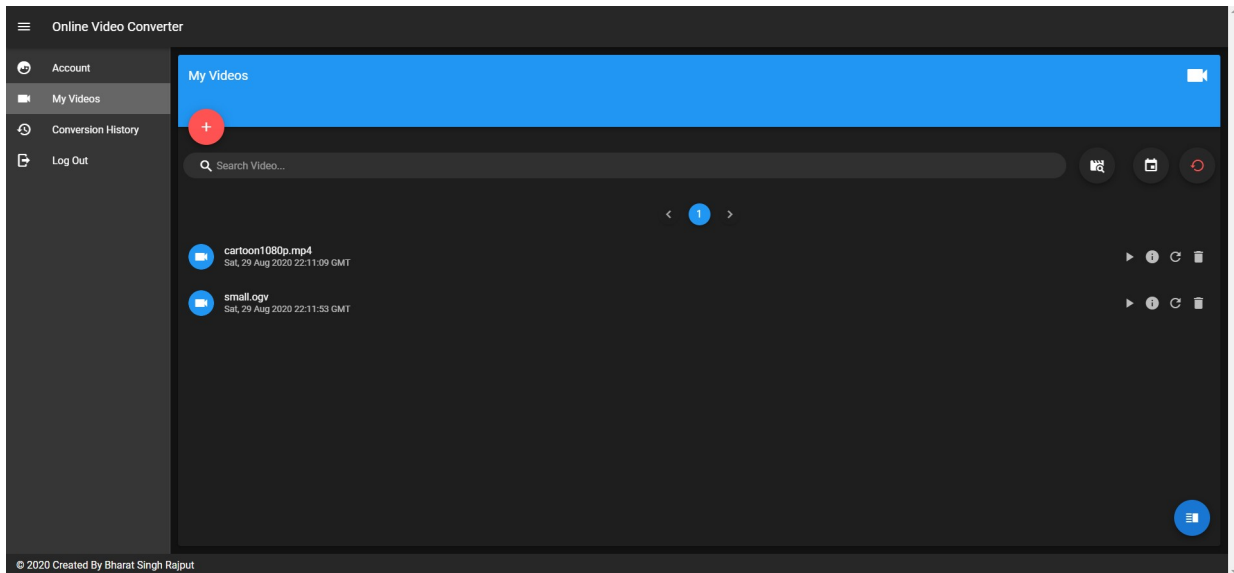
Edit User Account



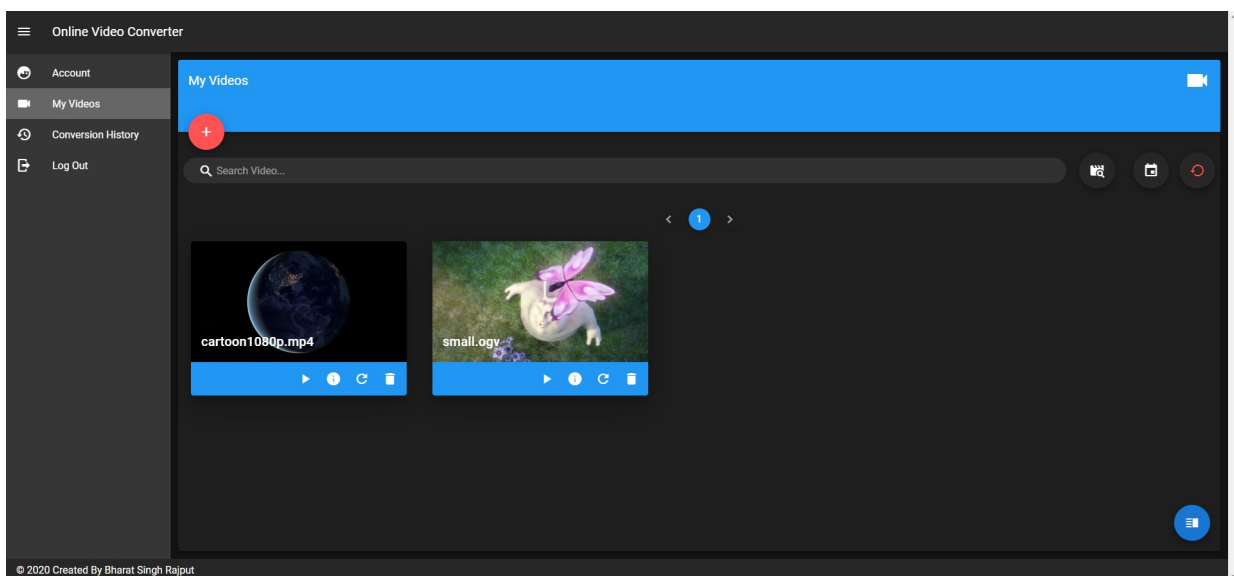
Reset Password Email



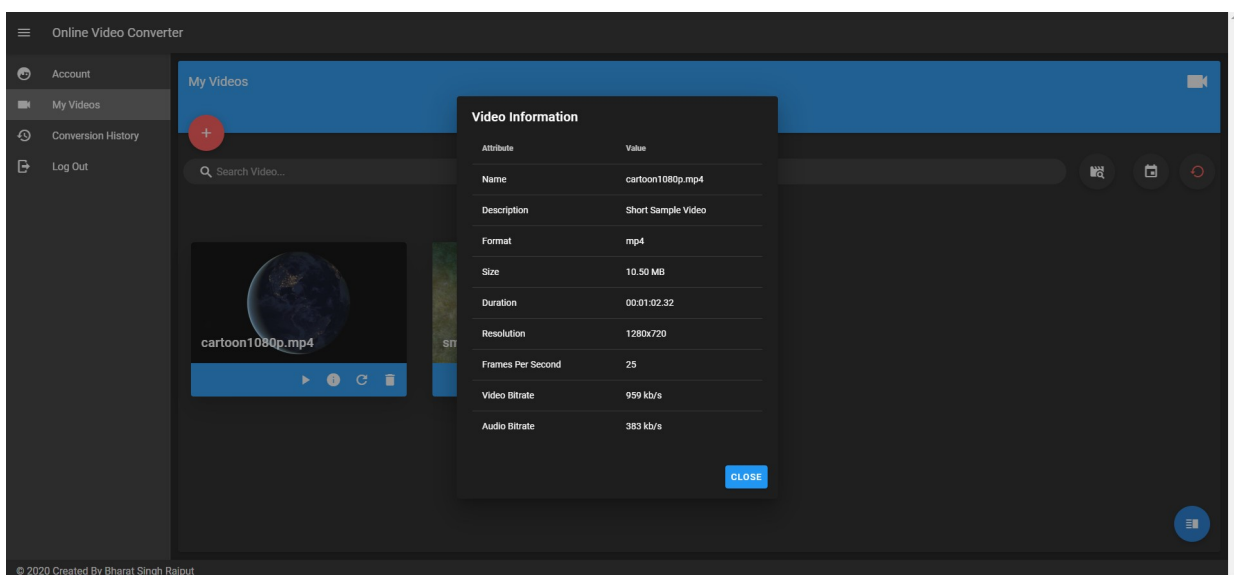
Reset Password Landing Page



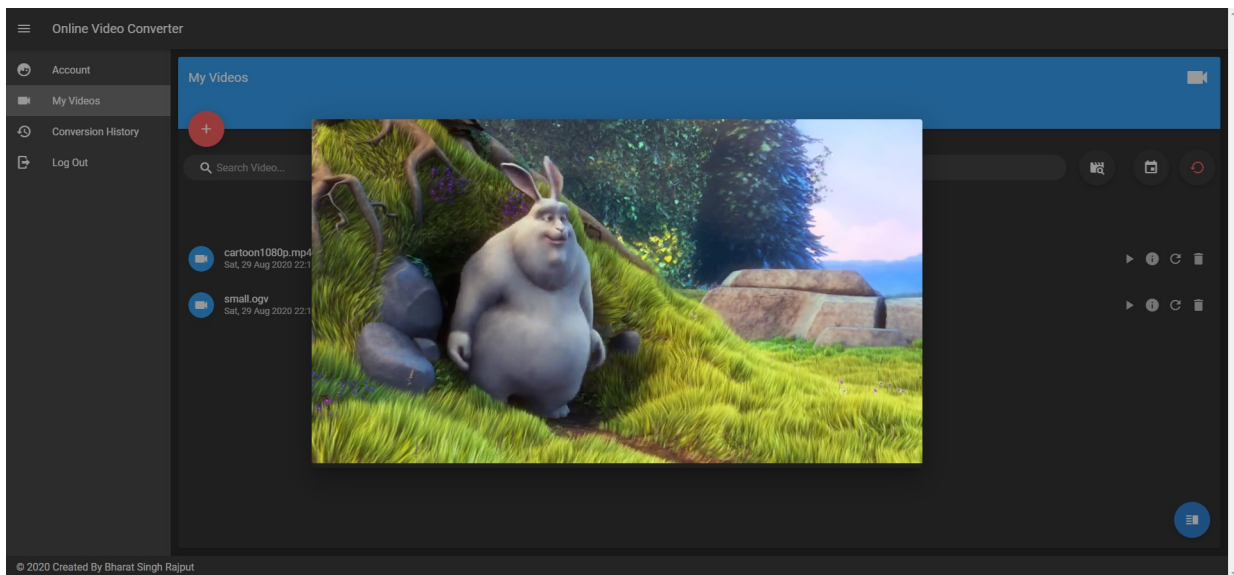
User Videos List View



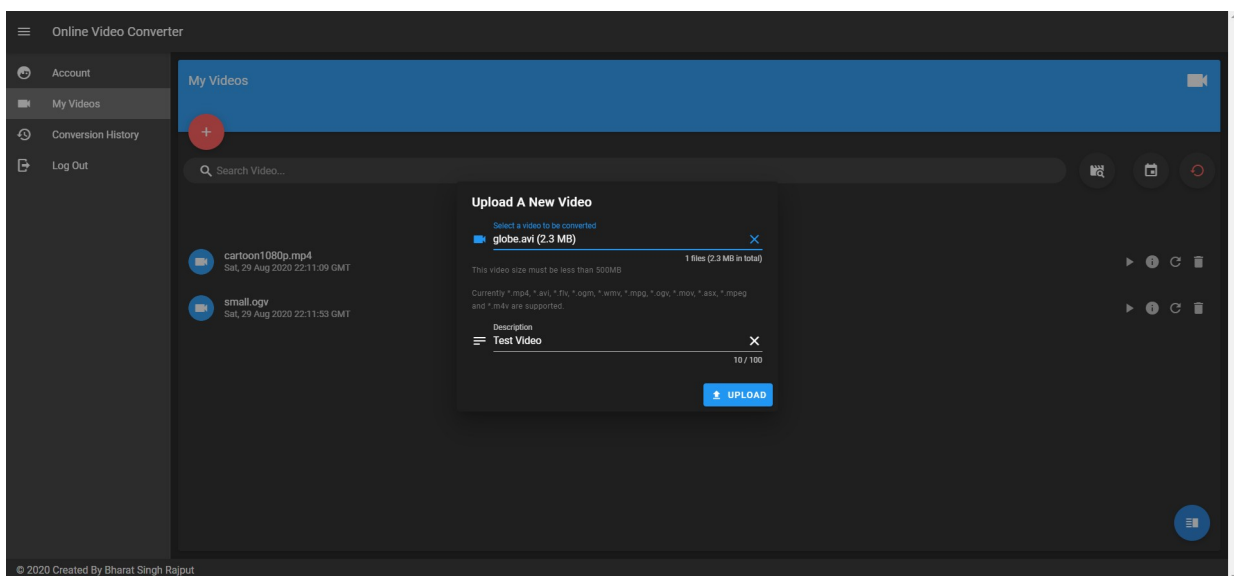
User Videos Card View



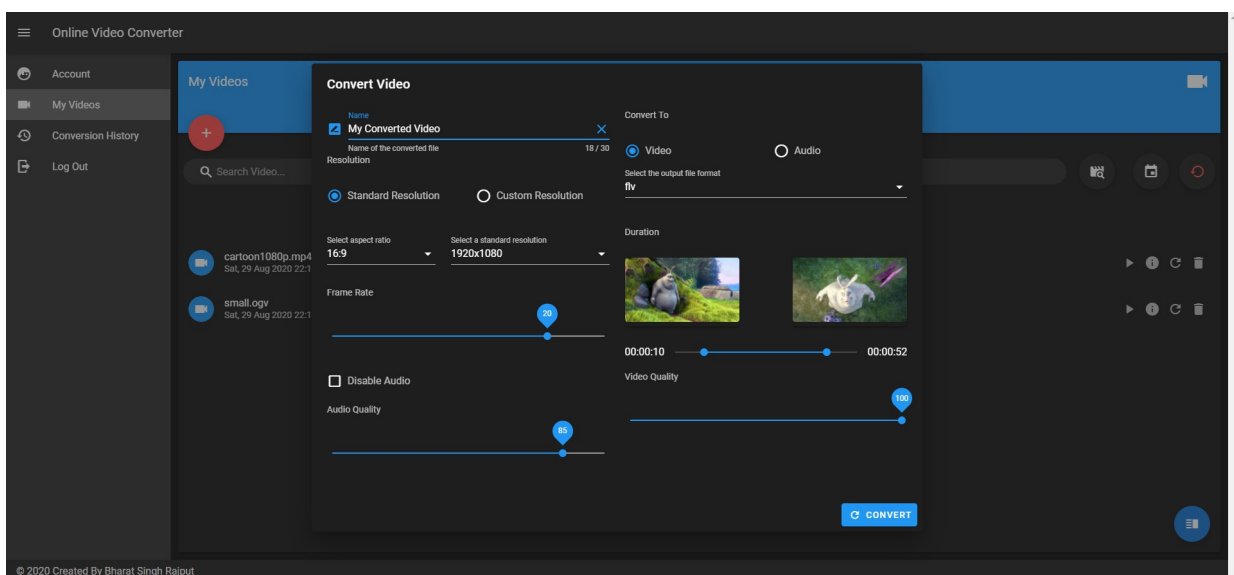
Video Information



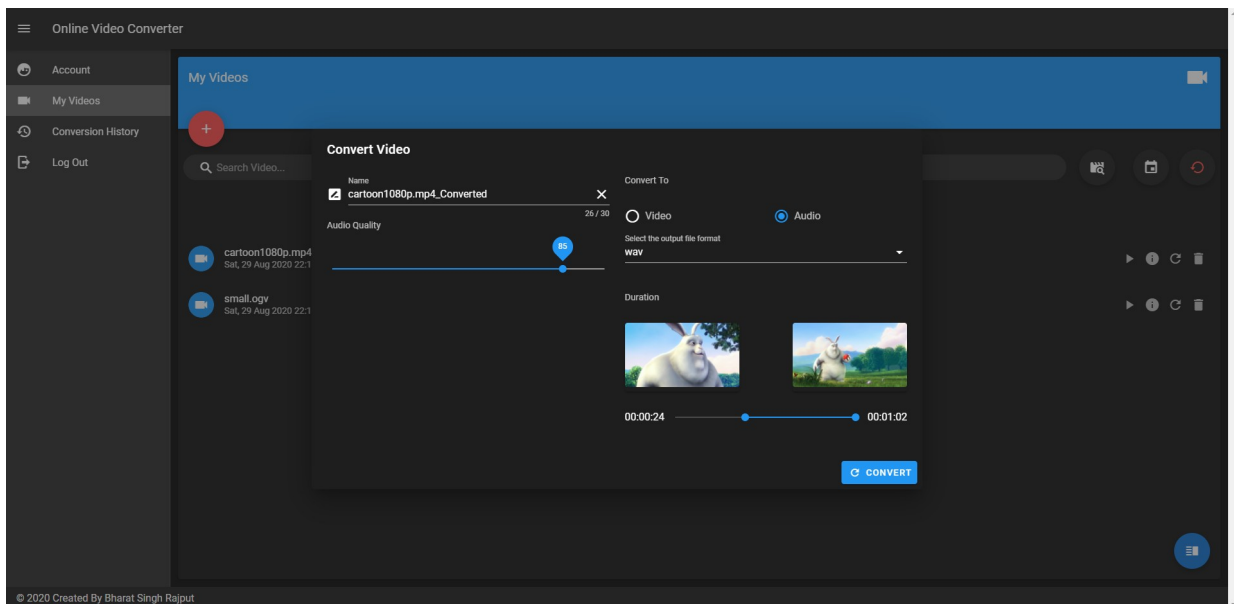
Video Streaming



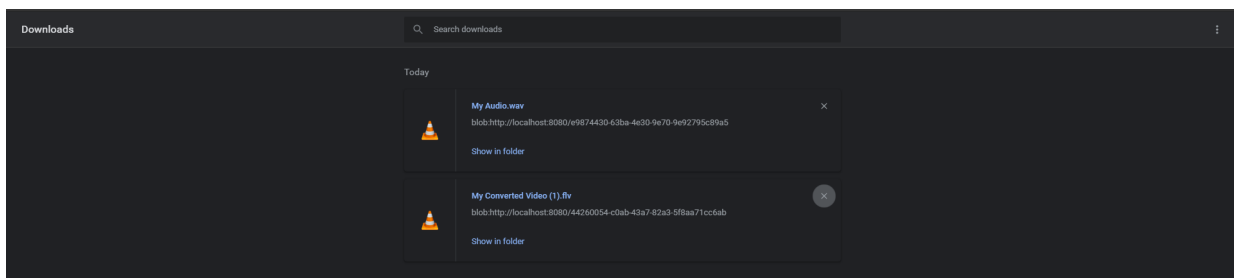
Upload New Video



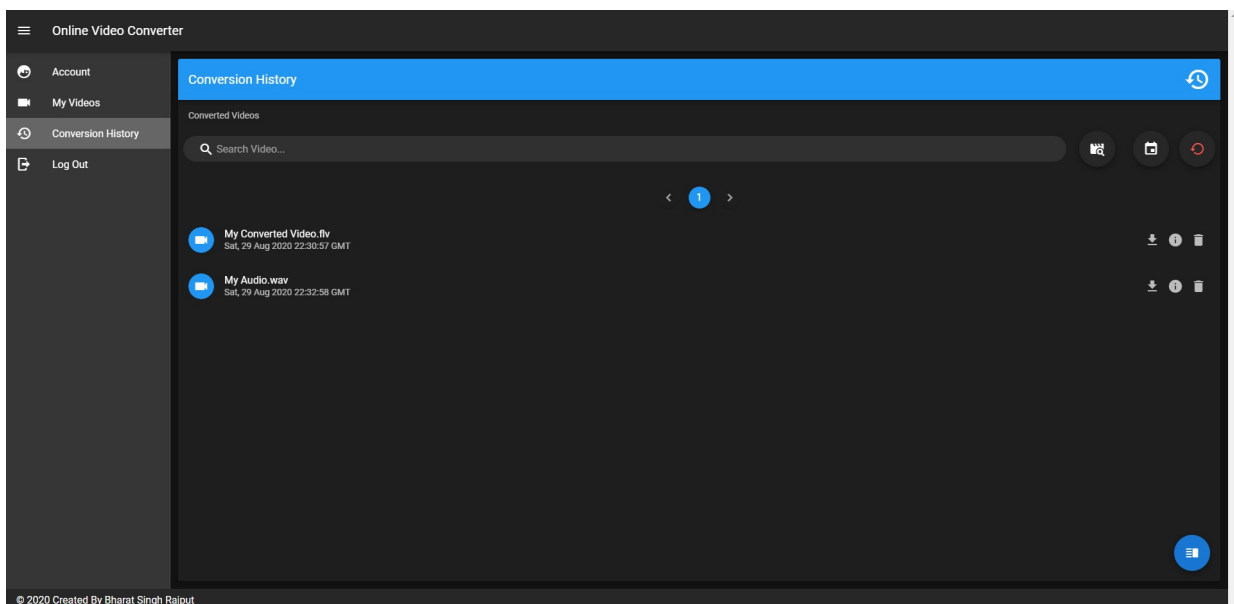
Video To Video Conversion



