

|  |    |
|--|----|
| Documentation . . . . .                                  | 3  |
| Team Overview . . . . .                                  | 5  |
| Weekly achievement . . . . .                             | 7  |
| VShare project description - 5 Vegetable Birds . . . . . | 11 |
| Meetings . . . . .                                       | 14 |
| Client Meetings . . . . .                                | 15 |
| 10.08.2022 First Meeting with Client . . . . .           | 16 |
| 17-08-2022 . . . . .                                     | 17 |
| 21.09.22 . . . . .                                       | 18 |
| 20/10/2022 . . . . .                                     | 19 |
| Supervisor Meetings . . . . .                            | 20 |
| Team Meetings . . . . .                                  | 21 |
| 16-08-2022 . . . . .                                     | 22 |
| 23-08-2022 . . . . .                                     | 23 |
| 30-08-22 . . . . .                                       | 24 |
| 3.09.22 . . . . .  | 25 |
| 13.09.22 . . . . .                                       | 26 |
| 27.09.22 . . . . .                                       | 27 |
| 17/10/2022 . . . . .                                     | 28 |
| 18/10/2022 . . . . .                                     | 29 |
| Communication . . . . .                                  | 30 |
| Updates till 2/09 . . . . .                              | 31 |
| Updates till 28/09 . . . . .                             | 39 |
| update until 22/10 . . . . .                             | 45 |
| UI Design . . . . .                                      | 56 |
| Homepage . . . . .                                       | 57 |
| Sorting method . . . . .                                 | 58 |
| The video page . . . . .                                 | 59 |
| Account homepage . . . . .                               | 60 |
| Upload Page . . . . .                                    | 61 |
| Super Account Page (Administrator Page) . . . . .        | 62 |
| Account sign in/up . . . . .                             | 63 |
| Requirements . . . . .                                   | 65 |
| Development . . . . .                                    | 67 |
| Architecture . . . . .                                   | 68 |
| Architecture Diagram . . . . .                           | 69 |
| UML Diagram . . . . .                                    | 70 |
| Motivational Model (MM) . . . . .                        | 71 |
| Team Roadmap . . . . .                                   | 72 |
| Assessment . . . . .                                     | 73 |
| Week 6 progress . . . . .                                | 74 |
| Week 9 Progress . . . . .                                | 78 |
| Week 12 Process Checklist . . . . .                      | 84 |
| Team decisions . . . . .                                 | 90 |
| Testing . . . . .  | 91 |

|   |     |
|---|-----|
| B (back end) - General instruction .....                          | 94  |
| B - Sign up testing .....   | 96  |
| B - Sign in testing .....   | 100 |
| F (front end) - Sign up testing .....                             | 102 |
| F - Sign in testing .....   | 103 |
| F - reset password testing .....                                  | 105 |
| F - sort by like number .....                                     | 106 |
| F- Update profile imformation tests .....                         | 107 |
| F - search the video by name .....                                | 109 |
| F - Delete video testing .....                                    | 110 |
| F - Upload Video Function .....                                   | 111 |
| F - Delete user as well as all related videos in admin mode ..... | 113 |
| F - Update user information in admin mode .....                   | 114 |
| Quality .....   | 116 |
| Deployment .....  | 117 |
| Peer Feedback .....   | 118 |
| Xuxiao He .....   | 119 |
| Qi Guo .....  | 120 |
| Congcong Zhao .....   | 121 |
| Jialin Ding .....   | 122 |
| Liping Meng .....   | 123 |
| Product Report .....  | 124 |

# Documentation

## Welcome to your documentation space!

Here are a few suggestions to get started:

- Explain how this documentation space should be used by selecting the  Edit button and customizing this Overview page
- Select Create  to make a new page in your space
- Add links to tools, guides, and other documentation resources
- Bookmark our [documentation guide](#) to learn how to create and improve documentation
- Explore the sample pages we've created for you

 [How-to article](#)  [UNDEFINED](#)  [Meetings](#)  <https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/163957> Ca  
n't find link

 Remove this panel once you're ready to share your space with team members.

 Documentation overview |  User guide |  Most viewed documents |  Recently updated

 Search for a doc

## i Documentation overview

Explain the purpose of this space and provide a high-level overview of what documentation is included.

## User guide

Explain how your documentation is organized and how users should navigate this space. Type `/IMAGE` to add a diagram.

## Most viewed documents

Add the most viewed documents by typing `/LINK`.

## Recently updated

You'll see the 5 most recently updated pages that you and your team create.

### Product Report

Nov 10, 2022 • contributed by [Jialin Ding](#)

### Congcong Zhao

Nov 10, 2022 • contributed by [Congcong Zhao](#)

### Jialin Ding

Nov 10, 2022 • contributed by [qi guo](#)

### Liping Meng

Nov 10, 2022 • contributed by [qi guo](#)

### Qi Guo

Nov 10, 2022 • contributed by [Xuxiao He](#)



# Team Overview

## About us

We are the most fabulous 5 people among all 490 students taking IT Project COMP30023 semester 2, 2022.

We have been contributing to our belovely computing system study for 2+ years and is looking forward to further explore and develope it :)

Our goals: High Completeness, High Satisfaction, High Productivity

We are looking to 120% accomplish all tasks handed to us. We wish our client always happy and satisfied with what we provide. We will finish all tasks with quality and efficiency.

| Team members  | Student ID | Student E-Mail                  |
|---------------|------------|---------------------------------|
| Xuxiao He     | 1107130    | xuxiaoh@student.unimelb.edu.au  |
| Lipeng Meng   | 1166308    | lipingm@student.unimelb.edu.au  |
| Qi Guo        | 1107125    | guoqg@student.unimelb.edu.au    |
| Congcong Zhao | 1107156    | congcong@student.unimelb.edu.au |
| Jialin Ding   | 1166285    | jialding@student.unimelb.edu.au |

| Name          | Role                | Responsibility   |
|---------------|---------------------|--|
| Xuxiao He     | Front-end Developer | Front-end, testing, UI Design, mainly for homepage, profile page and upload page               |
| Qi Guo        | Front-end Developer | Front-end, testing, UI Design , mainly for the log and sign up page and category page          |
| CongCong Zhao | Front-end Manager   | Front-end, testing, UI Design, mainly for video part and administrator account management part |
| Jialin Ding   | Back-end Manager    | Back-end, mainly for video part, Documentation, testing, Planner                               |
| Liping Meng   | Back-end Developer  | Back-end, mainly for user part Documentation, testing, Planner                                 |

## Catalog

- [Introduction & Project Descriptions](#)
- [Client meetings](#)
- [Supervisor meetings](#)
- [Team Meetings](#)
- [Requirements](#)
- [Development](#)



# Weekly achievement

This page will record all group works been done during the week.

## Week 3-4 10/08 - 16/08

- Had a meeting with the client
- Requirements 1st draft
- UI design (1.0 Version)
- decide design tool, Front-End: JavaScript, Vue. Back-End: JavaScript, Node.js

## Week 4-5 17/08 - 23/08

- Team meeting checking on everyone's process
- Requirements 2nd draft
- UI design (1.1 Version)
- Motivation model
- Github activity (Backend - installed dependency, express + tslint)
- Continue study coding

## Week 5-6 24/08 - 30/08

- Team meeting checking on everyone's process
- Requirements 3rd draft
- UI design (1.2 Version)
- Github activity (Backend - connected to MongoDB, sign in/up)
- Continue study coding

## Week 6-7 31/08 - 6/09

- Team meeting checking on everyone's process
- Progress checklist
- Backend coding - Upload videos
- Fronted coding - Account page, video playing page, sign in/up and searching page

## Week 7-8 7/09 - 13/09

- Backend coding -
  - done
    - upload video part
    - like & dislike video
    - delete the account for the super account

- get all data from mongo DB such as id, username, and email for the super account
- fix password -reset problem
- still doing :
  - Search a single user (for normal user and administrator)
  - Search a video
  - user can update(edit) personal information.
- Frontend coding -
  - done
    - stylesheet for uploading video page
    - stylesheet for administrator page
    - stylesheet for sign-in/sign-up page
    - stylesheet for video-playing page
    - stylesheet for searching page
    - stylesheet for the profile page
  - still doing :
    - adding functional requirements on pages
    - try to make connections with the local servers for sending and getting data
- Team discussion in progress:
  - how to show and store videos
  - how to modify the data forms and improve functions.

### **Week 8-9 14/09 - 20/09**

- Backend coding
  - done
    - Search a user/video
    - add more details when the user signs up, including age & gender
    - edit user's information
    - change log in part. Section when return token and user name when user login
  - Still doing:
    - Testing
- Frontend coding
  - done
    - a single unit of code for each functional requirement (administrator system for managing accounts, sign in/sign up, uploading videos, searching)
    - successful connections with the local server including sending requests to the server and getting data back (sign in and sign up functions change user password )
  - still doing :
    - Adding the Vue framework to the current code
    - considering what kind of data and variables should be used as the access for some functions (like how to maintain the login status)
    - unit testing for each function
- Team discussion in progress: consider project integration and maintenance as well as following modifications

## Week 9-10 21/09 - 4/10

- Client meeting
  - Demonstrated current work
  - The client came up with new requirements, user could delete video
- Backend coding
  - Delete video
  - Video streaming
  - Add age & gender
  - Correct input formats
- Frontend coding
  - update components, all tools(html.css) into Vue
  - router navigation
  - animation on functions
  - add and link specific functions to each page

## Week 10-11 5/10 - 11/10

- Backend coding
  - Delete video
  - Video streaming
  - Add age & gender
  - Correct input formats
- Frontend coding
  - done
    - improve single unit of code for functional requirement **account profile** (now it can track all videos user uploaded)
    - improve single unit of code for functional requirement **manage account information** (now it can directly update user's information in account profile)
    - improve single unit of code for search videos sorting function depending on like or dislike numbers in descending or ascending order
    - improve single unit of code for functional requirement **administrator system** (add a search bar to the administrator system now it can search user account by username or email)
  - still doing :
    - Trying to integrate all current completed pieces of work together using Vue router
    - considering how to resolve the path of video sources
    - unit testing for other uncompleted functions like **delete videos**, **play the video** as well as **upload videos**

## Week 11-12 12/10 - 21/10

- Backend
  - coding - add search video by category
  - coding - add video descriptions
  - coding - optimization
  - Deployment
  - Testing
- Frontend coding

- done
  - add preview picture for each video
  - add help and team page
  - add video paging function (now all videos would be soperated into several pages)
  - improve video category part (user can quickly find videos by video category)
  - integrate all Vue components and test all funtions of the whole website (project) and fix minor bugs
  - test the delete videos, play videos as well as upload videos (now video can be deleted, uploaded and played)
  - solve the path of videos (now it can get video path form the backend and play the video)
  - finish all requirements with high and medium priority required by client
- possssible further expectation:
  - try to add more css style for each component
  - try to imporve security issue
- Team presentation: Presentation PPT and assign presentation content for each member
- Client meeting - Demonstrated product and asked for feedback

# VShare project description - 5 Vegetable Birds

## Introduction

VShare is a video website for users to share and upload their videos. Users can have access to more functions like making comments, and storing their favorite videos. We aim to provide users with user-friendly interface so that it can be available to users of all ages. We design a navigation bar that can be easy for new users to find videos they want by using the search bar and category list. On the homepage you can see a sliding panel which provides users with the hottest and most popular videos at the moment.

Github link: <https://github.com/Kisupan/IT-PROJECT.git>

Website: Heroku | Welcome to your new app! (vshare-5vb-au.herokuapp.com)

## Roles and Responsibilities

Below is our roles and responsibilities for this project:

| Name          | Role                | Responsibility   |
|---------------|---------------------|--|
| Xuxiao He     | Front-end Developer | Front-end, testing, UI Design, mainly for homepage, profile page and upload page               |
| Qi Guo        | Front-end Developer | Front-end, testing, UI Design , mainly for the log and sign up page and category page          |
| CongCong Zhao | Front-end Manager   | Front-end, testing, UI Design, mainly for video part and administrator account management part |
| Jialin Ding   | Back-end Manager    | Back-end, mainly for video part, Documentation, testing, Planner                               |
| Liping Meng   | Back-end Developer  | Back-end, mainly for user part Documentation, testing, Planner                                 |

## Required Functions

The purpose of this project (VShare) is to develop a video website similar to Youtube and Tiktok. Main required functions of this web are playing, commenting and uploading videos, searching, sign-in, and sign-up, managing and editing personal information. The client also required an administrator account to manage all registered accounts.

## Timeline

This project would be split into 3 stages

- Design Stage (Week 1-3)

In the design stage, the team would first actively communicate with the client and summarize a list of requirements for the project, then demonstrate a sample. Once the client gets satisfied with the sample and requirements, the team will move on to the next stage. All the video format is accepted.

- Coding Stage (Week 4-10)

In the coding stage, the team would try their best to achieve most functions according to their priorities.

- Testing and deploying Stage (Week 11-12)

In the testing stage, the team would focus on 2 areas. One is the quality of the product itself if there are any bugs or missing functions.

The other is to check on client's opinions, what they think, and what they could like to change or add. The team will also deploy the project if the client requires it.

## Meeting Arrangement

There are three types of meetings, each has a different time and purpose.

- Client meeting:

Client meetings are held on every other Wednesday, sometimes less frequent subject to the client's availability. Additional meetings will be held if necessary, ie. enquiry changes, uncertainties.

During client meetings, client's requests would be discussed in details. In designing stage, team members would raise their concerns about blurry instructions, check certain task's priority, and discuss ways and approaches to achieve those. In deploying stage, the team will update and demonstrate client with its progress. In testing stage, the team would once again confirm all client's requirements have been met.

- Supervisor meeting

Supervisor meetings are held every workshop.

During supervisor meetings, the team would perform a stand-up presentation on what the whole team and individual have done during the past week. Team would also ask for suggestions on specific topics including planning, coding, and improvements.

- Team meeting

Team meetings are held twice per week. Additional meetings will be held to clarify certain content.

During team meetings, each member will briefly report their progress so the team can discuss and decide on what changes need to be done, what tasks are on next, how to achieve, and set new deadlines. It is also a great chance to raise any obstacles during individual work, so other team member could help.

## Decisions on Front-end

For the front-end, we decided to use HTML, CSS, JavaScript, Vue, axios and Bootstrap based on the following considerations:

- Bootstrap

Bootstrap is a powerful, extensible, and feature-packed frontend toolkit containing the most popular CSS Framework for developing website, saving lots of time.

- Vue as a framework

Vue provides more built-in features and companion libraries from the core team. Vue framework is more friendly for a newer developer and not as familiar with advanced JavaScript concepts. This makes the development faster and smoother.

- Axios

Axios is a promised-based HTTP request library. It can combine with Vue framework to make HTTP requests from the vue.js and handle the transformation of request and response data because of Vue's response-based data property.

## Decisions on Back-end

For the back-end, we decided to use a combination of node.js, Express, and Mongo DB according to the following consideration:

- node.js. as environment

- Front and back-end unification.
- Easy to Learn and Quick to adapt. As new learners, developers can quickly study it and make it easy to understand.
- Improves App Response Time and Boosts Performance

- Express as framework.

- Using Express.JS would be able to do the code of both frontend and backend with the help of using JavaScript

- Express makes the process easier
- Using Express.JS for backend development could be able to scale applications quickly
- Mongo DB as database
  - Full cloud-based developer data platform.
  - Flexible document schemas.

## Decisions on Other Tools

For other helper tools, we are using based on following consideration:

- Confluence - Documentation

Confluence has a strong organizing function which is perfect for recording all documentations. Words, files, and diagrams are categorized and stored for easy look-up. Live editing with team is also great for working collaboratively.

- Trello - Task management

Trello is simple to use and view all past, present, and future tasks. Our Trello units are divided by weeks. Each task has a deadline and is assigned to specific team member(s) for a clear view. Some team members had experience of using Trello as well.

- Canva - Design

Besides its support to beginner-level design, Canva also has a massive templates database, which is extremely friendly to our team who are not very confident on appearance design. Some team members had experience of using Canva as well.

- Github - Coding repository

Github is a great back-up option and helps reduce the risk of losing works. It is very useful for version control and collaboration. All team members had experience of using Github.

- Lucidchart - UML diagrams

All team members had experience of using Lucidchart. It has powerful functionalities like the ability to drag a line from a shape and an auto-prompt for which shape to choose next, you can diagram as quickly as you think.

- Motivational model editor - Motivation model

This is highly recommended by our lecture. It supports live-editing so is good for team collaborating.

# Meetings

## Date

## Participants

- 
- 

## Goals

- 

## Discussion topics

| Time | Item | Presenter | Notes |
|------|------|-----------|-------|
|      |      |           | •     |
|      |      |           |       |

## Action items

- 

## Decisions



## Client Meetings

- Client meeting is held fortnightly on Wednesday 9 PM via Zoom, might change depends on client's availability
- Our group will inform client about our progress so far
- Client will raise their concerns and requirements

# 10.08.2022 First Meeting with Client

Meeting host: Jialin Ding

Meeting Details :

1. Introduce ourselves

2. Discuss the topics and expectation about the product: **keeping track of items like an item register like movie review site. Talking about more in next week meeting to determine the specific topic.**

3. Requirement: **(1) Basic functions: update, delete, change, look up, login** For example: User can search by name or by author  
**(2) Website**

4. Schedule the periodic meetings (per fortnightly)

5. Next Meeting time: 9:00 pm 17/08/2022

Meeting host: Jialin Ding

Meeting Details :

(1) Showed the client UI design(1.0 version) and check requirements with her such as if a user does not sign in, she/he could only watch videos and couldn't publish or commit. If a user signs in, she/he could publish videos and commit.

(2) The clients are pleased with almost all our functions of web and gave some feedback.

(3) Make the 'Upload' more obvious.

(4) Add a supervisor account

- Only need one supervisor account
- can access all users' information such as User ID, Username, email address.
- A supervisor account could add (create a normal account), delete(delete user), change(change normal user's name, email, basic user details), lookup(search the normal user).

(5) After check requirement, our member split the work.

## 21.09.22

Meeting host: Jialin Ding

Meeting time: 9:00 - 9:30 PM

Meeting Details :

1. Catch-up with client, demonstrated current progress: log in/out, sign up, searching, and UI designs. Overall, client is satisfied with the presentation so far.
2. Listed below are new requirements client came up with. We had let client know that since they are not major functions of the project and also it is a little bit late to bring these up, they will only be assigned low priority and our team cannot guarantee that they would be accomplished.
  - a. **Delete/edit/re-upload video:** We assigned it as low since our database don't have enough space for supporting storing many videos. Even current rules for uploading videos are limited (less than 15 mb).
  - b. **Admin page:** Paging users (ie. 20 user info per page) instead of displaying all in one page. We decided to assigned it as low since we already have a searching function which admin can navigate to a specific user easily. Paging is relevant pointless when searching can reduce the effort of searching someone.
3. Client has not decided if they would like to deploy the project.

20/10/2022

Showed the VShare to the client. The client is very satisfied with our product, all functions are available, some small details can be improved such as the appearance of the main page design, video upload horizontal screen as much as possible.