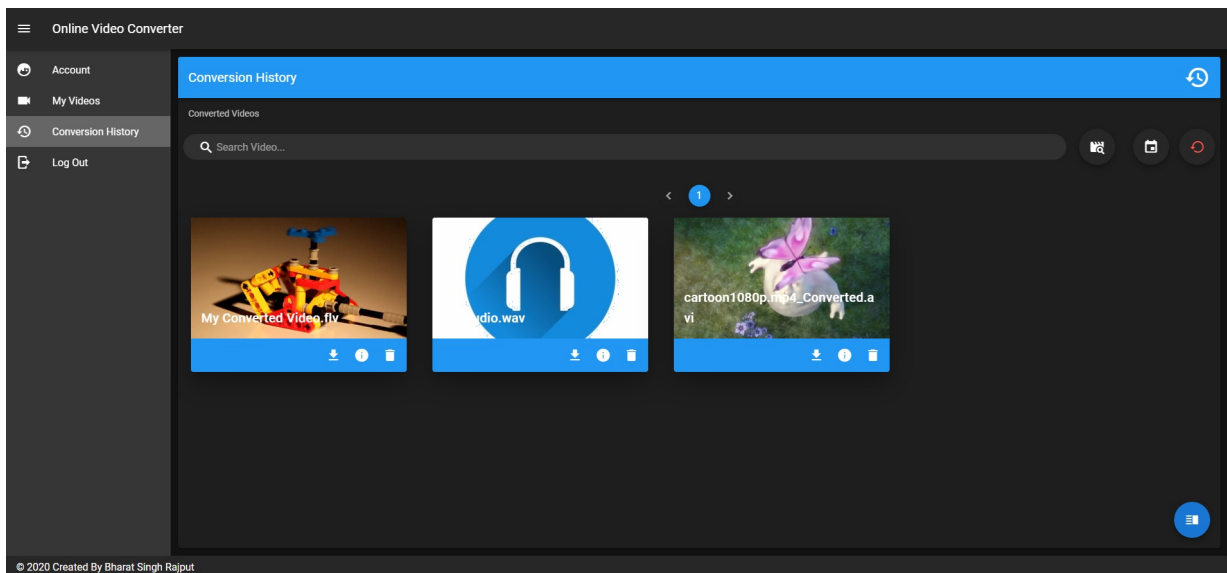
Video To Audio Conversion



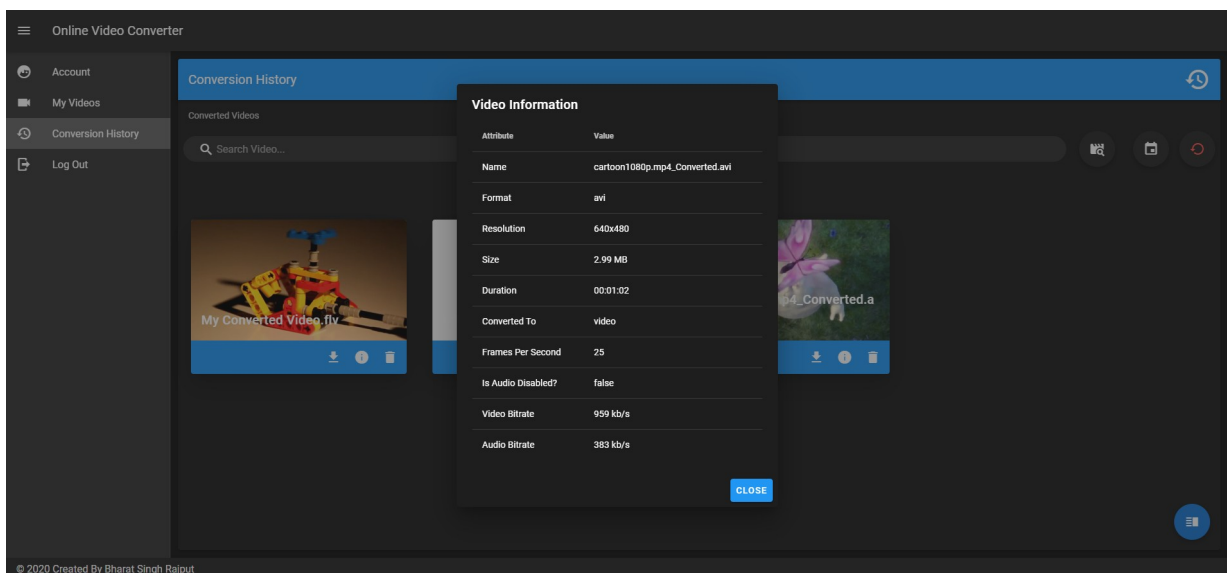
Converted Files Are Downloaded Automatically



User Conversion History List View



User Conversion History Card View



Converted Video Information

11 Implementation

Implementation or execution in any project is done when the project planning is complete as detailed as possible. In this phase, it monitors everything that could jeopardize the project or part of it in other words the theory is being tested by the implementation to make sure the data in the system changes according