Client confirmed deployment is not needed.

| VShare page/function  | Client Satisfaction | Need to change?  |
|-----------------------|---------------------|--|
| Sign in               | Satisfied           |  |
| Sign up               | Satisfied           |  |
| home page             | good                | if a user do not sign in, profile picture of user should not be showed up. |
| user information page | Satisfied           |  |
| upload a video        | Satisfied           |  |
| delete a video        | Satisfied           |  |
| search function       | Satisfied           |  |
| delete a user         | Satisfied           |  |
| administrator page    | Satisfied           |  |
| design of web page    | good                | Could be improved  |

## Supervisor Meetings

- Supervisor meeting is held every Wednesday during workshop
- Supervisor will check on progress (group & individual)
- Supervisor will inform what need to be done by next week

## Team Meetings

- Team meeting is held every Monday around 8-9 PM, delivered either via Zoom or WeChat
- During the meeting, members will share about their process and discuss about their concerns

16-08-2022

16/08

Chatted with client, mind changed.

Client prefered to have a mock YouTube website, which will need a homepage, and personal page/channel.

After logged into, users can upload videos, like, and comment.

23-08-2022

(1) We shared with each other what we had individually done in the previous week such as motivational model , UI design

(2) Talk about details of design.

- We decided that a user cannot change her email, a user is bound by an email, if this user uses a new email, it means she creates a new account.
- We decided not to change the background of the UI design for the moment
- We decided to continue write Documentation for UI design and each member is responsible for a part

(3) We decide how to start the coding part. We start the back-end and front-end drawing the UML, and then the front-end starts. After the front-end team members finish some parts, they can jump to the back-end and help the back-end.

30-08-22

We discussed and worked on following concepts:

Backend - Mongo DB connection, will try to combine 2 versions of current work together tomorrow during workshop

Fronted - UML designs.

Whole group - Progressive checklist.

3.09.22

- Week 6 progress checklist
- Backend coding - Upload videos
- Frontend coding - Account page, video playing page, sign in/up and searching page

13.09.22

- Checked on current jobs:
  - Sign in/up, like/dislike testing successful
  - Sign in/up connection to the frontend successful
  - Able to delete account
  - Able to reset password
- Future plan:
  - Search account/video
  - Edit personal info
  - Displaying video

27.09.22

- Checked on current jobs:
  - Optimised reset password
  - Search videos/users
  - Pop-up windows
  - Style sheet
  - Sign in/up page frontend and backend connection
  - Progress checklist
- Future plan:
  - Liping - Sign in & reset password testing
  - Jialin - Sign up & search users testing
  - Congcong - Code maintenance on admin and video page
  - Xuxiao - Account page server connection
  - Qi - Code maintenance on searching page

17/10/2022

- Decided how to construct the structure of presentation
- Decided deploy after presentation
- Talked about how to entertain when presentation
- Talked about the week 12 progress

18/10/2022

Talked about the presentation

- Decided what to present
- Talked about content of each part
- Decided which team member is responsible for which part
- Optimized website pages of VShare
- Optimized the order of presentation content

## Communication

## Updates till 2/09



杰奎琳加固林贾归林(②)

client怪忙的今天问问她想每周见还是双周见呢

08-10 16:03:41



何徐笑



08-10 16:42:18



杰奎琳加固林贾归林(②)



08-10 16:59:48

杰奎琳加固林贾归林(②)

team name please

08-10 16:59:53



聪聪

996 coders

08-10 17:26:23

聪聪

codeholics

08-10 17:26:38



杰奎琳加固林贾归林(②)

taicanle

08-10 17:29:56



polaris  
nightmare coders

08-10 17:58:25



聪聪  
家人们

08-10 19:37:25

聪聪  
给client的introduction弄好了吗



polaris  
我正打算写一下

08-10 19:38:38

polaris  
但我要先交一个我心理课的作业



聪聪  
没事的

08-10 19:40:19

聪聪  
我把我的也搞一下



polaris  
好

08-10 19:42:24



(\*ΦωΦ\*)

周三晚上9点开会可以吗

08-10 00:35:45



杰奎琳加固林贾归林①

08-10 15:38:06

Zoom meeting on Aug 10, 2022 09:00 PM Canberra,  
Melbourne, Sydney

Join from PC, Mac, iOS or Android: <https://unimelb.zoom.us/j/86927025079?pwd=YmxPV1dVMmtVc1JIUTdxT2R3eFM1dz09>

Password: 488107

Need to dial-in instead? Enter the meeting ID: 869 2702 50  
79 via +61 3 7018 2005 or +61 2 8015 6011

Or join from a H.323/SIP room system:

Dial: [86927025079@global.zoomcrc.com](mailto:86927025079@global.zoomcrc.com) | or SIP: [86927025079@zmau.us](tel:86927025079@zmau.us) | or [103.122.166.55](http://103.122.166.55)

with meeting ID: 86927025079 and password: 488107

Help: <https://unimelb.service-now.com/it>

Legal: <https://www.unimelb.edu.au/legal>

聪聪

08-10 20:31:59

我们组名叫什么

聪聪

08-10 20:36:55

菜鸟

聪聪

08-10 20:37:00

这个很形象



杰奎琳加固林贾归林①

08-10 20:37:48

5 vegetable birds

- 
-  何徐笑 08-16 16:00:14  
家人们，今天晚上要不开个meeting
-  聪聪 08-16 16:14:04  
我们今天开meeting的话
- 聪聪 08-16 16:14:11  
做些什么呢
-  polaris 08-16 16:20:53  
画个motivation model ?
-  Bill国 08-16 18:24:31  
她说不是要搜索吗
-  聪聪 08-16 18:24:41  
这是基本功能
-  Bill国 08-16 18:24:43  
还有如果有空要什么推荐吗
- 聪聪 08-16 18:25:28  
现在确定的是
- 聪聪 08-16 18:25:35  
要用登录系统
-  Bill国 08-16 18:25:35  
帮人家决定不太好吧[破涕为笑]
-  聪聪 08-16 18:25:49  
搜索和评论

聪聪  
现在确定的是

08-16 18:25:28

聪聪  
要用登录系统

08-16 18:25:35

 Bill国  
帮人家决定不太好吧[破涕为笑]

08-16 18:25:35

 聪聪  
搜索和评论

08-16 18:25:49

---

 杰奎琳加固林贾归林②  
九点半吗?

08-16

杰奎琳加固林贾归林②  
zoom吗

08-16

杰奎琳加固林贾归林②  
还是微信

08-16

 聪聪  
都可以

08-16

 polaris  
可

08-16

 聪聪  
<https://unimelb.zoom.us/j/5606335020?pwd=TIVxcXZOUlZaeUQ2NUcvZWkwZzFudz09>

08-16

 杰奎琳加固林贾归林④  
九点半吗?

08-16

杰奎琳加固林贾归林④  
zoom吗

08-16

杰奎琳加固林贾归林④  
还是微信

08-16

 聪聪  
都可以

08-16

 polaris  
可

08-16

 聪聪  
<https://unimelb.zoom.us/j/5606335020?pwd=TIVxcXZOUlZaeUQ2NUcvZWkwZzFudz09>

08-16

 杰奎琳加固林贾归林④

08-16 23:35:20

杰奎琳加固林贾归林④  
朋友们

08-16 23:35:21

杰奎琳加固林贾归林④  
有空写写这个

08-16 23:35:32

@polaris 我们先看看这个学学把网站试着搭起来

杰奎琳加固林贾归林(1)

08-17 23:07:17

下周开会前做完：

1. 各自页面的UI、功能展示
2. Requirement
3. motivation model(轮流权限)
4. 学习有用的东西



聪聪

08-17 23:43:48

我们这周requirement写完给她看

聪聪

08-17 23:47:56

她是client有时候提一些我们可能办不到的要求 这个时候我们就  
需要协商了

聪聪

08-17 23:48:17

我个人感觉今天的meeting下来总体结果还是偏好的

聪聪

08-22 16:30:33

家人们今天晚上再开个会吧

聪聪

08-22 16:44:58

晚上9点老时间



何徐笑

08-22 21:05:26

要不就vx吧



聪聪  
晚上开个会?

08-30 18:07:40



polaris  
可

08-30 18:11:19



聪聪  
<https://space.bilibili.com/2035620040>

08-30 21:32:45



杰奎琳加固林贾归林

08-31 16:44:42



## Updates till 28/09



脱脱

晚上开会么家人们

09-12 18:23:46



杰奎琳加固林贾归林

半见?

09-13 21:02:34



polaris

我都可

09-13 21:04:31



脱脱

行的

09-13 21:04:44



何徐笑



09-07 16:31:30



polaris



09-07 16:47:50



杰奎琳加固林贾归林



09-07 16:48:59

杰奎琳加固林贾归林



09-07 16:57:34

杰奎琳加固林贾归林

你试试这个

09-07 16:57:49



polaris



09-13 21:52:45

杰奎琳加固林贾归林②

09-14 22:27:42

我在看数据格式



聪聪

我突然意识到

09-14 22:27:58

聪聪

09-14 22:28:22

好像后端大部分需要的是表单数据

聪聪

09-14 22:28:40

那我们前端稍微改一下就好了

聪聪

09-14 22:28:59

现在还打不开

聪聪

09-14 22:29:05

再看怎么改

---

聪聪

09-14 22:44:30



聪聪

09-14 22:45:22

后端大部分接收的是form形式的数据

聪聪

09-14 22:45:39

但是可以发个前端json的东西

聪聪

09-14 22:46:10

我们前端发东西中间处理一下就好了



杰奎琳加固林贾归林②

09-14 22:48:16

前端request body里面传json好像也一样能行



聪聪

09-14 22:48:45

是的

 聪聪 09-14 22:48:45  
是的

聪聪 09-14 23:10:39  
我连上后端了

聪聪 09-14 23:11:10  
sign in 没问题

 杰奎琳加固林贾归林② 09-14 23:20:33  
哇

杰奎琳加固林贾归林② 09-19 14:44:50  
要不要见一下client

 聪聪 09-19 14:45:03  
肯定的

 杰奎琳加固林贾归林② 09-19 14:45:12  
zoom吗

 聪聪 09-19 14:45:20  
行的

 杰奎琳加固林贾归林② 09-19 14:45:35  
我发个链接上来

 聪聪 09-19 14:48:50  
你们那个forgetpassword

聪聪 09-19 14:48:54  
好像有点问题

聪聪 09-19 14:49:06  
何徐笑试了半天

聪聪 09-19 14:49:12  
密码没改成功

聪聪 09-19 14:49:28  
但是返回了一个status 200

聪聪 09-19 14:49:32  
很奇怪

 polaris 09-19 15:00:25  
对，那个确实有问题，上周我改了改，但我好像还没push到  
github上



杰奎琳加固林贾归林②  
我有个问题

09-19 15:19:29

杰奎琳加固林贾归林②  
search的时候

09-19 15:19:35

杰奎琳加固林贾归林②  
是看关键字吗

09-19 15:19:56

杰奎琳加固林贾归林②  
就比如说

09-19 15:20:00

杰奎琳加固林贾归林②  
我打了5个字进去

09-19 15:20:11

杰奎琳加固林贾归林②  
四个吧打了个魑魅魍魉

09-19 15:20:48

杰奎琳加固林贾归林②  
然后我们没有匹配的东西

09-19 15:20:57

有一个视频里面只有魑字

杰奎琳加固林贾归林②  
还要显示吗

09-19 15:21:42



杰奎琳加固林贾归林②  
我觉得可以

09-19 15:22:33



杰奎琳加固林贾归林②  
对就是我在想是要完全匹配还是局部匹配

09-19 15:26:37

杰奎琳加固林贾归林②  
完全匹配

09-19 15:27:10

杰奎琳加固林贾归林②  
不用出东西哈哈哈哈哈哈

09-19 15:27:16

杰奎琳加固林贾归林②  
搜什么都没有可能

09-19 15:27:27



杰奎琳加固林贾归林②  
也是哦

09-19 15:27:29

杰奎琳加固林贾归林②  
那就出两个呗

09-19 15:27:40

 杰奎琳加固林贾归林②  
一般用户通道的搜索视频+用户

09-19 15:46:46

 聪聪  
像视频和图片这种流媒体文件

09-19 16:01:06

聪聪  
是要放到其他地方的

聪聪  
然后数据库里就是视频或者图片的url

 polaris  
对，放在了本地，视频路径存在了数据库里

09-19 16:02:41

 杰奎琳加固林贾归林②  
这是最好的

09-19 16:02:43

 聪聪  
能不能加个gender、date of birth在注册的时候

09-20 20:26:23

 polaris  
「  
聪聪：能不能加个gender、date of birth在注册的时候」  
-----  
可

09-20 20:28:11

 聪聪  
我注册界面也加一个

09-20 20:28:43

聪聪  
行的

09-20 20:29:36

 polaris  
我俩写的进度不一样，我等全部都Push上一次再改

09-20 20:29:42

polaris  
那修改个人信息，update的时候，gender，date of birth需要  
修改吗

09-20 20:32:59

 聪聪  
是的

09-20 20:33:33

 杰奎琳加固林贾归林②  
今天要抓client开会吗

09-21 11:18:39



聪聪

抓啊

09-21 11:18:58



杰奎琳加固林贾归林

要展示吗

09-21 11:19:24

杰奎琳加固林贾归林

展示的话

09-21 11:19:35

杰奎琳加固林贾归林

还是zoom吧

09-21 11:19:39

## Team 36 (7)



丁嘉霖 Jacqueline

Zoom meeting on Sep 21, 2022 09:00 PM Canberra, Melbourne, Sydney

Join from PC, Mac, iOS or Android: <https://unimelb.zoom.us/j/83802531705?pwd=cHRCemNCUXYyQ2oyN3dDSUxTNjEzUT09>

Password: 227101

Need to dial-in instead? Enter the meeting ID: 838 0253 1705 via +61 3 7018 2005 or +61 2 8015 6011

Or join from a H.323/SIP room system:

Dial: [83802531705@global.zoomcrc.com](mailto:83802531705@global.zoomcrc.com) | or SIP: [83802531705@zmau.us](mailto:83802531705@zmau.us) | or [103.122.166.55](http://103.122.166.55)

with meeting ID: 83802531705 and password: 227101

Help: <https://unimelb.service-now.com/it>  
Legal: <https://www.unimelb.edu.au/legal>

The University of Melbourne collects your personal information via Zoom to facilitate virtual meetings, webinars and events. This may include your name, email address and any personal information you share via Zoom during the session. The University's General Privacy Statement details how we collect and process personal information.

update until 22/10

 聪聪 10-02 20:56:58  
我search user这个api用get方法卡半天了

 聪聪 10-02 20:57:14  
咋办

 聪聪 10-02 20:57:37  
奇怪的是我用thunder测是可以搜出来的

 聪聪 10-02 20:57:47  
前端用get不行啊

 聪聪 10-02 20:58:31  


 聪聪 10-02 20:59:36  


 杰奎琳加固林贾归林 10-02 21:03:57  
因为这个是form

 聪聪 10-02 21:04:09  
讲个笑话

 聪聪 10-02 21:04:22  
我在thunder用json也能出来

 杰奎琳加固林贾归林 10-02 21:04:45  
新的版本是prama

 杰奎琳加固林贾归林 10-02 21:05:01  
但是新的版本传不上来

 杰奎琳加固林贾归林 10-02 21:05:40  
你的搜索

 聪聪 10-02 21:05:47  
你们controller那一块儿写的是form接收

-  杰奎琳加固林贾归林(宀) 10-02 21:05:58  
现在都基于你下面的框里
- 杰奎琳加固林贾归林(宀) 10-02 21:06:03  
打进去的东西
- 杰奎琳加固林贾归林(宀) 10-02 21:06:20  
跟search后面的没关系
- 杰奎琳加固林贾归林(宀) 10-02 21:07:00  
但是我后面写video的时候找到了一个聪明一点的读法
- 杰奎琳加固林贾归林(宀) 10-02 21:07:23  

- 杰奎琳加固林贾归林(宀) 10-02 21:07:34  
打叉的这里
- 杰奎琳加固林贾归林(宀) 10-02 21:07:39  
他只看你打勾那里进去的东西
-  聪聪 10-02 21:08:24  
:user那个不是get传递的参数么
-  杰奎琳加固林贾归林(宀) 10-02 21:08:28  
你去pull一个最新版的就好了
- 杰奎琳加固林贾归林(宀) 10-02 21:08:32  
git上
-  聪聪 10-02 21:08:49  
识别username="这种形式
-  杰奎琳加固林贾归林(宀) 10-02 21:09:08  
并不是
- 杰奎琳加固林贾归林(宀) 10-02 21:09:25  
他是两个分开的东西
- 杰奎琳加固林贾归林(宀) 10-02 21:09:52  
不过你下最新版就没这个问题了

 杰奎琳加固林贾归林②  
因为你们还没分哪一页用来search啥

10-02 21:16:29

杰奎琳加固林贾归林②  
所以目前只有search和video-search

杰奎琳加固林贾归林②  
后段所有test的路径都在confluence里有

 聪聪  
但是我看搜索不都是search/?这个形式么

10-02 21:17:12

 杰奎琳加固林贾归林②  
你对着用就行

10-02 21:17:19

 聪聪  
穿的是query参数

10-02 21:17:24

聪聪  
我先去试试新的

10-02 21:17:40

video-search才是搜视频的

10-02 21:18:21

 聪聪  
对啊

10-02 21:25:06

聪聪  
我们说的都是一个东西

10-02 21:25:14

聪聪  
只是我疑惑那个query

10-02 21:25:25

聪聪  
怎么直接写成? cc这个样子

10-02 21:25:46

聪聪  
照理说应该写成? username=cc

10-02 21:26:01

 杰奎琳加固林贾归林②  
query是这样

10-02 21:26:30

杰奎琳加固林贾归林②  
所以你按照confluence上的路径来

10-02 21:27:58

 聪聪 10-02 21:40:46  
我看其他的组都做完了在优化

 杰奎琳加固林贾归林 10-02 21:40:49  
我们还得弄这个视频怎么展示

杰奎琳加固林贾归林 10-02 21:41:15  
我们传也能传了

 polaris 10-02 21:41:19  
视频写完了不是，先存本地吧

 杰奎琳加固林贾归林 10-02 21:41:19  
但是放出来

杰奎琳加固林贾归林 10-02 21:41:23  
很难呢

 polaris 10-02 21:41:31  
不难吧

 杰奎琳加固林贾归林 10-03 15:53:34  
朋友们

杰奎琳加固林贾归林 10-03 15:53:54  
我今天晚上要test一下

杰奎琳加固林贾归林 10-03 15:54:18  
我把user数据先都清空下

杰奎琳加固林贾归林 10-03 15:54:22  
留一个admin

杰奎琳加固林贾归林 10-03 15:54:28  
还是可以用的

杰奎琳加固林贾归林 10-03 15:54:51  
然后我test那里

杰奎琳加固林贾归林 10-03 15:55:04  
留了三个账号以后可以用

 polaris 10-04 15:37:47  
朋友们，我写了个video streaming，应该可以用，不过我不会  
写前端测试，没有test

polaris 10-04 15:38:18  
如果想分段发送数，可以考虑用这个

polaris 10-04 15:38:23  
数据

 杰奎琳加固林贾归林② 10-05 01:36:12  


杰奎琳加固林贾归林② 10-05 01:36:20  
gender和age都能用了

 聪聪 10-05 01:36:51  
可

 杰奎琳加固林贾归林② 10-05 01:39:44  
好耶

我把mongo db里

杰奎琳加固林贾归林② 10-05 01:39:58  
之前的用户都删了

杰奎琳加固林贾归林② 10-05 01:40:09  
因为现在要填的东西多了

杰奎琳加固林贾归林② 10-05 01:40:13  
统一下呢

杰奎琳加固林贾归林② 10-05 01:40:35  
现在两个账号

杰奎琳加固林贾归林② 10-05 01:40:38  
一个user1

杰奎琳加固林贾归林② 10-05 01:40:41  
一个admin

杰奎琳加固林贾归林② 10-05 01:40:45  
都是能用的

杰奎琳加固林贾归林(⊙)

自己测试的时候就不要再创奇怪的账号了

10-05 01:41:21



聪聪

我创了可以删掉

10-05 01:41:41

聪聪

因为我是管理员 😊

10-05 01:41:52



杰奎琳加固林贾归林(⊙)

字符字数限制都加上去了

10-05 01:42:19

杰奎琳加固林贾归林(⊙)

你小心

10-05 01:42:22



聪聪

没问题

10-05 01:42:27



杰奎琳加固林贾归林(⊙)

如果报错501

10-05 01:42:53

杰奎琳加固林贾归林(⊙)

10-05 01:43:01

= (qs, xs, ts', q0', as')

where

ts' = [((q1,x1),r1) | ((r1,x1),q1) <- ts]

as' = [q0]

q0' = [z | z <- as]

杰奎琳加固林贾归林(⊙)

10-05 16:14:52

heroku

杰奎琳加固林贾归林(⊙)

10-06 17:11:22

昨天说要改啥



聪聪

update接口

10-06 17:11:41

聪聪

然后delete video接口不要有 /: video\_id这种

10-06 17:12:57



Bill国

给vedii加个属性

10-06 17:13:25

Bill国

news

10-06 17:13:26

Bill国

sport

10-06 17:13:29

Bill国

我再看看

10-06 17:13:43

Bill国

technology

10-06 17:13:50



10-06 17:14:46



10-06 17:15:09

Bill国

10-06 17:15:16

nxx

Bill国

能不能加个others

10-06 17:15:48

Bill国

@何徐笑

10-06 17:15:54

Bill国

你主页加个others

10-06 17:16:00



聪聪

10-06 17:17:53

只能更改username password gender和age

聪聪



10-06 17:18:48



polaris

update我今晚可以写写

10-06 17:39:35

polaris

不过delete我还没明白 直接改成update能用吗

10-06 17:39:52

 聪聪  
@polaris

10-07 17:33:12

聪聪

10-07 17:33:19

聪聪

我把你写的接口稍微改了一下

10-07 17:33:36

聪聪

因为我是管理员

10-07 17:33:57

聪聪

所以在修改账户信息的时候默认这些email username gender age都是有的

10-07 17:34:32

聪聪

所以我把判断输入是否为空都删了

10-07 17:35:04

聪聪

然后要修改密码我直接用的forgetpassword那个接口

10-07 17:35:49

 polaris  
好滴好滴okkk

10-07 18:16:13

 杰奎琳加固林贾归林②  


10-10 14:36:01

杰奎琳加固林贾归林②  
这个名字被占掉了

10-10 14:36:13

 聪聪  
VShare呢

10-10 14:37:06

 杰奎琳加固林贾归林②  
只能用小写

10-10 14:37:16

杰奎琳加固林贾归林②  
正好

10-10 14:37:52

杰奎琳加固林贾归林②  
我在deploy

10-11 12:31:11

杰奎琳加固林贾归林②  
晚上写吧

10-11 12:31:15

杰奎琳加固林贾归林②  
你们都要啥东西

10-11 12:31:19

杰奎琳加固林贾归林②  
列一下

10-11 12:31:22

杰奎琳加固林贾归林②  
写confluence写群里都行

10-11 12:31:44

杰奎琳加固林贾归林②  
写一下

10-11 12:31:48

何徐笑  
暂时就我说的那个

10-11 14:46:19

杰奎琳加固林贾归林②  
 10-12 13:44:41

杰奎琳加固林贾归林②  
好像这样就算deploy好了

10-12 13:44:50

杰奎琳加固林贾归林②  
你们连连看吧

10-12 13:45:15

聪聪  
现在upload那个接口能演示一下怎么传视频的么

10-13 19:25:39

聪聪  
我看那个接口多了验证什么的

10-13 19:25:53

polaris  
 10-13 19:28:41

polaris  
先signin

10-13 19:28:45

 聪聪 10-14 21:20:22  
我把你们的like 和 dislike接口改了一下

聪聪 10-14 21:20:42  
现在可以取消点赞了

 polaris 10-14 21:41:41  
可以可以

 何徐笑 10-15 00:23:06  
我们现在一共有几个video

何徐笑 10-15 00:24:17  
为什么我getallvideo

何徐笑 10-15 00:24:21  
一共就俩

何徐笑 10-15 00:24:47  
有一个路径还不在那个video文件夹里

 何徐笑 10-17 15:59:39  
今天晚上要不搞下ppt?

 polaris 10-17 16:01:02  
可

 polaris 10-17 21:15:44  


 杰奎琳加固林贾归林② 10-17 21:24:38  
zoom吧要不

杰奎琳加固林贾归林② 10-17 21:24:48  
可以share screen

 何徐笑 10-17 21:28:35  
<https://unimelb.zoom.us/j/84750807295?pwd=KzBLRUUVUTEFBWUNBM1I6T25KNkpVUT09>

 聪聪 10-18 21:09:48  
干活了各位

聪明

分配一下任务

10-18 21:10:08

 polaris  
要不还用zoom?

10-18 21:10:43

 杰奎琳加固林贾归林②  
我半来😊

10-18 21:10:46

 何徐笑  
<https://unimelb.zoom.us/j/86766824767?pwd=WXY3SklnQzMrZGk1MFY0amdsUVB2QT09>

10-18 21:45:28

 polaris  
<https://unimelb.zoom.us/j/86766824767?pwd=WXY3SklnQzMrZGk1MFY0amdsUVB2QT09>

10-19 00:18:27

 杰奎琳加固林贾归林②  
直接zoom里invite就行

10-19 00:18:37

杰奎琳加固林贾归林②  
今天present

10-19 00:18:37

 杰奎琳加固林贾归林②  
直接zoom里invite就行

杰奎琳加固林贾归林②  
今天present

10-19 15:06:14

杰奎琳加固林贾归林②  
进zoom

10-19 15:06:19

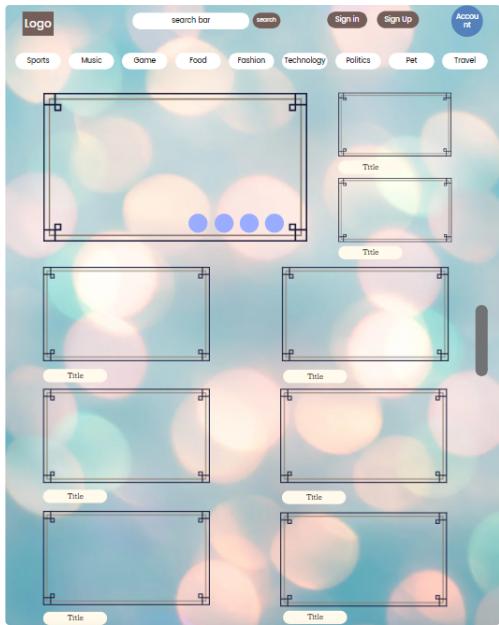
杰奎琳加固林贾归林②  
share screen就行了

10-19 15:06:24

## UI Design

This is our UI design of our website and we would like to provide client with user-friendly interface.

# Homepage



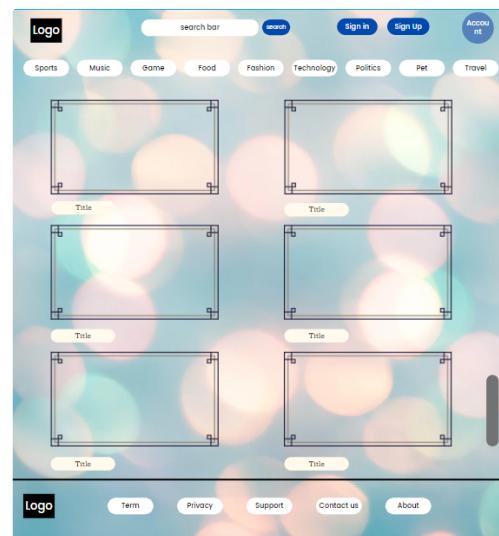
This is the home page for the website. There is a search bar on the top of the page and it can help users to search the video they want and it can easily save users lots of the time.

It is clearly that there is a log in system on our website and users can sign up for an account and then can have access to upload their videos and make comments on others' videos.

On the top right of the website, there is a profile and it can shows the photo of the account after user sign in.

Then, there is a video classification list below that has classified the videos so that users can quickly browse the videos of interest.

The main body of the page consists of a big section which shows the hottest videos and recommended videos user may interested in.



When scroll down the whole page you can see a service bar that shows our privacy and support.

## Sorting method



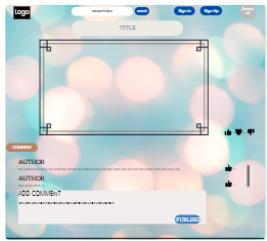
This is the sorting method that have high priority in our project. This method brings the convenience by useing labels, the user can easily find the vedios he/she want.

When people uploading the video, he/need to give the title of this video. Then the system will record this video's uploading time and its length automaticly and represent these information with the author's user name together beside the video in this page.

The next important interact function is the like, love and comment. The user can click the like button to add the popularity of the video he/she like, the video with more like will be represent at first. If the user click the love , he/she can find this video he/she love in his/her personal homepage. The last comment part can only represent the number of comments, if the user want to add comment, he/she need log on and click the vedio to access the another page to leave comment.

At bottom, there is some buttons for user to use to choose the pages he/she want to view. The page 1 and last page will always appear on this page to let user can turn to first and last page easily.

## The video page



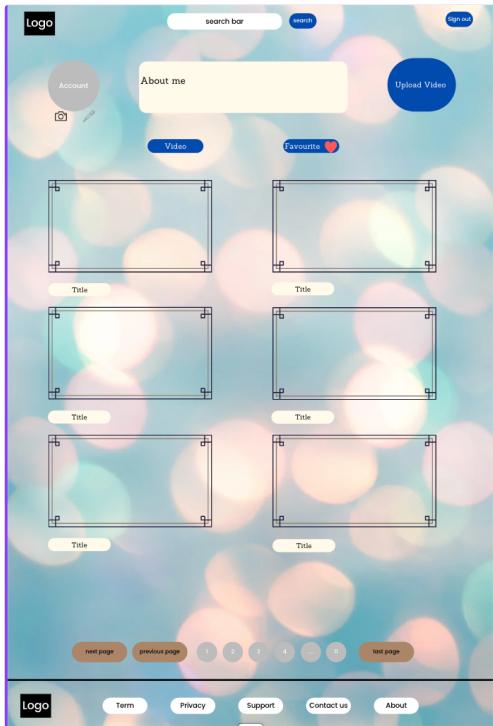
This is the page the user will access when clicking the picture . On this page, there is a big title on the top.

In the right bottom part, there is three hot keys, like love and dislike, the like will add the popularity however the dislike will reduce the popularity.

At bottom, the user can see the comments leaved by the other users. He/she can also click the like hotkey to support the other user's comment.

Also the user can leave his/her own comment by useing the bar at bottom, however he/she need to log first that the system can record his/her name. This can help the administer to punsih the people who is offensive .

## Account homepage



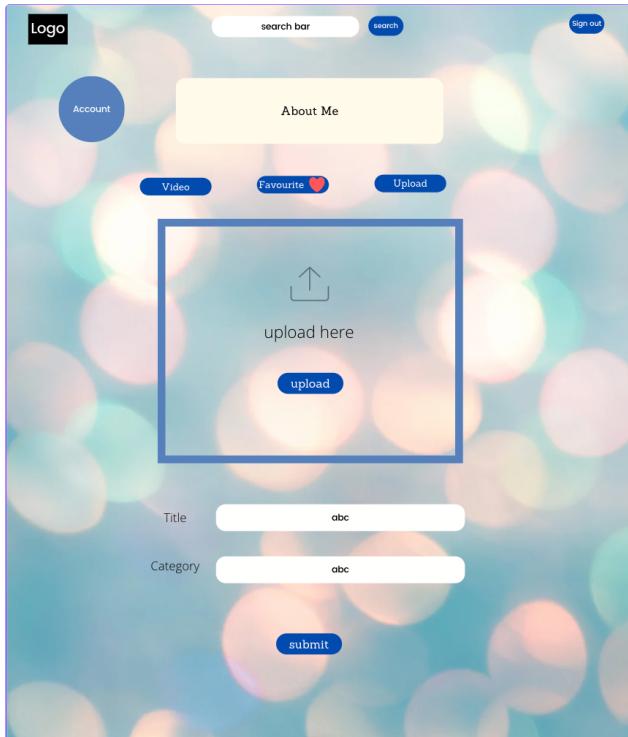
The purpose of the personal account homepage is to manage the videos and edit personal information. Firstly, we have the logo at the top-right of the page. The search bar here helps the users to find their posted videos or favorite videos. You can upload your selfie to your profile picture by clicking the camera pattern below the profile picture. Also, you can edit your personal information like gender, username, introduction, etc. by clicking the pencil pattern below the profile picture. Lastly, you can sign out from the account page.

As the client required, we adjust the size of the 'upload video' button to make it more obvious on the page. By clicking the 'video' button and 'favorite' button, we can access the posted videos and favorite videos.

When the users click the 'upload video' button, it will direct users to the [upload page](#) and users can post their videos on that page. Below is the pop-up window of after clicking the "edit-information" button. Users can change the details of their personal information.

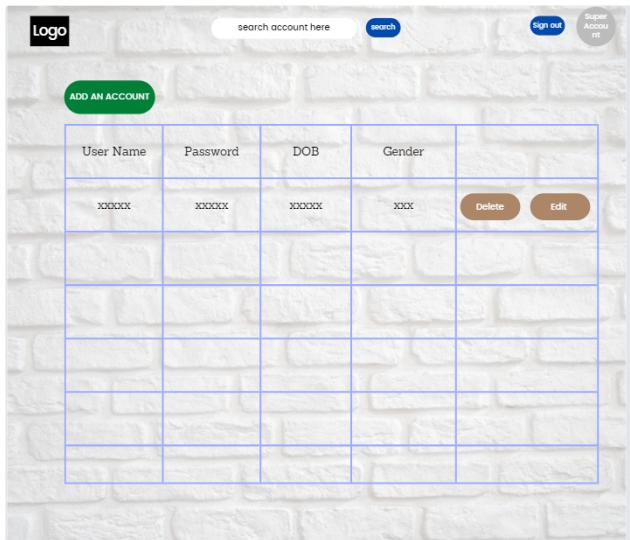
A modal dialog box titled 'Edit Information'. It contains four input fields: 'Username' with value 'hxx', 'Password' with masked value '\*\*\*\*\*', 'Date of birth' with value '29/03/2000', and 'Gender' with value 'Male'. Below these fields is a large text area labeled 'About Me'. At the bottom of the dialog are two buttons: 'Save' (blue) and 'Cancel' (blue).

## Upload Page



This is the upload page of our website. Users can post their videos on this page. Users also need to specify the title and category of the video. By clicking the 'upload' button, users can upload the video to the page. On top of that, users need to submit the video, title, and category together to send these data to the backend.

## Super Account Page (Administrator Page)



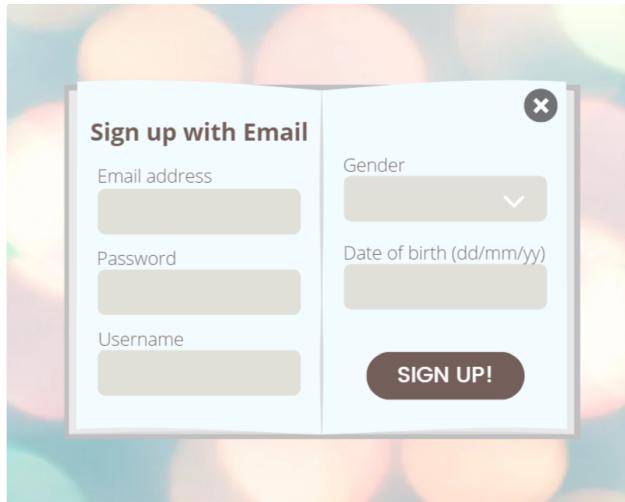
This page is Super Account Page (Administrator Page). There is only one administrator for this website. After the administrator log in, she/he can see the page. There is a search bar at the top of the page which is used for search a normal account.

There is a 'add an account' bar in the page, so administrator could add a normal user account. If a new user could not sign up an account successfully, he/she can contact Administrator, Administrator could help he/she create an new account.

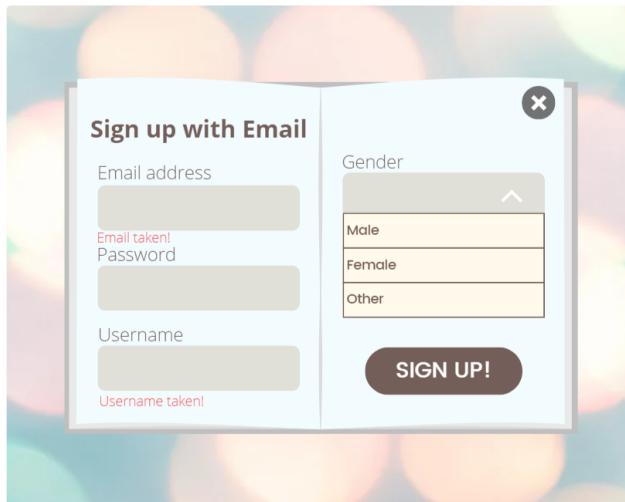
There is a table following the 'add an account' bar, which shows all normal user. This table contains information about users including User Name, Password, DOB (data of birth), Gender. The administrator could edit a user's information. For example, if a user meets some problems which could not edit their profile (user name, password), the administrator could help she/he to edit this. The administrator can decide to delete a user if that user has done something inappropriate

## Account sign in/up

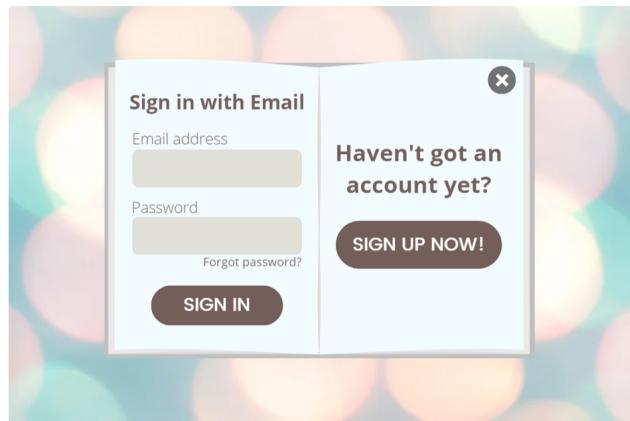
Users need to sign up an account in order to like, comment, and upload.



Information we collect for new account is email address, password, username, gender, and date of birth. Both email address and username has to be unique. Gender could be selected from male/female/other, and date of birth needs to be in dd/mm/yy format.



Once an account is created, users can use email address and password to log in. A forget password link is also provided for those who cannot remember their password.