to the plan. To keep track of the project plan and keep ourselves in control all the tools and steps involved are applied by the project manager.

Steps in which all the tools are used:

- Take action according to plan.

- Record and document all the work, changes and results.
- Analyze the result of action or changes made according to the plan.

The steps involved here in this particular phase are:

- Installation of Hardware and Software
- Conversion of Data
- System Training
- Documentation of the System

11.1 Installation of Hardware and Software

Before implementation of the hardware and the software required for debugging the system should be made fully operative so that the translation can occur smoothly which is one of the most critical and costly activities in the system development life cycle.

11.2 Conversion of Data

As the old system is converted into a new system similarly the data from the old system should also be converted to run in the new system within new format. In this step the recovery procedures are needed to be fully defined and also the database is configured with security. .

11.3 System Training

All the programs of the system are loaded onto the computer systems of the user during this step. After loading the system we check by whether the project is under the control of the user and works according to the plan. To confirm the above sentence we need to complete the following steps:

- To execute the package
- To pass the data
- To process the data
- To take out the reports/records

12 Testing

The testing phase is very important for a successful system. In this phase before implementing the new system into operation, for eliminating bugs a test run of the system is done. After completing codes for the whole programs of the system, a test plan should be developed and run one given set of test data.

Using the test data subsequent test run is carried out:

- **12.1 Unit Testing**

It is a method by which individual units of source code, sets of one or more program modules collectively with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit can be an entire module but is more commonly an individual function. In object-oriented programming, a unit is an entire interface but could be an individual method. A unit test is created by programmers or by white box testers during the development process.