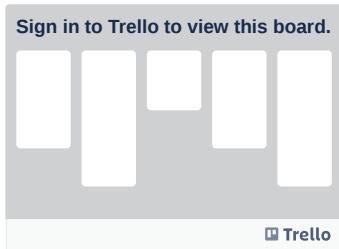
# Requirements

| As a <Role>   | I want to <Do something>      | So that<br><Achieve some goals>   | Priority |
|---------------|-------------------------------|---|----------|
| User          | Sign up                       | They has an account for this website. Normally, if the user first to access the website, they needs to signup. If the user does not sign in, they can only browser the website and could not do any actions.  | MEDIUM   |
| User          | Sign in                       | They can have the access to upload videos, and like/comment below videos. If the user does not sign in, they can only browser the website and could not do any actions.   | HIGH     |
| User          | Sign out                      | They can sign out the website if the user does not want to log in this website anymore.   | HIGH     |
| User          | Search key words              | They can find specific videos or users. The user can type key words in the search bar and web will return the list of relation videos.  | MEDIUM   |
| User          | Publish videos                | They can publish whatever they like. If the user want to upload a video, they can click upload video bar and add title and description. After publish the video, the user and others could see and comment the video.   | HIGH     |
| User          | Have profile and edit profile | They can edit their personal information. The information includes username, email. And user can choose whether to add description or not.  | MEDIUM   |
| User          | Have favourite file           | They can store their favourite videos. If the user like a video, they can store video in a store list.  | LOW      |
| Administrator | add a normal user account     | If a new user could not sign up an account successfully, they can contact Administrator, Administrator could help them create a new account.  | HIGH     |
| Administrator | search a normal account       | can access a normal user quickly. The administrator can search a user through username in search bar  | HIGH     |
| Administrator | change a user's profile       | If a user meets some problems which could not edit their profile (user name, password), the Administrator could help them to edit this.   | HIGH     |
| Administrator | delete a user                 | The administrator can decide whether to delete a user. The administrator has responsibility to detect whether a user has appropriate comment. If the administrator thinks a user has unappropriated actions, the administrator can decide to delete the user directly. So all information about this user will not delete from database and other user will not see the user anymore. | HIGH     |

|                    |  |   |        |
|--------------------|--|---|--------|
| Software developer | it is easy to maintain and update                                  | it is easy to maintain and update. Software developers should easy fix problems and bugs if an emergency event occurs. For example, if the client want to add a new function, the developer can quickly decide which part to add or change. | MEDIUM |
| User               | publish comment  | They can interact with other user.  | LOW    |
| User               | like a video or dislike a video                                    | They can help to adjust the video's popularity.   | MEDIUM |
| User               | upload and publish video   | The user can upload a video and publish it.   | HIGH   |
| User               | edit the profile details   | They can easily update their information such as username and other description about themselves.   | MEDIUM |
| User               | Delete/edit a video  | They can delete/edit videos that they no longer wish to share and re-upload a new version   | LOW    |
| User               | upload cover photo for video                                       | They can upload a photo as the cover to promote their videos, rather than system snipping one random scene from the video   | LOW    |
| Admin              | Have all users not listed in one page but in several pages instead | Don't need to view all users in one page  | LOW    |

## Development

Please refer to our trello page for the detailed plan.

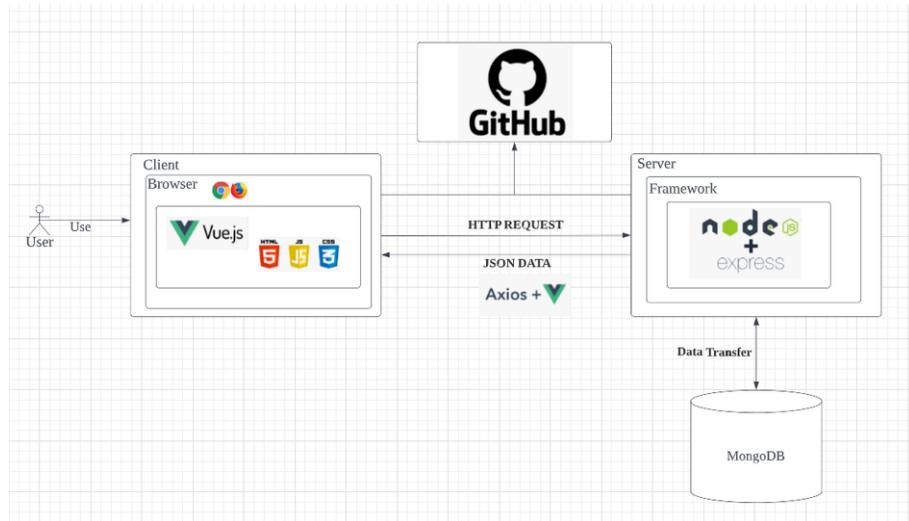


## Architecture

This is the part where you can clearly see the website architecture, UML as well as motivational model.

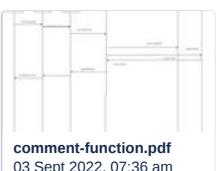
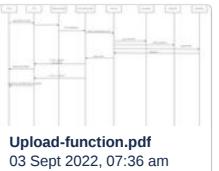
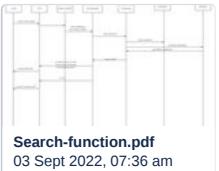
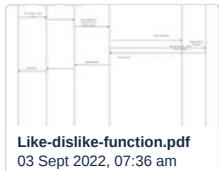
## Architecture Diagram

Below is the draft architecture diagram we have already developed and it may be modified later since we still have an issue on how to store the media file like mp4. We have not deployed the server on the cloud platform and we would run our website on local machine first.



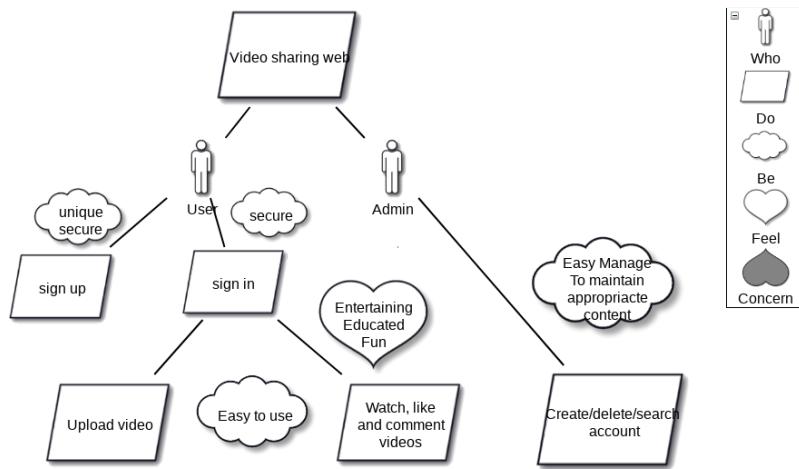
## UML Diagram

Below are the sequence diagrams we developed, they include all the main functions we needed. If we may miss some functions, we will add more diagrams later.



## Motivational Model (MM)

Below is Motivational Model (MM) diagram: Focusing on emotions, decision thinking and function of website.



There are two main types of users in the video sharing network. One is a normal user and the other is an administrator.

The normal user can signup, sign in. Sign-up and sign-in should be secure.

Once logged in, users can upload, watch, and commit videos. For this part, the user should be easy to use.

For the administrator, the main responsibility is to be able to create, delete, and search accounts. For this part, it should be easy to use and easy to manage.

## Team Roadmap

## Assessment

This page contains a list of assessments we need to turn in on canvas progressly including:

- Week 6 progress checklist
- Week 9 progress checklist
- Week 12 progress checklist
- Team presentation
- Final report

# Week 6 progress

## COMP30022 Progress Checklist

### Team:

Team Number: 36

Team Name: Vegetable Birds

### Reporting period:

Week 1- 6

### Process:

#### Team structure

| Name          | Role                | Responsibility                   |
|---------------|---------------------|----------------------------------|
| Xuxiao He     | Front-end Developer | Front-end, testing, UI Design    |
| Qi Guo        | Front-end Developer | Front-end, testing, UI Design    |
| CongCong Zhao | Front-end Manager   | Front-end, testing, UI Design    |
| Jialin Ding   | Back-end Manager    | Back-end, Documentation, Planner |
| Liping Meng   | Back-end Developer  | Back-end, Documentation, Planner |

Please refer to our team overview via this link:

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/229457/Team+Overview>

Please refer to our project description via this link:

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/262198/VShare+project+description+-+5+Vegetable+Birds>

### **Team contribution:**

Xuxiao He: Designed, drew, and wrote requirements for account homepage, upload page, UML diagrams, and documentation. Finished partial code for homepage.

Qi Guo: Designed, drew, and wrote requirements for video page, sorting method, UML diagrams, and documentation. Finished partial code for sign in/up page.

Congcong Zhao: Designed, drew, and wrote requirements for homepage, sorting method, UML diagrams, and documentation. Finished partial code for video page.

Jialin Ding: Recorded supervisor meetings and client meetings. Designed, drew, and wrote requirements for sign in/up, motivation model, and documentation. Finished connecting server to database.

Liping Meng: Recorded team meetings and client meetings. Designed, drew, and wrote requirements for admin account, motivation model, and documentation. Finished coding part for sign in/up connection.

#### **Team internal communication:**

Team internal communication mainly via WeChat.

<https://it-project-team36.atlassian.net/l/cp/zGEEkcAT>

#### **Team decision-making:**

For all decision-making reasons, please refer to the link after each argument

- Coding:
  - For the front-end, we decided to use HTML, CSS, JavaScript, Vue, Axios, and Bootstrap. <https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/262198/VShare+project+description+-+5+Vegetable+Birds#Decisions-on-Front-end>
  - For the back-end, we decided to use Node.js, Express.js, and MongoDB. <https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/262198/VShare+project+description+-+5+Vegetable+Birds#Decisions-on-Back-end>
- Other general tools:
  - Trello - task management
  - Confluence - documentation
  - Canva - design
  - Github - repository
  - Lucidchart - diagrams

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/262198/VShare+project+description+-+5+Vegetable+Birds#Decisions-on-Other-Tools>

Our Trello page at 3/09/22 10:50 PM

### Communication with client

Please refer to following meeting records:

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/294929>

### Team communications

Please refer to following meeting records:

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/65627>

#### **Tools used**

Confluence, Canva, Lucidchart, VScode, GitHub, and Trello.

Please refer to our project description page for decisions on above tools:

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/262198>

### **Artefacts**

#### **Requirements**

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/65648>

#### **Design**

UI design includes Home Page, Sorting Method, Video Page, Account Homepage, Upload Page, Super account Page, Account sign in/up.

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/2424879>

Architecture includes Architecture Diagram, UML Diagram , Motivational Model.

<https://it-project-team36.atlassian.net/l/cp/0hkmtbyY>

#### **Coding**

GitHub link: <https://github.com/Kisupan/IT-PROJECT.git>

#### **Testing**

Not started yet.

#### **Deployment**

Not started yet.

# Week 9 Progress

## Team:

Team Number: 36

Team Name: Vegetable Birds

## Reporting period:

Week 7-9

## Process:

### Team structure

| Name          | Role                | Responsibility   |
|---------------|---------------------|--|
| Xuxiao He     | Front-end Developer | Front-end, testing, UI Design, mainly for homepage, profile page and upload page               |
| Qi Guo        | Front-end Developer | Front-end, testing, UI Design , mainly for the log and sign up page and category page          |
| CongCong Zhao | Front-end Manager   | Front-end, testing, UI Design, mainly for video part and administrator account management part |
| Jialin Ding   | Back-end Manager    | Back-end, mainly for video part, Documentation, testing, Planner                               |
| Liping Meng   | Back-end Developer  | Back-end, mainly for user part Documentation, testing, Planner                                 |

### Adherence to agile ceremonies

Please refer to our team overview via this link:

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/229457/Team+Overview>

Please refer to our **project description** including goals and stages via this link:

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/262198/VShare+project+description+-+5+Vegetable+Birds>

Please refer to our weekly achievement for a more specific goal setting & achieving

 [Weekly achievement](#)

### Team contribution:

- Xuxiao He:

- testing and writing test cases
  - finished coding for changing the information on the profile page
  - successfully reset the password
  - restyle the "manage profile function" to the form of the pop-up window
  - coding for the upload page without function.
- Qi Guo:
  - Designed, drew, and wrote requirements for video pages, sorting methods, UML diagrams, and documentation. Finished partial code for sign-in/up page.
  - add pop-window for the front part
  - now working the vue-component
  - restyle the css
  - currently working on the search vedio part
  - currently working on getting video's discription from the backend
- Congcong Zhao
  - Contact the client and discuss the meeting time.
  - Updated weekly achievement on frontend part.
  - Updated weekly events we should do on trello.
  - Updated requirements, front end decision making.
  - write unit function test file sign\_in\_up\_test.html and administrator\_test.html and successfully get the data from the backend.
  - Finished basic functions on VideoPlayPage.
  - Adding pop-window to AdministratorPage when editing the account.
  - Currently working on adding serach bar for searching the corresponding account.
  - Currently working on combining current codes with the Vue component.
- Jialin Ding
  - Recorded all types of meetings.
  - Updated requirements, team decisions, and UI design for sign-in/up.
  - Finished coding for like/dislike a video, search for users/videos.
  - Currently working on sign-up and search user testing.
- Liping Meng:
  - Recorded all types of meetings.
  - Finished coding for show all user(admin page), delete a user(admin page), reset password, edit&update personal information, and storing video on local memory
  - Currently working on sign in and rest password testing

#### **Team internal communication:**

Team internal communication mainly via WeChat.

[✉ Updates till 2/09](#)

[✉ Updates till 28/09](#)

#### **Team decision-making:**

For all decision-making reasons, please refer to the link after each argument

- Coding:
  - For the front end, we decided to use HTML, CSS, JavaScript, Vue, Axios, and Bootstrap. [✉ VShare project description - 5 Vegetabl e Birds | Decisions on Front end](#)

- For the back end, we decided to use Node.js, Express.js, and MongoDB.

- General tools:

- Trello - task management
- Confluence - documentation
- Canva - design
- GitHub - repository
- Lucidchart - diagrams

VShare project description - 5 Vegetable Birds | Decisions on Other Tools

- Other obstacles and their decisions during the project

Team decisions

week 7

The screenshot shows a Trello board titled "Week 7 7/09 - 13/09". The board has several lists:

- Week 7 workshop**: Tasks include "Frontend coding" (Sep 7 - Sep 14), "API connection" (Sep 7 - Sep 14), "Delete account" (Sep 7 - Sep 14), and "Like/dislike" (Sep 7 - Sep 14). Each task has assigned members (CZ, JD, LM, QG, XH).
- Show all accounts**: Task (Sep 7 - Sep 14) assigned to LM.
- Reset password**: Task (Sep 7 - Sep 14) assigned to LM.
- Html coding for AdministratorPage**: Task (Sep 7 - Sep 14) assigned to CZ.
- Frontend Css self-learning**: Task (Sep 7 - Sep 14) assigned to CZ, QG, XH.
- Front-end coding for the profile page**: Task (Sep 7 - Sep 14) assigned to XH.

At the bottom right of the board, there is a "+ Add a card" button.

week 8

**Week 8 14/09 - 20/09**

- Week 8 workshop** (Due Sep 14) assigned to CZ, JD, LM, QG, XH.
- Video testing** (Due Sep 7 - Sep 21) assigned to CZ, JD, LM, QG, XH.
- Search single user/video** (Due Sep 14 - Sep 21) assigned to JP.
- Edit and update personal information** (Due Sep 14 - Sep 21) assigned to LM.
- Html coding for VideoPage adding style sheet as well** (Due Sep 14 - Sep 21) assigned to CZ.
- Adding functions to the AdministratorPage and VideoPage using Vue framework** (Due Sep 14 - Sep 21) assigned to CZ.

**Make connections with the backend for sending and getting data. Unit testing for sign in/sign up page and administratorPage which includes adding, deleting and update accounts.** (Due Sep 14 - Sep 21) assigned to CZ.

- Frontend Vue self-learning** (Due Sep 14 - Oct 3) assigned to CZ, QG, XH.
- Front-end coding for upload video page** (Due Sep 14 - Sep 21) assigned to XH.
- Add functions to the profile page using Vue** (Due Sep 14 - Sep 21) assigned to XH.
- re-style the administrator page and adding pop-up window when editing the account** (Due Sep 14 - Sep 21) assigned to CZ.

## week 9

**Week 9 21/09 - 5/10**

- Week 9 workshop** (Due Sep 21) assigned to CZ, JD, LM, QG, XH.
- Client meeting @ 9 PM** (Due Sep 21 - Sep 21) assigned to CZ, JD, LM, QG, XH.
- Backend coding includes improvements to upload videos.** (Due Sep 14 - Oct 5) assigned to JD, LM.
- Frontend Vue Component self-learning** (Due Sep 14 - Sep 28) assigned to CZ, QG, XH.
- Improve code maintenance and combine with Vue component on AdministratorPage and VideoPage** (Due Sep 14 - Oct 3) assigned to CZ.
- testing and write test cases** (Due Sep 14 - Sep 28) assigned to XH.
- Connect to the server and successfully change the information such as password of the account** (Due Sep 14 - Sep 28) assigned to XH.

**re-style the mange-profile function to the pop-up window** (Due Sep 14 - Sep 28) assigned to XH.

**Documentaion on the frontend weekly achievement as well as frontend decision making** (Due Sep 14 - Sep 28) assigned to CZ, QG, XH.

- Sign in testing** (Due Sep 25 - Oct 1) assigned to LM.
- Reset password testing** (Due Sep 25 - Oct 1) assigned to LM.
- Sign up testing** (Due Sep 25 - Oct 1) assigned to JD.
- Search user testing** (Due Sep 25 - Oct 1) assigned to JD.
- try to use the vue component to make the code more clear and easier to maintance** (Due Sep 28) assigned to QG.

Our Trello page at 28/09/22 11 PM

## Communication with client

Please refer to the following meeting records:

Client Meetings

## **Team communications**

Please refer to the following meeting records:

 [Team Meetings](#)

## **Tools used**

Confluence, Canva, Lucidchart, VScode, GitHub, and Trello.

Please refer to our project description page for decisions on the above tools:

 [VShare project description - 5 Vegetable Birds](#)

## **Artefacts**

### **Requirements**

 [Requirements](#)

Updated client's new requirements from 21.09 meeting

### **Design**

#### **UI design**

Include Home Page, Sorting Method, Video Page, Account Homepage, Upload Page, Super account Page, and Account sign-in/up.

Updated latest UI design from last reporting period.

 [UI Design](#)

#### **Architectural design**

Please refer to the architecture which includes Architecture Diagram, UML Diagram, Motivational Model.

 [Architecture](#)

## **Coding**

GitHub link:

 [GitHub - Kisupan/IT-PROJECT](#)

## **Testing**

Please refer to our testing folder for testing timeline and current testing in process.

 [Testing](#)

## **Deployment**

Not started yet.

## **Maintenance activities**

- Back end
  - Fixed problem "password-reset can't word correctly."
    - The original version showed that the change was successful after resetting the password, but when logging in, the old password was still required. So we modified the password-reset function. So we modified the function.
  - Optimised searching method for "search user"

- First draft searching for “search user” depends on manually feeding the server a keyword again. Now it will automatically take the key as input.
- Front end
  - Fixed problem “password-reset cannot work correctly.”
    - The original version showed that the change was successful after resetting the password, but when logging in, the old password was still required. So we contact the backend members and fix the code.
  - Optimised current codes using Vue component.
    - Our current codes are pretty messy since we include all things like .js .css and Vue instance in one html page so that it is difficult to maintain and modify. We are trying to use Vue component to make code modular which means each component is responsible for one thing and it can be reused for the same logic.

# Week 12 Process Checklist

## Team:

Team Number: 36

Team Name: Vegetable Birds

## Reporting period:

Week 10-12

## Process:

### Team structure

| Name          | Role                | Responsibility  |
|---------------|---------------------|---|
| Xuxiao He     | Front-end Developer | Front-end, testing, UI Design, mainly for homepage, profile page and upload page              |
| Qi Guo        | Front-end Developer | Front-end, testing, UI Design , mainly for the log and sign up page and category page         |
| Congcong Zhao | Front-end Manager   | Front-end, testing, UI Design,mainly for video part and administrator account management part |
| Jialin Ding   | Back-end Manager    | Back-end, mainly for video part, Documentation, testing, Planner                              |
| Liping Meng   | Back-end Developer  | Back-end, mainly for user part<br>Documentation, testing, Planner                             |

### Adherence to agile ceremonies

Please refer to our team overview via this link:

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/229457/Team+Overview>

Please refer to our **project description** including goals and stages via this link:

<https://it-project-team36.atlassian.net/wiki/spaces/DOCUMENTAT/pages/262198/VShare+project+description+-+5+Vegetable+Birds>

Please refer to our weekly achievement for a more specific goal setting & achieving

 Weekly achievement

#### **Team contribution:**

- Xuxiao He:
  - testing and writing test cases
  - finish converting all my previous code to Vue components for better maintenance and modularization
  - code router for Vue components
  - integration for front-end development
  - improve my code and fix bugs
  - finish all my responsible functions (eg. upload, and delete videos ...)
  - fix the video path problem so that video pages can get the correct path
- Qi Guo:
  - write the search page with sorting by like number and the paging function with the easy and detail forms.
  - add the style to the footer and add it in the main component so that it can stay in the bottom of each page
  - add description and picture of the video in the category page
  - fix some refresh bug and improve the code
  - linked my page with the video page written by congcong(give his page the combined path and the video name)
  - add the help page and team page in footer and put the router in the router index
- Congcong Zhao
  - Contact the client and discuss the meeting time for product demonstration
  - Updated weekly achievement on frontend part, partial weekly events we should do on trello.
  - Updated front end decision making.
  - Integration for front-end development by combining my Vue components with Xuxiao's Vue components
  - improve my code and fix bugs (user now can like or dislike videos, search bar in administrator system, when delete user delete all information about this deleted user, refresh problem)
  - finish all my responsible functions (eg. administrator system, video play and sign in/up/out system)
- Jialin Ding
  - Recorded all types of meetings.
  - Optimised code
  - Finished testing for: sign up, search user
  - Currently working on deployment
- Liping Meng:
  - Recorded all types of meetings.
  - Finished coding for show all user(admin page), delete a user(admin page), reset password, edit&update personal information, and storing video on local memory
  - Fix password reset bug
  - Finished Video streaming part and delete video function
  - Finished sign in testing

#### **Team internal communication:**

Team internal communication mainly via WeChat.

 [Updates till 2/09](#)

 [Updates till 28/09](#)

 [update until 22/10](#)

#### **Team decision-making:**

For all decision-making reasons, please refer to the link after each argument

- Coding:
  - For the front end, we decided to use HTML, CSS, JavaScript, Vue, Axios, and Bootstrap. [VShare project description - 5 Vegetable Birds | Decisions on Front end](#)
  - For the back end, we decided to use Node.js, Express.js, and MongoDB. the
- General tools:
  - Trello - task management
  - Confluence - documentation
  - Canva - design
  - GitHub - repository
  - Lucidchart - diagrams

#### [VShare project description - 5 Vegetable Birds | Decisions on Other Tools](#)

- Other obstacles and their decisions during the project

#### [Team decisions](#)

### week 10

Week 10 5/10 - 11/10

Week 10 workshop  
Oct 5  
CZ JD LM QG XH

Backend coding: username in video, update personal info (age & gender)  
Oct 5 - Oct 12  
JD LM

Deploy - Heroku - self learn  
Oct 5 - Oct 12  
CZ JD LM QG XH

Backend testing - refer to confluence plan

Week 10 5/10 - 11/10

Backend testing - refer to confluence plan  
Oct 5 - Oct 11  
JD LM

Frontend coding - covert all current codes to Vue components and integrate all current completed functions  
Oct 5 - Oct 11  
CZ QG XH

Documentation on the frontend weekly achievement as well as frontend descision making  
Oct 5 - Oct 11  
CZ QG XH

+ Add a card

### week 11

**Week 11 12/10 - 18/10**

Week 11 workshop  
Bell icon User icon Oct 12  
CZ JD LM QG XH

Backend code modification - modify current update user api and delete user api.  
Bell icon User icon Oct 12 - Oct 18  
JD LM

Backend testing - refer to confluence plan  
User icon Oct 12 - Oct 18  
JD LM

**Week 11 12/10 - 18/10**

User icon Oct 12 - Oct 18  
JD LM

Deploy - Heroku - first attempt  
User icon Oct 12 - Oct 18  
JD

Presentation slides  
User icon Oct 14 - Oct 17  
CZ JD LM QG XH

Add extra serach bar to the administrator system(now it can both serach username and email)  
User icon Oct 12 - Oct 18  
CZ

Add like and dislike functions (now)  
+ Add a card

**Week 11 12/10 - 18/10**

Add like and dislike functions (now users can like and dislike videos) and description so that user can see the description of the video  
User icon Oct 12 - Oct 18  
CZ

Improve management profile (now it can track all uploaded videos) and manage account information (now it can directly update user's information in account profile)  
User icon Oct 12 - Oct 18  
XH

solve video path resolution (now video can play and upload)  
User icon Oct 12 - Oct 18  
XH

+ Add a card

whole like page refreshing and css styles on different size of screen  
User icon Oct 12 - Oct 18  
CZ QG XH

Documentation on the frontend weekly achievement as well as frontend descision making  
User icon Oct 12 - Oct 18  
CZ QG XH

Add video sorting function (user can serach videos by like and dislike number)  
User icon Oct 12 - Oct 18  
QG

Add video preview picture for each video and video paging (now all videos would be sperated into several pages)  
User icon Oct 12 - Oct 18  
QG

Add footer content like about us, policy and terms  
User icon Oct 12 - Oct 18  
QG

## week 12

**Week 12 19/10 - 24/10**

Week 12 workshop - presentation  
User icon Oct 19  
CZ JD LM QG XH

Client meeting - Final  
Bell icon User icon Oct 20 - Oct 20  
CZ JD LM QG XH

Week 12 progress checklist  
Bell icon User icon Oct 19 - Oct 23  
CZ JD LM QG XH

+ Add a card

**After week 12 25/10 - 10/11**

Challenge - Deploy

Product maintenance

+ Add a card

## Communication with client

Please refer to the following meeting records:

[Client Meetings](#)

## Team communications

Please refer to the following meeting records:

[Team Meetings](#)

## Tools used

Confluence, Canva, Lucidchart, VScode, GitHub, and Trello.

Please refer to our project description page for decisions on the above tools:

[VShare project description - 5 Vegetable Birds](#)

## Artefacts

### Requirements

[Requirements](#)

Updated client's new requirements from 21.09 meeting

### Design

#### UI design

Include Home Page, Sorting Method, Video Page, Account Homepage, Upload Page, Super account Page, and Account sign-in/up.

Updated latest UI design from last reporting period.

[UI Design](#)

### Architectural design

Please refer to the architecture which includes Architecture Diagram, UML Diagram, Motivational Model.

[Architecture](#)

### Coding

GitHub link:

[GitHub - Kisupan/IT-PROJECT](#)

### Testing

Please refer to our testing folder for testing timeline and current testing in process.

[Testing](#)

### Deployment

May deploy after week 12 since it is optional and client has no requirement for deployment

Please refer to our deployment plan for more details.

[Deployment](#)

## Maintenance activities

- Back end
  - Fixed problem “password-reset can't word correctly.”
    - The original version showed that the change was successful after resetting the password, but when logging in, the old password was still required. So we modified the password-reset function. So we modified the function.
  - Optimised searching method for “search user”
    - First draft searching for “search user” depends on manually feeding the server a keyword again. Now it will automatically take the key as input.
  - Optimize user information (add more user information)
    - before update, user only have username, password, email. After update, user have username, password, email, gender, age
  - Optimize video uploading (add category of video)
    - in order to have more detailed video classification, add video category. When user upload video, user could choose video type such as sports, food.
  - Optimize database (add username in the video database)
    - We found the owner\_id in video database could not show who upload video correctly. So we add username when user upload video to connect user database and video database.
- Front end
  - Optimised css styles based on current styles
    - The original version of the css style is pretty simple and we use add more css styles in bootstrap and elemnt ui to the website to make it more beautiful.
  - Slightly modified upload video interface
    - When users want to upload the video, they also need to select video category as well as adding video description.
  - Filling more user information when sign up a new account
    - Now a valid account should be signed up when includes username, email, age, gender, password

## Team decisions

This page will be updated with latest decisions made by the team relative to project content.

# Testing

This page will contain all testing information including tested function, methods, test cases, tested time, and instructions.

F - testing on frontend function performance

B - testing on backend function performance

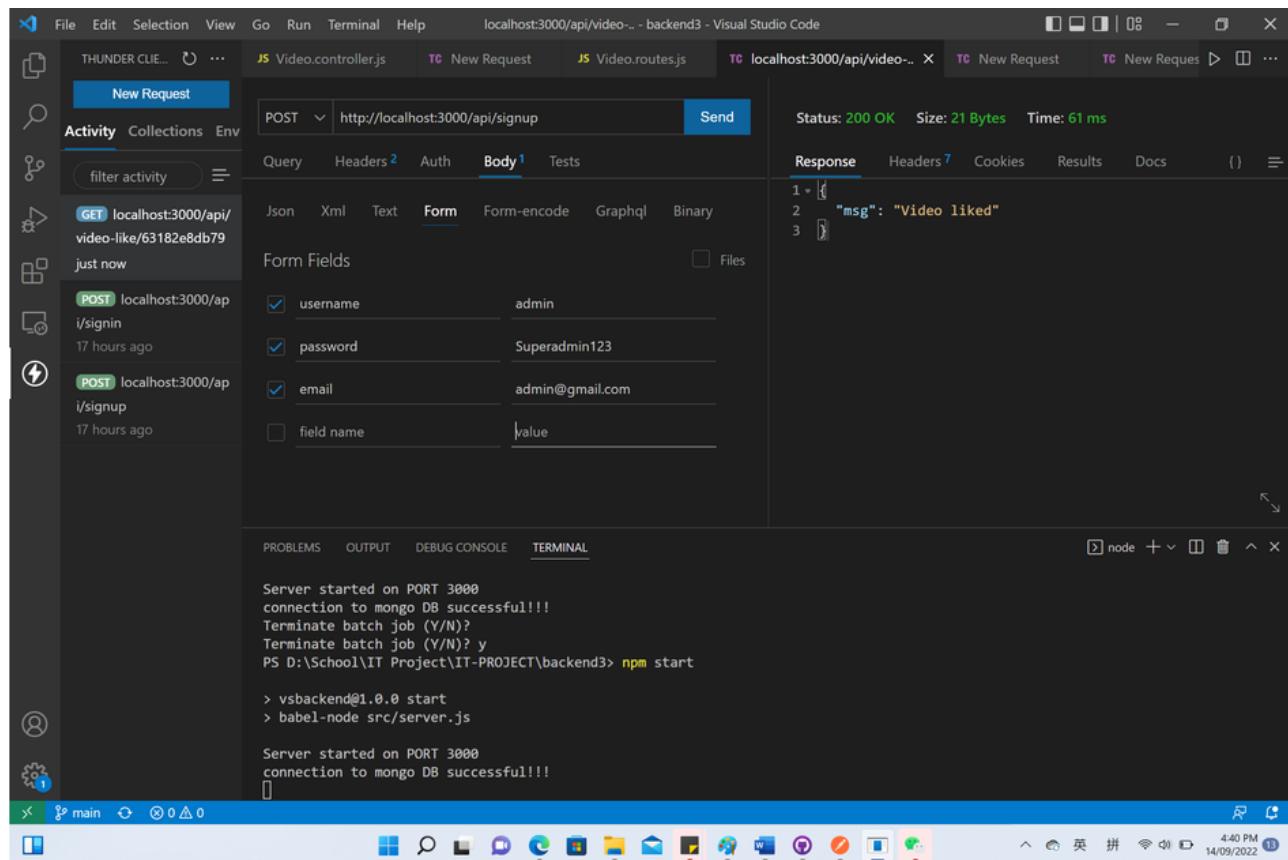
C - combined testing which involve both ends

---

## Testing timeline

- Backend
  - Sign in - 25/09 - 1/09
  - Sign up - 25/09 - 1/09
  - Reset password - 25/09 - 1/10
  - Search user - 25/09 - 1/10
  - Show all users - 2/10 - 8/10
  - Delete user - 2/10 - 8/10
  - Upload video - 2/10 - 8/10
  - Like/dislike video - 2/10 - 8/10
  - Search video - 2/10 - 8/10
- Frontend
  - Sign in - 25/09 - 1/09
  - Sign up - 25/09 - 1/09
  - Reset password - 25/09 - 1/10
  - Add new user from administrator page - 25/09 - 1/10
  - Delete user from administrator page- 2/10 - 8/10
  - Upload video - 8/10 - 15/10
  - Like/dislike video - 8/10 - 15/10
  - Search video - 8/10 - 15/10
  - sort video- 8/10 - 15/10

**Admin login details PLEASE KEEP**



## Testing format

C/F/B - Function xxx test

This page will contain testing method and test cases for xxxxxxx function

|                 |                                    |
|-----------------|------------------------------------|
| Test type       |                                    |
| Execution type  |                                    |
| Objective       |                                    |
| Setup           |                                    |
| Pre-conditions  |                                    |
| Notes           | 1. xxx<br>2. xxx<br>*asdjhjkds     |
| Time constraint | Minimum: ?? min<br>Maximum: ?? min |

| No. | Test case    | Test steps       | Test data | Expected result | Actual result |
|-----|--------------|------------------|-----------|-----------------|---------------|
| 1   | Verify xxxxx | 1. xxx<br>2. xxx | xxx = xxx |                 |               |



## B (back end) - General instruction

Listed below are all testing steps for different functions

- To run backend server:

```
npm start
```

- To test via thunder client:

--> Signup

POST --> <http://localhost:3000/api/signup>

Body --> Form --> fill in username, email, password

--> Signin

POST --> <http://localhost:3000/api/signin>

Body --> Form --> fill in account(username or email), password

--> password reset

POST --> <http://localhost:3000/api/password-reset>

Body --> Form --> fill in email, new password

--> Delete account

POST --> <http://localhost:3000/api/delete>

Body --> Form --> fill in email

--> Find all accounts

GET --> <http://localhost:3000/api/findall>

--> Like

GET --> <http://localhost:3000/api/video-like/>

Body --> Form --> fill in id (video\_id), operation (boolean value default is false, if false like -1, if true like +1)

--> Dislike

GET --> <http://localhost:3000/api/video-dislike/>

Body --> Form --> fill in id (video\_id), operation (boolean value default is false, if false dislike -1, if true dislike +1)

→ Search user in admin mode

GET → <http://localhost:3000/api/user-search>

Body --> Form --> fill in username, isadmin (boolean value default is false, admin mode fill in true)

--> Search all videos

GET --> <http://localhost:3000/api/findallV>

--> Search videos by category

GET --> <http://localhost:3000/api/cat-search/>

Body --> Form --> fill in key (category)

--> Search user's all uploaded videos

GET --> <http://localhost:3000/api/user-video-search/>

Body --> Form --> fill in key (username)

--> Delete user's all uploaded videos in admin mode

Delete --> <http://localhost:3000/api/user-video-delete/>

Body --> Form --> fill in username, video\_list (contains all videos path)

--> Delete user's upload videos

Delete --> <http://localhost:3000/api/video/>

Body --> Form --> fill in video\_id, video\_path

--> Upload the video

GET --> <http://localhost:3000/api/video>

Headers --> x-auth-token: token (given when login in)

Content-type --> multipart/form-data

Body --> Form --> fill in username, category, name, video (file), description

## B - Sign up testing

This page contain testing methods and test cases for sign up function

|                 |  |
|-----------------|--|
| Test type       |  |
| Execution type  | Manual   |
| Objective       | Verify if a user is successfully signed up   |
| Setup           | cd backend 3<br>npm start  |
| Pre-conditions  |  |
| Notes           | 1. Add user name<br>2. Add user email<br>3. Add password<br>4. Select gender<br>5. Add age<br>6. Verify if the user is signed up<br><br>*All fields are mandatory<br><br>*Username and email need to be unique.<br><br>*Maximum allowed number of chars(username&password:20, email: 40)<br><br>*Only numbers between 0-150 is allowed for age fields<br><br>*Special character @ and . must be included in email field, besides alphabets and numbers, special characters _ is allowed in email address<br><br>*No special characters allowed for username, alphabets and numbers only. |
| Time constraint | Minimum: 10 min<br><br>Maximum: 15 min   |

| No. | Test case      | Test steps  | Test data   | Expected result                                      | Actual result |
|-----|----------------|---|---|--|---------------|
| 1   | Verify sign up | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = admin<br><br>email = <a href="mailto:admin@gmail.com">admin@gmail.com</a><br><br>password =<br>Superadmin123<br><br>gender = Prefer not to say<br><br>age = 30 | Status: 201<br><br>MSG: Account created successfully | Passed        |
| 2   | Verify sign up | 1. Go to sign up page<br>2. Enter email   | username = admin  | Status: 501  | Passed        |

|   |   |   |   |                                  |        |
|---|---|---|---|----------------------------------|--------|
|   | duplicate username                        | 3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up  | email = <a href="mailto:123@gmail.com">123@gmail.com</a><br>password = Sjh123<br>gender = Prefer not to say<br>age = 30                                     | MSG: format error                |        |
| 3 | Verify sign up<br>duplicate email address | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = admin2<br>email = <a href="mailto:admin@gmail.com">admin@gmail.com</a><br>password = Admin222<br>gender = Prefer not to say<br>age = 30          | Status: 501<br>MSG: format error | Passed |
| 4 | Verify sign up<br>missing email           | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = admin<br>email =<br>password = Sjh123<br>gender = Prefer not to say<br>age = 30  | Status: 501<br>MSG: format error | Passed |
| 5 | Verify sign up<br>missing username        | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username =<br>email = <a href="mailto:321@gmail.com">321@gmail.com</a><br>password = Sjh123<br>gender = Prefer not to say<br>age = 30                       | Status: 501<br>MSG: format error | Passed |
| 6 | Verify sign up<br>invalid email address   | Go to sign up page<br>1. Enter email<br>2. Enter username<br>3. Enter password<br>4. Select gender<br>5. Enter age<br>6. Click sign up    | username = halfemail<br>email = <a href="mailto:invalidemail@gmail">invalidemail@gmail</a><br>password = invalidE<br>gender = Prefer not to say<br>age = 30 | Status: 501<br>MSG: format error |        |
| 7 | Verify sign up<br>invalid email address   | Go to sign up page<br>1. Enter email<br>2. Enter username<br>3. Enter password<br>4. Select gender<br>5. Enter age<br>6. Click sign up    | username = missingdomain<br>email = <a href="mailto:invalidemail">invalidemail</a><br>password = invalidE<br>gender = Prefer not to say<br>age = 30         | Status: 501<br>MSG: format error |        |
| 8 | Verify sign up<br>invalid email address   | Go to sign up page<br>1. Enter email<br>2. Enter username<br>3. Enter password  | username = emailcharacters  | Status: 501<br>MSG: format error | Passed |