Each test case is independent of the others: substitutes like method stubs, mock objects can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended. Its implementation can vary from being very manual to being formalized as part of build automation.

- **12.2 Integration Testing**

It is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

- **12.3 Program Testing**

After the programs have been coded, compiled, and carried out to working conditions, they must be independently tested with the prepared test data. Any unwanted happening should be noted and debugged.

• 12.4 System Testing

When the program test for each of the programs of the system is written and errors are removed then the system test is complete. At this stage, the test is done on actual data. The complete system is put into execution on the actual data. At every stage of the execution, the output of the system is studied. During the outcome analysis, it may be found that the outputs are not matching the estimated output of the system. In such a situation, the bugs or errors in the particular programs are recognized and are fixed and further verified for the expected output. When it is confirmed that the system is running error-free, the users are called with their real data so that the system could be presented running as per their requirements.

The project is tested by Peer Testing under a controlled environment and encountered bugs and recommendations are kept under the light.

13 System Security Measures

Some of the measures can be taken as follows:

• Data Validation

Data validation checks have been performed on the client-side. However, if someone tries to send unvalidated data through raw HTTP requests, validations are also implemented on the server-side to prevent invalid data to be stored in the database.

Client-Side Validation

```
<v-form v-model="valid" ref="form">
  <!-- Email -->
  <v-text-field
    label="Email"
    name="email"
    prepend-icon="mdi-email"
    type="email"
    hint="Your existing email"
    v-model="user.email"
    maxlength="30"
    counter="30"
    clearable
    :rules="emailRules" <!-- Validation Data -->
  />
  ...
```

```

<script>
...
emailRules: [(v) => /.+@.+.test(v) || 'E-mail must be valid'], // Regex Validation
...

```

Server-Side Validation

```

/* Mongoose Validates Data Before Passing It To MongoDB */
const schema = new mongooseClient.Schema(
{
  displayName: { type: String, required: true },
  email: {
    type: String,
    unique: true,
    lowercase: true,
    required: true,
  },
  password: { type: String, required: true },
  ...

```

• User Authentication

Online Video Converter implements a secure JWT token-based authentication which is an internet standard nowadays. The REST API is secured using tokens and only authenticated users with valid tokens can use the service.

```

# Send Get Request To Server Without Authentication

curl --location --request GET 'http://localhost:3030/uploads' \
--data-raw ''

# Server Responds With Appropriate Error

{
  "name": "NotAuthenticated",
  "message": "Invalid authentication information (no `strategy` set)",
  "code": 401,
  "className": "not-authenticated",
  "errors": {}
}

```

Performing Authentication

```

# Send Login Credentials To The Server To Receive JWT

curl --location --request POST 'http://localhost:3030/authentication' \
--header 'Content-Type: application/json' \
--data-raw '{
  "email": "ch3rag@gmail.com",
  "password": "Abc@1234",
  "strategy": "local"
}'

# Server Responds With Appropriate Token And User Data

```


streaming no matter what is the format of the video. However, it also adds overhead to the server.

14.2 Future Scope Of The Project

- The backend, being a REST API, can be integrated into any frontend and thus, opening a huge amount of possibilities such as creating a native iOS or Android app implementing cloud conversion service.
- The backend service itself can be used as a third party service in any other application implementing some sort of video conversion. For example, a video compression app on Android or iOS can use our REST API in the background to perform the conversion task.
- In the future, with appropriate resources, this system can be upgraded to a subscription-based service where users have to buy tokens for video conversion. Free plans can also be continued for small videos. An option for buying additional cloud storage can also be added.

15 Bibliography

Following websites and documentations are referred to create this project:

- **Vue.js Docs** <https://vuejs.org/v2/guide/>
- **Vuetify.js Docs** <https://vuetifyjs.com/en/getting-started/quick-start/>
- **feathers-vuex Docs** <https://vuex.feathersjs.com/api-overview.html>
- **node.js Docs** <https://nodejs.org/en/docs/>
- **express.js Docs** <https://expressjs.com/en/api.html>
- **feathers.js Docs** <https://docs.feathersjs.com/api/>
- **fluent-ffmpeg Docs** <https://github.com/fluent-ffmpeg/node-fluent-ffmpeg>
- **mongoose.js Docs** <https://mongoosejs.com/docs/api.html>
- **mongoDB Docs** <https://docs.mongodb.com/>