|    |                                     |   |   |   |        |
|----|-------------------------------------|---|---|---|--------|
|    |                                     | 4. Select gender<br>5. Enter age<br>6. Click sign up  | email = invalidemail!!!@gmail.com<br><br>password = invalidE<br><br>gender = Prefer not to say<br><br>age = 30  |   |        |
| 9  | Verify sign up<br>invalid username  | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = username.character<br><br>email = invaliduser@gmail.com<br><br>password = invalidU<br><br>gender = Prefer not to say<br><br>age = 30   | Status: 501<br><br>MSG: format error                          | Passed |
| 10 | Verify sign up<br>username too long | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = superlongusernameisnotallowed<br><br>email = longusername@gmail.com<br><br>password = longgU<br><br>gender = Prefer not to say<br><br>age = 30                                     | Status: 501<br><br>MSG: format error                          | Passed |
| 11 | Verify sign up<br>email too long    | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = longemail<br><br>email = this.is.a.really.long.email.address.contains.more.than.40.characters@gmail.com<br><br>password = longgE<br><br>gender = Prefer not to say<br><br>age = 30 | Status: 501<br><br>MSG: format error                          | Passed |
| 12 | Verify sign up<br>password too long | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = longpwd<br><br>email = longpwd@gmail.com<br><br>password = longpasswordlikethisisnotallowed<br><br>gender = Prefer not to say<br><br>age = 30                                      | Status: 503<br><br>MSG: Password length need to be 6-20 range | Passed |
| 13 | Verify sign up                      | 1. Go to sign up page<br>2. Enter email<br>3. Enter username  | username = nogender   | Status: 501<br><br>MSG: format error                          | Passed |

|    |                                   |   |  |  |        |
|----|-----------------------------------|---|--|--|--------|
|    | gender not selected               | 4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up   | email = nogender@gmail.com<br><br>password = hehehehehe<br><br>gender =<br><br>age = 30  |  |        |
| 14 | Verify sign up<br>age not entered | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = noage<br><br>email = noage@gmail.com<br><br>password = hahahahaha<br><br>gender = Prefer not to say<br><br>age =              | Status: 501<br><br>MSG: format error                 | Passed |
| 15 | Verify sign up<br>invalid age     | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = longlife<br><br>email = eternal.life@gmail.com<br><br>password = live4ever<br><br>gender = Prefer not to say<br><br>age = 300 | Status: 502<br><br>MSG: Invalid age                  | Passed |
| 16 | Verify sign up                    | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = user1<br><br>email = user1@gmail.com<br><br>password = Superuser1<br><br>gender = Prefer not to say<br><br>age = 30           | Status: 201<br><br>MSG: Account created successfully | Passed |
| 17 | Verify sign up                    | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = user2<br><br>email = user_2@gmail.com<br><br>password = Superuser2<br><br>gender = male<br><br>age = 30                       | Status: 201<br><br>MSG: Account created successfully | Passed |
| 18 | Verify sign up                    | 1. Go to sign up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Select gender<br>6. Enter age<br>7. Click sign up | username = User1<br><br>email = user.1@gmail.com<br><br>password = Superuser1<br><br>gender = male<br><br>age = 30                       | Status: 201<br><br>MSG: Account created successfully | Passed |
| 19 |                                   |   |  |  |        |

## B - Sign in testing

This page contain testing methods and test cases for sign in function

|                 |  |
|-----------------|--|
| Test type       | Functional   |
| Execution type  | Manual   |
| Objective       | Verify if a user is successfully signed in   |
| Setup           | cd backend 3<br>npm start  |
| Pre-conditions  |  |
| Notes           | 1. Add account (username or email)<br>2. Add password<br>3. Verify if the user is signed in<br>*All fields are mandatory<br>*Maximum allowed number of chars(username, password:20, email: 40)<br>*Only numbers between 0-120 is allowed for age fields<br>*Special character @ and . must be included in email field, besides alphabets and numbers, special characters - and are allowed in email address<br>*No special characters allowed for username and password, alphabets and numbers only. |
| Time constraint | Minimum: 10 min<br>Maximum: 15 min   |

| No. | Test case                      | Test steps   | Test data  | Expected result   | Actual result |
|-----|--------------------------------|--|--|---|---------------|
| 1   | Verify sign in                 | 1. Go to sign in page<br>2. Enter email<br>3. Enter password<br>4. Click sign up | account = <a href="mailto:admin@gmail.com">admin@gmail.com</a><br>password = Superadmin123 | Status: 200<br>MSG: sign in successfully                      |               |
| 2   | Verify sign in missing account | 1. Go to sign in page<br>2. Enter email<br>3. Enter password<br>4. Click sign up | account= <a href="mailto:Sjh123">Sjh123</a><br>password = Sjh123                           | Status: 404<br>MSG: "username of this account does not exist" |               |
| 7   | Verify sign in invalid account | 1. Go to sign in page<br>2. Enter email  | account = <a href="mailto:invalidemail">invalidemail</a><br>password = invalidE            | Status: 404   |               |

|    |                                  |  |  |  |  |
|----|----------------------------------|--|--|--|--|
|    |                                  | 3. Enter password<br>4. Click sign up  |  | MSG: "username of this account does not exist" |  |
| 10 | Verify sign in invalid password  | 1. Go to sign in page<br>2. Enter email<br>3. Enter password<br>4. Click sign up | account = invalidpwd@gmail.com<br><br>password = password.character  | Status: 400<br><br>MSG: Invalid password       |  |
| 11 | Verify sign in account too long  | 1. Go to sign in page<br>2. Enter email<br>3. Enter password<br>4. Click sign up | account = superlongusernameisnotallowed<br><br>password = longU  | Status: 501<br><br>MSG: format error           |  |
| 12 | Verify sign in email too long    | 1. Go to sign in page<br>2. Enter email<br>3. Enter password<br>4. Click sign up | account = this.is.a.really.long.email.address.contains.more.than.40.characters@gmail.com<br><br>password = longE | Status: 501<br><br>MSG: format error           |  |
| 13 | Verify sign in password too long | 1. Go to sign in page<br>2. Enter email<br>3. Enter password<br>4. Click sign up | account = longpwd@gmail.com<br><br>password = longpasswordlikethisisnotallowed                                   | Status: 501<br><br>MSG: format error           |  |
| 17 | Verify sign in                   | 1. Go to sign in page<br>2. Enter email<br>3. Enter password<br>4. Click sign up | account= user1@gmail.com<br><br>password = Superuser1  | Status: 201<br><br>MSG: sign in successfully   |  |
| 18 | Verify sign in                   | 1. Go to sign in page<br>2. Enter email<br>3. Enter password<br>4. Click sign up | account= user-2@gmail.com<br><br>password = Superuser2   | Status: 201<br><br>MSG: sign in successfully   |  |
| 19 | Verify sign in                   | 1. Go to sign in page<br>2. Enter email<br>3. Enter password<br>4. Click sign up | account= User1<br><br>password = Superuser1  | Status: 201<br><br>MSG: sign in successfully   |  |
| 20 |                                  |  |  |  |  |

## F (front end) - Sign up testing

This page will contain testing method and test cases for the sign-up function

|                 |   |
|-----------------|---|
| Test type       | Sign up   |
| Execution type  | Manual  |
| Objective       | Verify if a user is successfully signed up  |
| Setup           | cd backend3<br>npm start  |
| Pre-conditions  |   |
| Notes           | 1. Enter user name<br>2. Enter email<br>3. Enter password<br>*All fields are mandatory<br>*Username and email need to be unique.<br>*Maximum allowed number of chars(username:20, email: 50 password:16)<br>*Special character @ must be included in email field. |
| Time constraint | Minimum: 5 min<br>Maximum: 10 min   |

| No. | Test case                                      | Test steps  | Test data  | Expected result        | Actual result                                  |
|-----|--|---|--|------------------------|--|
| 1   | Verify sign-up with invalid email and password | 1. Go to Sign-Up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Click Sign up | email: cc<br><br>username: cc<br><br>password:123                    | "invalid email"        | "invalid email"                                |
| 2   | Verify sign-up with valid email and password   | 1. Go to Sign-Up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Click Sign up | email: cc@cc<br><br>username: cc<br><br>password:123                 | Sign up successfully   | Sign up successfully                           |
| 3   | Verify sign-up with valid email and password   | 1. Go to Sign-Up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Click Sign up | email:<br>user2@gmail.com<br><br>username: user2<br><br>password:123 | Sign up unsuccessfully | Sign up unsuccessfully, account already exist. |

## F - Sign in testing

This page will contain testing method and test cases for sign-in function

|                 |  |
|-----------------|--|
| Test type       | Sign up  |
| Execution type  | Manual   |
| Objective       | Verify if a user is successfully signed in   |
| Setup           | cd backend3<br>npm start   |
| Pre-conditions  |  |
| Notes           | 1. Enter user name or email<br>2. Enter password<br>*All fields are mandatory<br>*Username and email need to be unique.<br>*Maximum allowed number of chars(username:20, email: 50<br>password: 16)<br>*Special character @ must be included in email field. |
| Time constraint | Minimum: 5 min<br>Maximum: 10 min  |

| No. | Test case                                      | Test steps  | Test data                         | Expected result      | Actual result                        |
|-----|--|---|-----------------------------------|----------------------|--------------------------------------|
| 1   | Verify sign-in with invalid email and password | 1. Go to Sign-in page<br>2. Enter email<br>3. Enter password<br>4. Click Sign in    | email: cc<br>password:123         | "invalid email"      | "invalid email,<br>please try again" |
| 2   | Verify sign-in with valid email and password   | 1. Go to Sign-in page<br>2. Enter email<br>3. Enter password<br>4. Click Sign in    | email: cc@cc<br>password:123      | Sign in successfully | Sign in successfully                 |
| 3   | Verify sign-in with username and password      | 1. Go to Sign-in page<br>2. Enter username<br>3. Enter password<br>4. Click Sign in | username: abcdefg<br>password:123 | User not found       | User not found,<br>please sign up    |
| 4   | Verify sign-in with email and password         | 1. Go to Sign-in page<br>2. Enter email<br>3. Enter password<br>4. Click Sign in    | email: mnv@cc<br>password:123     | User not found       | User not found,<br>please sign up    |

|   |  |  |                                |                    |   |
|---|--|--|--------------------------------|--------------------|---|
| 5 | Verify sign-in with email and password | 1. Go to Sign-in page<br>2. Enter email<br>3. Enter password<br>4. Click Sign in | email: cc@cc<br>password:12345 | Incorrect password | Incorrect password,<br>please try again |
|---|--|--|--------------------------------|--------------------|---|

## F - reset password testing

This page will contain testing methods and test cases for reset password function

|                 |   |
|-----------------|---|
| Test type       | reset password  |
| Execution type  | Manual  |
| Objective       | Verify if a user is successfully reset the password   |
| Setup           | cd backend3<br>npm start  |
| Pre-conditions  | User has already logged in  |
| Notes           | 1. Go to profile page<br>2. Click manage profile<br>3. Enter new password<br>4. Click update<br><br>*Minimum allowed number of chars(password: 6) |
| Time constraint | Minimum: 5 min<br><br>Maximum: 10 min   |

| No. | Test case                                | Test steps   | Test data | Expected result | Actual result  |
|-----|--|--|-----------|-----------------|--|
| 1   | Change password with a valid password    | 1. Go to profile page<br>2. Click manage profile<br>3. Enter new password<br>4. Click update | q1w2e3r4  | Successful      | Update successfully  |
| 2   | Change password with a valid password    | 1. Go to profile page<br>2. Click manage profile<br>3. Enter new password<br>4. Click update | 123456    | Successful      | Update successfully  |
| 3   | Change password with an invalid password | 1. Go to profile page<br>2. Click manage profile<br>3. Enter new password<br>4. Click update | 12        | Unsuccessful    | Update unsuccessfully(password should be at least 6 characters long) |

## F - sort by like number

This page will contain testing methods and test cases for reset password function

|                 |   |
|-----------------|---|
| Test type       | sort by like number   |
| Execution type  | Manual  |
| Objective       | Verify if the sort function is work right   |
| Setup           | cd backend3<br>npm start  |
| Pre-conditions  | User has already logged in  |
| Notes           | 1. Go to search page<br>2. search anything<br>3. then click the sort by like (up or down) |
| Time constraint | less then one sec   |

| No. | Test case    | Test steps  | Test data | Expected result | Actual result                       |
|-----|--------------|---|-----------|-----------------|-------------------------------------|
| 1   | sort by up   | 1. Go to search page<br>2. search anything<br>3. then click the sort by like (up )  | demo      | Successful      | sort succeed all the data are right |
| 2   | sort by down | 1. Go to search page<br>2. search anything<br>3. then click the sort by like (down) | demo      | Successful      | sort succeed all the data are right |

## F- Update profile imformation tests

This page will contain testing methods and test cases for updating the profile information function

|                 |   |
|-----------------|---|
| Test type       | update profile  |
| Execution type  | Manual  |
| Objective       | Verify if a user can successfully update his/her profile  |
| Setup           | cd backend3<br>npm start  |
| Pre-conditions  | User has already logged in  |
| Notes           | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update<br><br>*Age is between 0-150<br><br>*Username can not be registered |
| Time constraint | Minimum: 5 min<br><br>Maximum: 10 min   |

| No. | Test case                                     | Test steps  | Test data | Expected result | Actual result          |
|-----|---|---|-----------|-----------------|------------------------|
| 1   | Enter a new username                          | 1. Go to profile page<br>2. Click manage profile<br>3. Enterupdated information<br>4. Click update  | hxx0329   | Successful      | Update successfully    |
| 2   | Enter a new username that has been registered | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update | user2     | Unsuccessful    | Username already exist |
| 3   | Enter updated age                             | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update | 18        | Successful      | Update successfully    |
| 4   | Enter an age out of the domain                | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update | 151       | Unsuccessful    | Age is above 150       |
| 5   | Enter an age out of the domain                | 1. Go to profile page<br>2. Click manage profile  | -1        | Unsuccessful    | Age is below 0         |

|   |                 |   |      |            |                   |
|---|-----------------|---|------|------------|-------------------|
|   |                 | 3. Enter updated information<br>4. Click update   |      |            |                   |
| 6 | Select a gender | 1. Go to profile page<br>2. Click manage profile<br>3. Select a gender<br>4. Click update | Male | Successful | Update successful |

## F - search the video by name

This page will contain testing methods and test cases for reset password function

|                 |   |
|-----------------|---|
| Test type       | search the video by name                    |
| Execution type  | Manual                                      |
| Objective       | Verify if the search function is work right |
| Setup           | cd backend3<br>npm start                    |
| Pre-conditions  | User has already logged in                  |
| Notes           | 1. Go to search page<br>2. search anything  |
| Time constraint | less then one sec                           |

| No. | Test case           | Test steps                                     | Test data    | Expected result | Actual result             |
|-----|---------------------|--|--------------|-----------------|---------------------------|
| 1   | search empty string | 1. Go to search page<br>2. search empty string | empty string | Successful      | return all the video      |
| 2   | search demo         | 1. Go to search page<br>2. search demo         | demo         | Successful      | return all the demo video |
| 3   | search 1            | 1. Go to search page<br>2. search 1            | 1            | Successful      | return demo1<br>demo11    |

## F - Delete video testing

This page will contain testing methods and test cases for the delete video function

|                 |   |
|-----------------|---|
| Test type       | delete video  |
| Execution type  | Manual  |
| Objective       | Delete a video that the user uploaded   |
| Setup           | cd backend3<br>npm start  |
| Pre-conditions  | User has already logged in  |
| Notes           | 1. Go to profile page<br>2. Click Manage video<br>3. Click delete<br>4. Click delete on the pop-up window |
| Time constraint | Minimum: 5 min<br>Maximum: 10 min   |

| No. | Test case                                     | Test steps  | Test data | Expected result | Actual result          |
|-----|---|---|-----------|-----------------|------------------------|
| 1   | Enter a new username                          | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update | hxx0329   | Successful      | Update successfully    |
| 2   | Enter a new username that has been registered | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update | user2     | Unsuccessful    | Username already exist |

## F - Upload Video Function

This page will contain testing methods and test cases for the upload video function

|                 |   |
|-----------------|---|
| Test type       | upload video  |
| Execution type  | Manual  |
| Objective       | upload a video  |
| Setup           | cd backend3<br>npm start  |
| Pre-conditions  | User has already logged in  |
| Notes           | 1. Go to profile page<br>2. Click Upload Video<br>3. Select a MP4 file<br>4. Enter the title, category and description<br>5. Click Upload |
| Time constraint | Minimum: 5 min<br><br>Maximum: 10 min   |

| No. | Test case  | Test steps  | Test data  | Expected result | Actual result                             |
|-----|--|---|--|-----------------|---|
| 1   | Upload a video with all the information required | 1. Go to profile page<br>2. Click Upload Video<br>3. Select a MP4 file<br>4. Enter the title, category and description<br>5. Click Upload | a MP4 file,<br>a title,<br>select a category,<br>a description | Successful      | Upload successfully                       |
| 2   | Upload a video without select a file             | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update                                       | a title,<br>select a category,<br>a description                | Unsuccessful    | Please select a file or fill out the form |
| 3   | Upload a video without a title                   | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update                                       | a MP4 file,<br>select a category,<br>a description             | Unsuccessful    | Please select a file or fill out the form |
| 4   | Upload a video without a category                | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update                                       | a MP4 file,<br>a title,<br>a description                       | Unsuccessful    | Please select a category                  |
| 5   | Upload a video without a description             | 1. Go to profile page<br>2. Click manage profile  | a MP4 file,  | Unsuccessful    | Please give a description                 |

|  |   |                                |  |
|--|---|--------------------------------|--|
|  | 3. Enter updated information<br>4. Click update | a title,<br>select a category, |  |
|--|---|--------------------------------|--|

## F - Delete user as well as all related videos in admin mode

This page will contain testing methods and test cases for the delete user function in admin mode

|                 |   |
|-----------------|---|
| Test type       | delete user in admin mode   |
| Execution type  | Manual  |
| Objective       | delete user and user's all uploaded videos should be deleted                          |
| Setup           | cd backend3<br>npm start  |
| Pre-conditions  | Administrator has already logged in   |
| Notes           | 1. Select account want to delete<br>2. Click Delete button<br>3. Press delete account |
| Time constraint | Less than 5 seconds   |

| No. | Test case    | Test steps   | Test data                           | Expected result  | Actual result   |
|-----|--------------|--|-------------------------------------|--|---|
| 1   | Delete user1 | 1. Sign in with administrator account<br>2. Select account want to delete<br>3. Click Delete button<br>4. Press delete account | Delete account with username: user1 | Delete account Successfully and all related videos also delete | Account Delete successfully and return to homepage related videos have been deleted |
| 2   | Delete user2 | 1. Sign in with administrator account<br>2. Select account want to delete<br>3. Click Delete button<br>4. Press delete account | Delete account with username: user2 | Delete account Successfully and all related videos also delete | Account Delete successfully and return to homepage related videos have been deleted |
| 3   | Delete user5 | 1. Sign in with administrator account<br>2. Select account want to delete<br>3. Click Delete button<br>4. Press delete account | Delete account with username: user5 | Delete account Successfully and all related videos also delete | Account Delete successfully and return to homepage related videos have been deleted |
| 4   | Delete admin | 1. Sign in with administrator account<br>2. Select account want to delete<br>3. Click Delete button<br>4. Press delete account | Delete account with username: admin | Fail to delete   | Administrator cannot be deleted   |

## F - Update user information in admin mode

This page will contain testing methods and test cases for the delete user function in admin mode

|                 |  |
|-----------------|--|
| Test type       | update user in admin mode  |
| Execution type  | Manual   |
| Objective       | update user information, go to account profile the author of all uploaded videos are updated and all updated information display correctly |
| Setup           | cd backend3<br>npm start   |
| Pre-conditions  | Administrator has already logged in  |
| Notes           | 1. Select account want to update<br>2. Click Edit button<br>3. Press update  |
| Time constraint | Minimum: 20 seconds<br>Maximum: 1 min  |

| No. | Test case    | Test steps  | Test data   | Expected result   | Actual result  |
|-----|--------------|---|---|---|--|
| 1   | Update user1 | 1. Sign in with administrator account<br>2. Select account want to update<br>3. Click Edit button<br>4. Fill in data want to update<br>5. Click update button | change username to user1000,<br>change age to 50,<br>change gender to male  | Update account Successfully, all updated information should be displayed correctly in account's profile and the author of all uploaded videos should be updated | Account updated successfully and go to account profile the author of all uploaded videos are updated and all updated information display correctly |
| 2   | Update user2 | 1. Sign in with administrator account<br>2. Select account want to update<br>3. Click Edit button<br>4. Fill in data want to update<br>5. Click update button | change username to user1000,<br>change age to 50,<br>change gender to male  | Fail to update  | Username has already existed.  |
| 3   | Update user2 | 1. Sign in with administrator account<br>2. Select account want to update<br>3. Click Edit button<br>4. Fill in data want to update                           | change username to user200,<br>change age to 30,<br>change gender to female | Update account Successfully, all updated information should be displayed correctly in account's profile and the author of all                                   | Account updated successfully and go to account profile the author of all uploaded videos are updated and all                                       |

|   |              |   |                           |                                      |  |
|---|--------------|---|---------------------------|--------------------------------------|--|
|   |              | 5. Click update button  |                           | uploaded videos<br>should be updated | updated information<br>display correctly |
| 4 | Update admin | 1. Sign in with administrator account<br>2. Select account want to update<br>3. Click Edit button<br>4. Fill in data want to update<br>5. Click update button | Change username to admin2 | Fail to update                       | Administrator cannot be updated          |

## Quality

# Deployment

This page keeps all track of deployment process.

\*\* Deployment is not part of our client's requirement but we would like to give it a try.

\*\* It is set as a challenge goal and will not be prioritized if there are other more important tasks need to be done.

Website link: [Heroku | Welcome to your new app! \(vshare-5vb-au.herokuapp.com\)](https://vshare-5vb-au.herokuapp.com)

- Up to 12/10:  
Deployed via github connection but build failed
- Up to 17/10:  
Build succussed

## Peer Feedback

| Peer Evaluation Form   |  |          |          |      |
|--|--|----------|----------|------|
| Criteria   | Filled out by:   |          |          |      |
|  | Member 2   | Member 3 | Member 4 | Self |
| <b>1 Quality of Technical Work:</b> Work is correct, clear, complete, and relevant to the problem. Equations, graphs, and notes are clear and intelligible.  |  |          |          |      |
| <b>2 Commitment to Team / Project:</b> Attends all meetings. Arrives on time or early. Prepared. Ready to work. Dependable, faithful, reliable.  |  |          |          |      |
| <b>3 Leadership:</b> Takes initiative, makes suggestions, provides focus. Creative? Energetic? Brings energy and excitement to the team. Has a "can do" attitude. Sparks creativity in others.   |  |          |          |      |
| <b>4 Responsibility:</b> Gladly accepts work and gets it done. Spirit of excellence.   |  |          |          |      |
| <b>5 Has abilities the team needs.</b> Makes the most of these abilities. Gives fully, doesn't hold back.  |  |          |          |      |
| <b>6 Communication:</b> Communicates clearly when he/she speaks and when she/he writes. Understands the team's direction.  |  |          |          |      |
| <b>7 Personality.</b> Positive attitudes, encourages others. Seeks consensus. Fun to deal with. Brings out best in others. Peacemaker. Pours water, not gasoline on fires.   |  |          |          |      |
| <b>Average grade:</b>  |  |          |          |      |
| <b>Grading scale:</b>  | 5 – Always, 4 – Most of the time, 3 – Sometimes, 2 – Rarely, 1 – Never |          |          |      |
| Keep in mind that if you award high scores to everyone, regardless of their contribution, team members who have worked unduly hard or provided extraordinary leadership will go unrecognized, as will those at the other end of the scale who need your corrective feedback. |  |          |          |      |

# Xuxiao He

| Criteria                              | Member 1 | Member 2 | Member 3 | Member 4 | Member 5 |
|---------------------------------------|----------|----------|----------|----------|----------|
| 1 Quality of Technical work           | 5        | 5        | 5        | 5        |          |
| 2 Commitment To team/project          | 5        | 5        | 5        | 5        |          |
| 3 Leadership                          | 5        | 5        | 5        | 5        |          |
| 4 Responsibility                      | 5        | 5        | 5        | 5        |          |
| 5 Has <b>abilities</b> the team needs | 5        | 5        | 5        | 5        |          |
| 6 Communication                       | 5        | 5        | 5        | 5        |          |
| 7 Personality.                        | 5        | 5        | 5        | 5        |          |

Xuxiao is our frontend developer in our team and is responsible for the UI design and interactions with users such as homepage display and account profile. I suppose Xuxiao really does a great job in his part and he is a nice person to get along with. When do the team meeting, he can always provide with some useful advice to help the whole team to develop the product well. He always tries his best to give sufficient support to someone who gets in stuck and in troubles.

Xuxiao is hardworking and contributed a lot towards user experience. He provided many solutions for certain technical issues and optimized sections of our frontend coding. Xuxiao is reliable and always finish his tasks on time. He is also willing to help out teammates if they need a hand.

Xuxiao is hardworking and do great job on frontend end, he is a nice person to communicate and give a lot of great idea to our group. He also do a good job on UI design

# Qi Guo

| Criteria                              | Member 1 | Member 2 | Member 3 | Member 4 | Member 5 |
|---------------------------------------|----------|----------|----------|----------|----------|
| 1 Quality of Technical work           | 5        | 5        | 5        | 5        |          |
| 2 Commitment To team/project          | 5        | 5        | 5        | 5        |          |
| 3 Leadership                          | 5        | 5        | 5        | 5        |          |
| 4 Responsibility                      | 5        | 5        | 5        | 5        |          |
| 5 Has <b>abilities</b> the team needs | 5        | 5        | 5        | 5        |          |
| 6 Communication                       | 5        | 5        | 5        | 5        |          |
| 7 Personality.                        | 5        | 5        | 5        | 5        |          |

Qi is our frontend developer in our team and is responsible for the UI design and interactions with the users such as searching and sorting. He is definitely the hardest person in our team since he has many other subjects to struggle with this semester, but he can coordinate his time well. He provides with some innovative ideas in the design stage and he takes good responsibility for his work. I think he is very friendly and humorous.

Qi takes his responsibilities seriously and looks into details. He often thinks from user's perspective and is keen to elevate and produce an intelligent system. Patiently explaining his ideas, Qi helps the team by keeping everyone on the same page. With a clear understanding of team progress and goal, everyone is able to work efficiently.

# Congcong Zhao

| Criteria                              | Member 1 | Member 2 | Member 3 | Member 4 | Member 5 |
|---------------------------------------|----------|----------|----------|----------|----------|
| 1 Quality of Technical work           | 5        | 5        | 5        | 5        |          |
| 2 Commitment To team/project          | 5        | 5        | 5        | 5        |          |
| 3 Leadership                          | 5        | 5        | 5        | 5        |          |
| 4 Responsibility                      | 5        | 5        | 5        | 5        |          |
| 5 Has <b>abilities</b> the team needs | 5        | 5        | 5        | 5        |          |
| 6 Communication                       | 5        | 5        | 5        | 5        |          |
| 7 Personality.                        | 5        | 5        | 5        | 5        |          |

Congcong handled most communication with client. He is persuasive enough to sell our ideas and designs to client. Within the team, he is supportive and checks on both ends frequently. His careful personality benefited us and made our product more intelligent, by targeting down minor features those could be designed better.

Congcong is a great and hard person, he always looks forward to communicate and give a lot of advices during the design part, he also codes hard in this project, we fixed a lot of bug together, and when I meet a bug ,he always gives me a hand to debug.

# Jialin Ding

| Criteria                              | Member 1 | Member 2 | Member 3 | Member 4 | Member 5 |
|---------------------------------------|----------|----------|----------|----------|----------|
| 1 Quality of Technical work           | 5        | 5        | 5        | 5        |          |
| 2 Commitment To team/project          | 5        | 5        | 5        | 5        |          |
| 3 Leadership                          | 5        | 5        | 5        | 5        |          |
| 4 Responsibility                      | 5        | 5        | 5        | 5        |          |
| 5 Has <b>abilities</b> the team needs | 5        | 5        | 5        | 5        |          |
| 6 Communication                       | 5        | 5        | 5        | 5        |          |
| 7 Personality.                        | 5        | 5        | 5        | 5        |          |

Jialin is our backend manager in our team and she is one of the backbones in our team. She is responsible for our meeting recording and always the person to summarize and make plans we would do next. For her work, she pay lots effort on the back end and always has positive and frequent communications with the front-end members to share her idea and improve the logic and structure of the product.

jialin is the people who show our product to our client, she also focus on the backend code. In my mind, she is the people like a manager. Also she is a positive person.

# Liping Meng

| eCriteria                             | Member 1 | Member 2 | Member 3 | Member 4 | Member 5 |
|---------------------------------------|----------|----------|----------|----------|----------|
| 1 Quality of Technical work           | 5        | 5        | 5        | 5        |          |
| 2 Commitment To team/project          | 5        | 5        | 5        | 5        |          |
| 3 Leadership                          | 5        | 5        | 5        | 5        |          |
| 4 Responsibility                      | 5        | 5        | 5        | 5        |          |
| 5 Has <b>abilities</b> the team needs | 5        | 5        | 5        | 5        |          |
| 6 Communication                       | 5        | 5        | 5        | 5        |          |
| 7 Personality.                        | 5        | 5        | 5        | 5        |          |

Liping is the backend developer in our team and she has searched lots of web development tutorials to finish her part well. In the team meeting, she can provides with some useful advice and always talk with the front-end members with her idea about the stucture and logic of the product. She takes her responsiblity well and solve a difficult problem like uploading the video. I think she is really a nice and patient person to get along with.

Liping always take her tasks seriously and finishes them ahead of schedule. This encourages other members in the team to work more and faster and she is the main factor boosting our efficiency. She has strong self-learn ability which makes her a reliable person to go to when there are difficult technical issues to be solved.

Liping is easy to communicate, she can solve all the request that our fro tend end need quickly, she also help us to make some big decisions. During the coding progress, she works hard and reply message quickly

# Product Report

VShare - by 5 vegetable birds

GitHub link: <https://github.com/Kisupan/IT-PROJECT.git>

Website: [Heroku | Welcome to your new app! \(vshare-5vb-au.herokuapp.com\)](https://vshare-5vb-au.herokuapp.com/)

If you cannot access any of these links, please contact  
[jialding@student.unimelb.edu.au](mailto:jialding@student.unimelb.edu.au)

You will need an account to access our website, please feel free to sign up your own account, or use one of our existed account:

Username: user3

Email: [user3@gmail.com](mailto:user3@gmail.com)

Password: 123456

\*Either username or email could be filled as account when logging in.

We still have some issues after deployment on the Internet. To completely show the functions of the product, you need open it on your local machine.

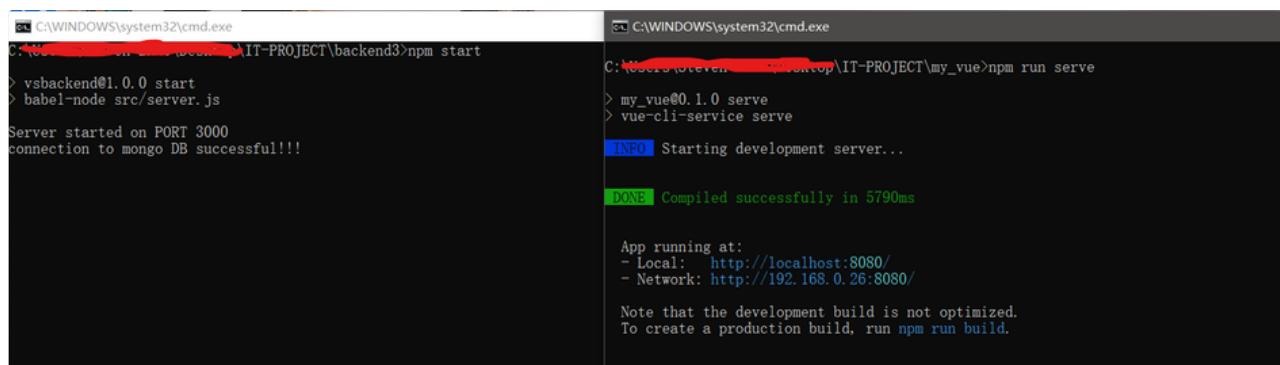
The followings are the process to run on the local machine:

- Find the file named server.js under the src folder in the backend3.

And then ignore the following codes with green lines as shown below

```
config();
const server = express();
// const history = require('connect-history-api-fallback')
// server.use(history())
// server.use(express.static('static'))
```

- Use the cmd terminal to enter the folder named backend3 in the IT-PROJECT directory type the command **npm start**
- Enter the folder my\_vue and type the command **npm run serve** and enter the URL in the browser and you can see the website. If nothing is shown on the homepage please refresh the page.



The image shows two separate command-line windows. The left window, titled 'C:\WINDOWS\system32\cmd.exe', is in the directory 'C:\Users\...\\IT-PROJECT\backend3'. It runs the command 'npm start' and outputs the following log:  
C:\> vsbackend@1.0.0 start  
> babel-node src/server.js  
Server started on PORT 3000  
connection to mongo DB successful!!!

The right window, also titled 'C:\WINDOWS\system32\cmd.exe', is in the directory 'C:\Users\...\\IT-PROJECT\my\_vue'. It runs the command 'npm run serve' and outputs the following log:  
C:\> my\_vue@0.1.0 serve  
> vue-cli-service serve  
INFO Starting development server...  
DONE Compiled successfully in 5790ms  
App running at:  
- Local: <http://localhost:8080/>  
- Network: <http://192.168.0.26:8080>  
Note that the development build is not optimized.  
To create a production build, run **npm run build**.

## Introduction

Our product is called VShare. VShare is a video website for users to share and upload their videos. Users can access more functions like making comments, and storing their favorite videos. We aim to provide users with a user-friendly interface to be available to users of all ages. We designed a navigation bar that can be easy for new users to find the videos they want by using the search bar and category list. On the homepage, you can see a sliding panel that provides users with the hottest and most popular videos at the moment.

## Requirement

The requirements are restricted for us. The client wanted us must have sign-in, sign-up, search and delete functions. In addition, we also asked for super accounts that can search for and delete users. For other requirements, such as likes and dislikes of the video, it is up to us. We decided to implement all the requirements as far as possible in order to make the website as fully functional as possible.

| As a <Role>   | I want to <Do something>      | So that<br><Achieve some goals>   | Priority |
|---------------|-------------------------------|---|----------|
| User          | Sign up                       | They has an account for this website. Normally, if the user first to access the website, they needs to signup. If the user does not sign in, they can only browser the website and could not do any actions.          | MEDIUM   |
| User          | Sign in                       | They can have the access to upload videos, and like/comment below videos. If the user does not sign in, they can only browser the website and could not do any actions.   | HIGH     |
| User          | Sign out                      | They can sign out the website if the user does not want to log in this website anymore.   | HIGH     |
| User          | Search key words              | They can find specific videos or users. The user can type key words in the search bar and web will return the list of relation videos.  | MEDIUM   |
| User          | Publish videos                | They can publish whatever they like. If the user want to upload a video, they can click upload video bar and add title and description. After publish the video, the user and others could see and comment the video. | HIGH     |
| User          | Have profile and edit profile | They can edit their personal information. The information includes username, email. And user can choose whether to add description or not.  | MEDIUM   |
| User          | Have favourite file           | They can store their favourite videos. If the user like a video, they can store video in a store list.  | LOW      |
| Administrator | add a normal user account     | If a new user could not sign up an account successfully, they can contact Administrator, Administrator could help them create a new account.  | HIGH     |
| Administrator | search a normal account       | can access a normal user quickly. The administrator can search a user through username in search bar  | HIGH     |
| Administrator | change a user's profile       | If a user meets some problems which could not edit their profile (user name, password), the Administrator could help them to edit this.   | HIGH     |
| Administrator | delete a user                 | The administrator can decide whether to delete a user. The administrator has responsibility to detect whether a user has  | HIGH     |

|                    |  |  |        |
|--------------------|--|--|--------|
|                    |  | appropriate comment. If the administrator thinks a user has unappropriated actions, the administrator can decide to delete the user directly. So all information about this user will not delete from database and other user will not see the user anymore. |        |
| Software developer | it is easy to maintain and update                                  | it is easy to maintain and update. Software developers should easy fix problems and bugs if an emergency event occurs. For example, if the client want to add a new function, the developer can quickly decide which part to add or change.                  | MEDIUM |
| User               | publish comment  | They can interact with other user.   | LOW    |
| User               | like a video or dislike a video                                    | They can help to adjust the video's popularity.  | MEDIUM |
| User               | upload and publish video   | The user can upload a video and publish it.  | HIGH   |
| User               | edit the profile details   | They can easily update their information such as username and other description about themselves.  | MEDIUM |
| User               | Delete/edit a video  | They can delete/edit videos that they no longer wish to share and re-upload a new version  | LOW    |
| User               | upload cover photo for video                                       | They can upload a photo as the cover to promote their videos, rather than system snipping one random scene from the video  | LOW    |
| Admin              | Have all users not listed in one page but in several pages instead | Don't need to view all users in one page   | MEDIUM |

## Design

A completed web development should include the back end and front end. Based on the current popular mode of web development, we decide to separate the front end and back end since it is convenient for us to divide the project into several pieces developing in parallel and check if errors occur on the front end or the back end.

### Front end

According to the front end, we use basic web tools including Html, CSS, JavaScript, and use the current prevalent front-end framework Vue as the main way to present the UI and interactions with the users.

### Back end

For the backend, we use the node.js environment and take the express module to establish the server for the project. Use MongoDB For data storing and retrieving.