16 Glossary

- **Activity:** In system development life cycle- a group of logically related tasks that make it possible to accomplish a specific objective; a group of related tasks.
- **Analysis:** Breaking a problem into successively manageable parts for individual study.
- **Cost/Benefit analysis:** The process of comparing projected savings and benefits to projected costs to decide whether a system change is justified.
- **Cost Estimation:** Approximation of the cost of a program, project, or operation.
- **Database:** A store of integrated data capable of being directly addressed for multiple uses; it is organized so that various files can be accessed through a single reference based on the relationship among records in the file rather than the physical location.
- **Data Dictionary:** A set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database.
- **Database Management System (DBMS):** The software that determines how Data must be structured to produce the user's view; manages, stores, and retrieves data and enforces procedures.
- **Data flow:** Movement of data in a system from a point of origin to a specific destination – indicated by a line and arrow.
- **Data Flow Diagram (DFD):** Graphic representation of data movement, processes, and files (data stores) used in support of an information system.
- **Design:** Process of developing the technical and operational specifications of a candidate system for implementation.
- **Documentation:** A means of communication; a written record of a phase of a specific project; it establishes design and performance criteria for phases of the project.
- **Entity:** Also called a data aggregate; something of interest to the user about which to collect or store data; represents a number of data elements.
- **Feasibility Study:** A procedure that identifies, describes, and evaluates candidate systems and selects the best system for the job.
- **Implementation:** In system development – a phase that focuses on user training, site preparation, and file conversion for installing a candidate system.

- **Information:** A meaningful set of data that tells something about the data relationships. 95
- **Initial Investigation:** An exploratory activity that determines whether a user request is valid and feasible before a course of action is recommended.
- **Interdependence:** Dependence of a system's components on one another for effective functioning.
- **Invalid Data:** Consists of problems, where the administrator tries to or accidentally fills the incorrect data into the form fields, such as incorrect date, characters instead of numbers, etc.
- **Maintenance:** To keep the thing in its consistent and original condition.
- **Model:** A logical or mathematical representation of a system that encompasses features of interest to the user.
- **Modularity:** In systems maintenance – a system is constructed in modular units of a limited size to simplify maintenance when necessary; in software reliability – the ease with which a package can be modified.
- **Operating System:** In data base – machine-based software that facilitates the availability of information or reports through the DBMS.
- **Organization:** Implies structure and order; a procedure that determines how components must be arranged to achieve objectives.
- **Password:** Identity authenticator; a key that allows access to a program, system, or procedure.
- **Planning:** Is studying a project course of action and determining what is to be done to meet stated goals.
- **Post-Implementation Review:** Evaluation of a new system after it has been in operation to determine its actual performance against expectations.
- **Process:** A procedure that transforms input into useful output; in a data flow diagram – indicated by a bubble or a circle.
- **Record:** A collection of aggregates or related items of data treated as a unit.
- **Relation:** Two-dimensional table.
- **Reliability:** Dependability or level of confidence; in systems work - the need to gather dependable information for use in making decisions about the system being studied.
- **Security:** The protection of data or hardware against accidental or intentional damage from a defined threat.

- **Source Code:** A procedure or format that allows enhancements on a software package.
- **System:** A regular or orderly arrangement of components or parts in a connected and interrelated series or whole; a series or group of components necessary to some operation.
- **System Design:** Detailed concentration on the technical and other specifications that will make the new system operational. 96
- **System Development:** The process of identifying the user's needs and designing a system that meets those needs through implementation.
- **System Development Life Cycle:** A structured sequence of phases for implementing an information system.
- **System Integrity:** The proper functioning of hardware, programs, and physical security, and the required degree of safety against eavesdropping and wire-tapping.
- **Systems Testing:** Testing the whole system by the user after major programs and subsystems has been tested.
- **System Analysis:** Reduction of an entire system by studying the various operations performed and their relationships within the system; an examination of a business activity with a view to identifying problem areas and recommending alternative solutions.
- **Task:** The smallest unit of work that is assigned to one person and controlled through a project management routine.
- **Unit Testing:** Testing changes made in an existing or new program.
- **Usability:** A criterion in software selection – easy to operate and user-friendly.
- **User Acceptance Test:** A test that verifies for the user that the system's procedures operate to system specifications and the integrity of vital data is maintained.
- **Validation:** Checking the quality of software in both simulated and live environments.
- **Variable:** A measurable quantity that has a definite numerical value at every instant.
- **Variable Cost:** Cost that varies with the volume of processing or number of shifts per day; examples are employee wages and costs of supplies and raw materials