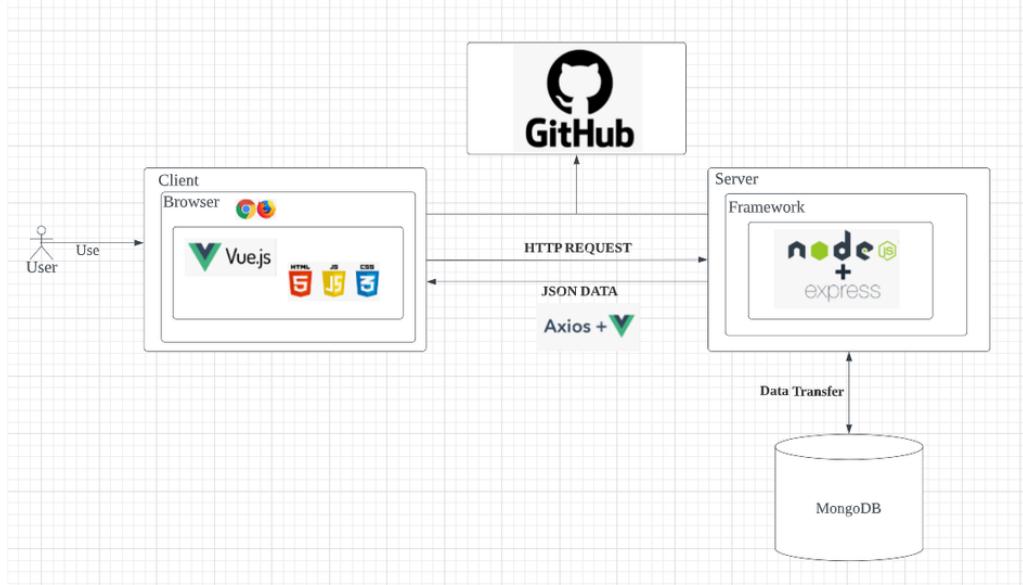
### Other tools

we also use other tools during the whole development like documentation with Confluence, Trello for task tracking, Wechat as the communication platform Canva for prototypes of the UI design, and GitHub for code management.

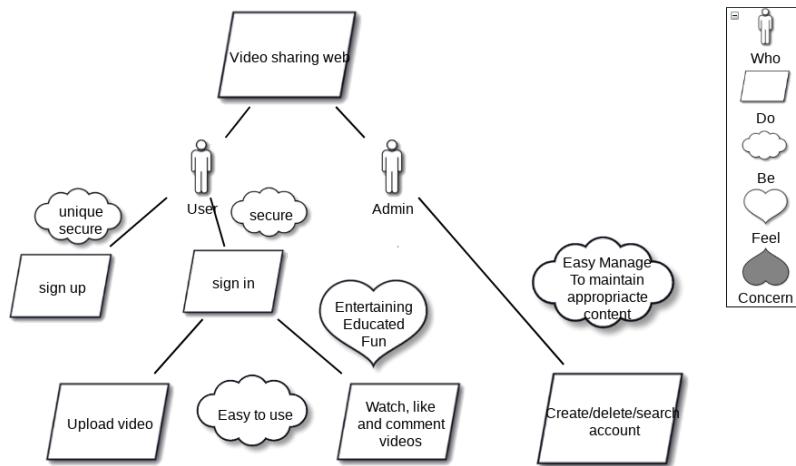
### Diagrams

Below are the Architecture Diagram and Motivational Model diagram for the product.

#### Architecture Diagram



**Motivational Model (MM) diagram**



## Features

### Sign in/up/out

We have a login system on our website, and users need to sign in first so that they can have access to the functions on the website, such as watching videos, managing profiles, like and disliking the video.



As shown above, we have a sign-in and a sign-up button on the top right of the website.

When clicking the sign-in button you need to fill in the account (username or email).

## Sign in

### Account

Email/User name

### Password

Password range 6-20

Cancel

Sign in

If you do not have an account you need to sign up for a new one, you need to fill out the correct form with the data like email address, user name, password, gender, and age.

\*All fields are mandatory

\*Username and email need to be unique.

\*Maximum allowed number of chars(username&password:20, email: 40)

\*Only numbers between 0-150 are allowed for age fields

\*Special character @ and . must be included in the email field, besides alphabets and numbers, special characters \_ is allowed in the email address

\*No special characters are allowed for usernames, alphabets, and numbers only.

## Sign up

### Email

example@example

### User name

User name

### Password

Password length need to be 6-20 range

### Gender

### Age

age between 0 ~ 150

Cancel

Sign up

After you have successfully signed in to the website, there is a sign-out button.

Sign out

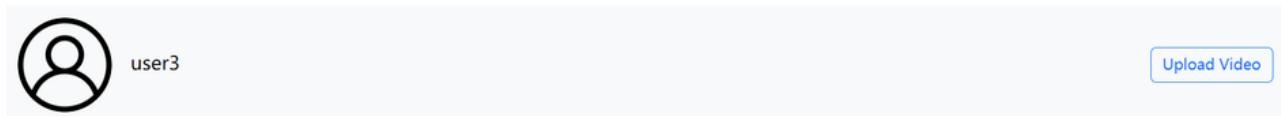
### Profile page

The profile page allows the user to view the video he/she has uploaded, delete videos, and modify profile details.

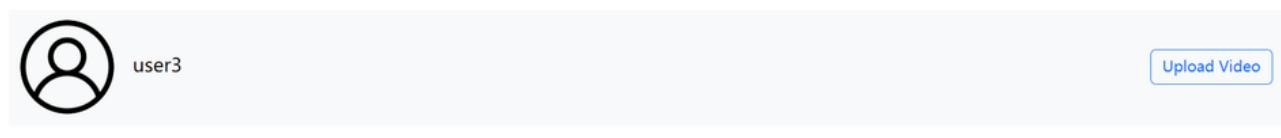
Click the avatar icon on any page to direct users to the profile page.

The “My Video” tab allows the user to access all the videos that have been uploaded by the user.

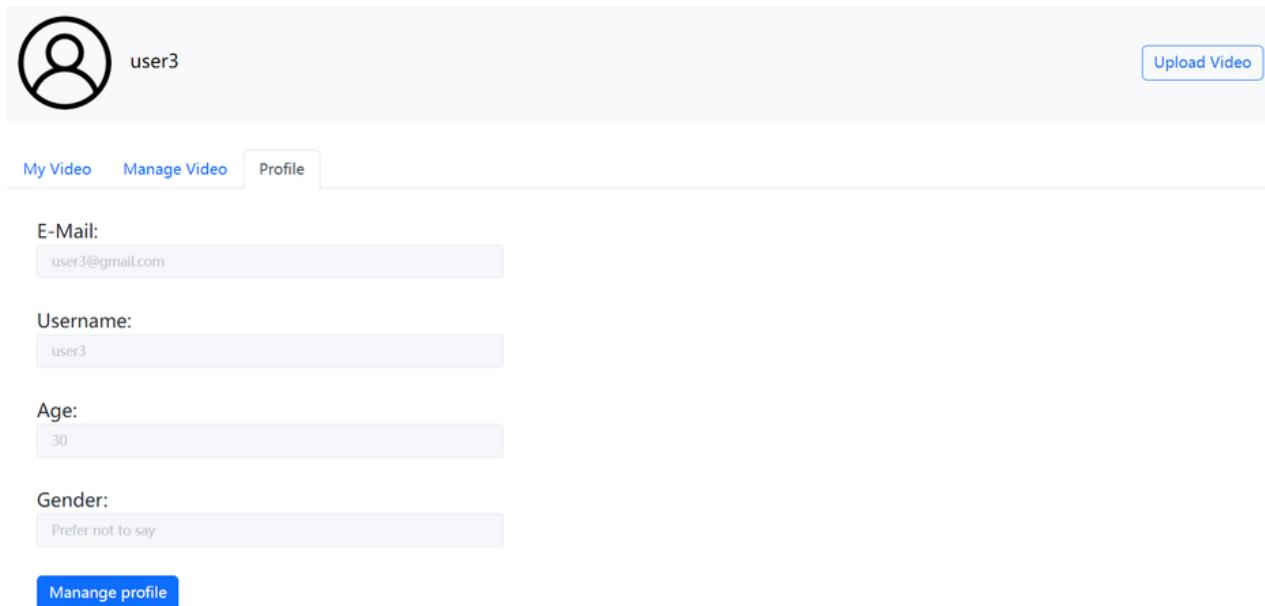
Click the “Manage Video” tab to delete videos, and the “Manage Profile” to view and change the profile details. Note that the e-mail cannot be changed.

A screenshot of the "My Video" tab. It features five video thumbnails arranged horizontally. Below each thumbnail is a white card containing the video title and uploader. From left to right: 1. A woman in a traditional outfit, titled "demo1" by "user3". 2. A person on a boat looking at a green landscape, titled "demo2" by "user3". 3. A corgi dog standing on a tiled floor, titled "demo3" by "user3". 4. An interior room scene, titled "demo5" by "user3".

| Video Title | Uploader |
|-------------|----------|
| demo1       | user3    |
| demo2       | user3    |
| demo3       | user3    |
| demo5       | user3    |

A screenshot of the "My Video" tab. It shows the same five video thumbnails as the previous screenshot. Below each thumbnail is a white card with the video title, uploader, and a red "Delete" button. From left to right: 1. A woman in a traditional outfit, titled "demo1" by "user3" with a "Delete" button. 2. A person on a boat looking at a green landscape, titled "demo2" by "user3" with a "Delete" button. 3. A corgi dog standing on a tiled floor, titled "demo3" by "user3" with a "Delete" button. 4. An interior room scene, titled "demo5" by "user3" with a "Delete" button.

| Video Title | Uploader | Action |
|-------------|----------|--------|
| demo1       | user3    | Delete |
| demo2       | user3    | Delete |
| demo3       | user3    | Delete |
| demo5       | user3    | Delete |



The screenshot shows a user profile page for 'user3'. At the top left is a placeholder user icon. To its right is the username 'user3'. On the far right is a blue 'Upload Video' button. Below the header is a navigation bar with three tabs: 'My Video' (blue), 'Manage Video' (blue), and 'Profile' (white). The 'Profile' tab is currently active. Underneath the tabs are four input fields: 'E-Mail:' with the value 'user3@gmail.com', 'Username:' with the value 'user3', 'Age:' with the value '30', and 'Gender:' with the value 'Prefer not to say'. At the bottom left of the profile section is a blue 'Manage profile' button.

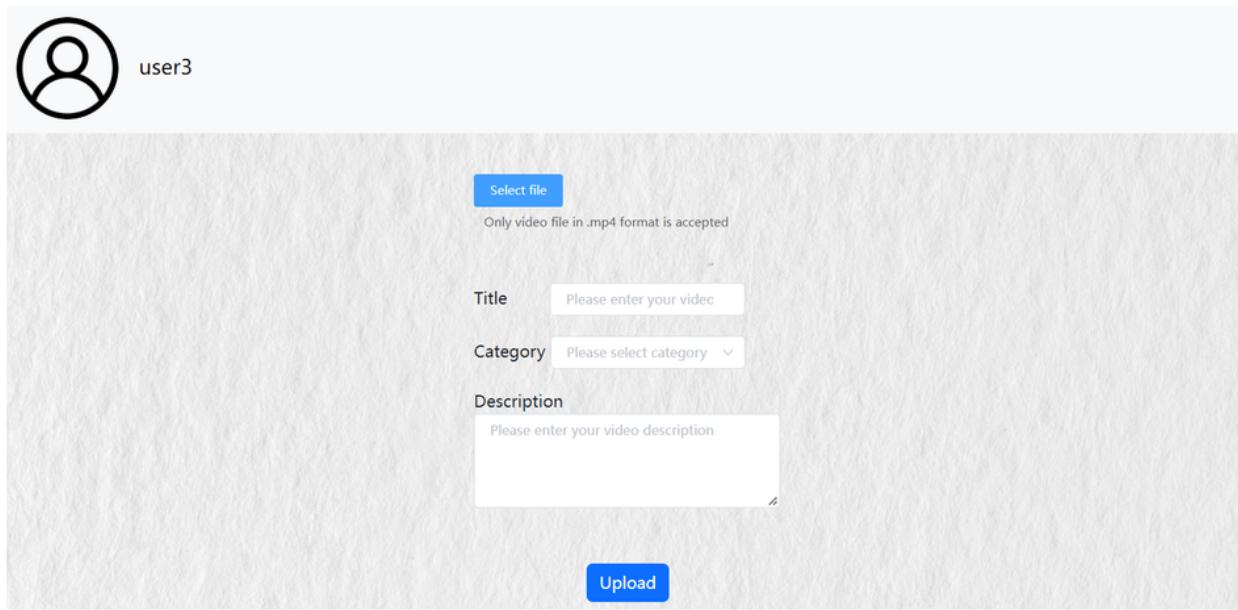
#### Upload videos

To upload videos, simply go to the profile page and click the “Upload Video” button, then it will direct you to the upload page.

The upload function only allows the user to upload one video at a time along with a video title, a description, and a category.

After all these steps, click the “Upload” button at the bottom of the page to upload.

Note: Video title should use the alphabet in lower case.

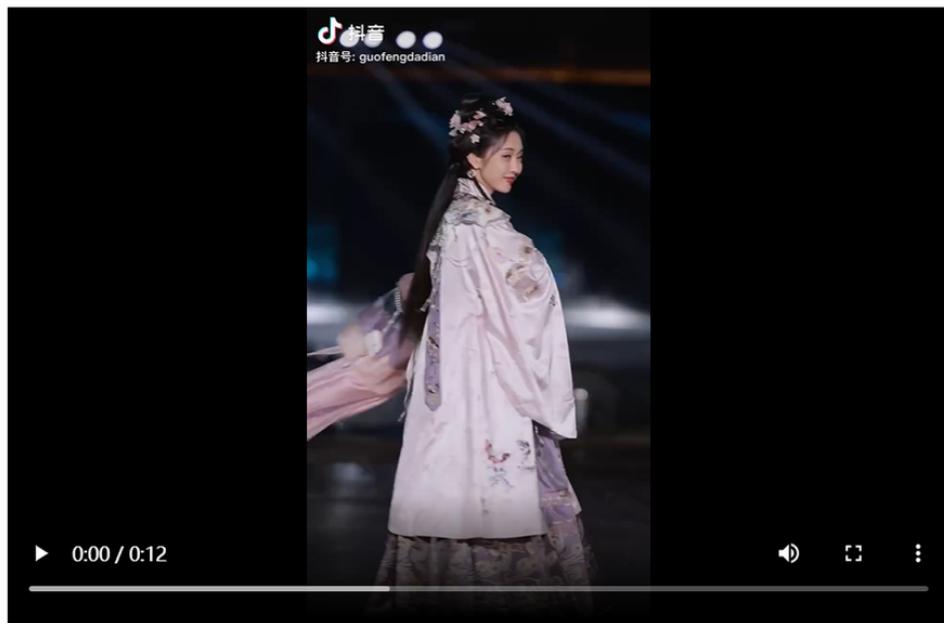


The screenshot shows the video upload interface for 'user3'. At the top left is a placeholder user icon. To its right is the username 'user3'. The main area contains a 'Select file' button with the instruction 'Only video file in .mp4 format is accepted'. Below this are three input fields: 'Title' (placeholder 'Please enter your video'), 'Category' (placeholder 'Please select category'), and 'Description' (placeholder 'Please enter your video description'). At the bottom right is a blue 'Upload' button.

#### Video play

This is a page where can watch the video and there are two icons below which are used for like and dislike a video by clicking these two icons.

# demo1



## Description

Chinese Hanfu

### Category page

you can find a video by its category:

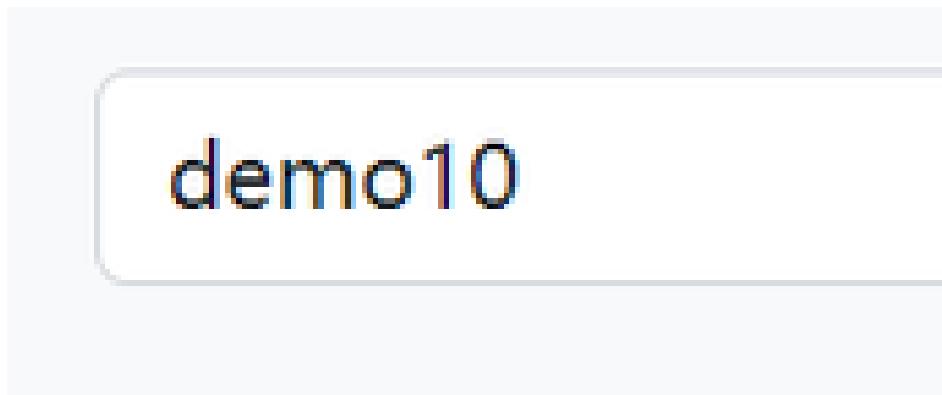
you can also change the form of the result:

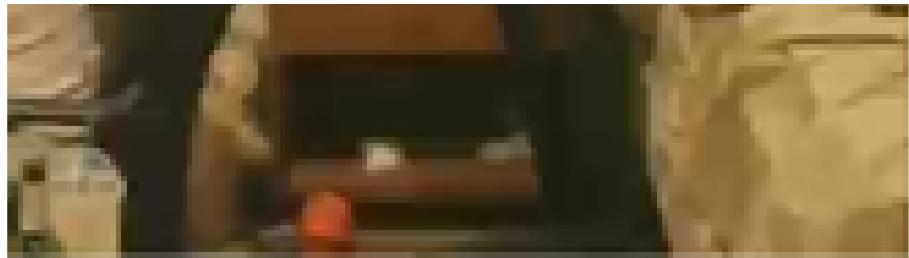
you can change the number of results per page:



Search and sort

By entering the key word in the search bar, it will show you the relative result that the name contains the key word.





You can also sort the result the number of likes.

### your search :

sort by like ▾

likes down

likes up

random



demo1

Author: user3

Description:Chinese Hanfu

label: Fashion

likes number:10 dislikes number:1



demo10

Author: user3

Description:Others

label: Others

likes number:6 dislikes number:1

### Admin mode

As client's request, we have an administration system for the website, and the administrator can manage all accounts that have already signed up on this website and do the following functions updating account information, deleting an account as well as adding a new account.

You need to sign in with an admin account and you can see the interface that shows all information of the accounts. (Account: admin, Password: Superadmin123)



Add Account

| Email Address   | User Name | Age | Gender            |                        |                      |
|-----------------|-----------|-----|-------------------|------------------------|----------------------|
| admin@gmail.com | admin     | 30  | Prefer not to say | <a href="#">Delete</a> | <a href="#">Edit</a> |
| user3@gmail.com | user3     | 30  | Prefer not to say | <a href="#">Delete</a> | <a href="#">Edit</a> |
| user4@gmail.com | user4     | 30  | Prefer not to say | <a href="#">Delete</a> | <a href="#">Edit</a> |
| sb@123456       | user5     | 20  | Prefer not to say | <a href="#">Delete</a> | <a href="#">Edit</a> |

Click the delete button to delete all information about the account including the uploaded videos from this account.

Click the Edit button and update the information.

Edit

---

Email

User name

Password

Gender

Age

---

[Cancel](#) [Update](#)

There is a search bar on the top of the admin system which can quickly find the related account.

| VShare          |           |     |                   | user3                  | Search               |
|-----------------|-----------|-----|-------------------|------------------------|----------------------|
| Back            |           |     |                   |                        |                      |
| Email Address   | User Name | Age | Gender            |                        |                      |
| user3@gmail.com | user3     | 30  | Prefer not to say | <a href="#">Delete</a> | <a href="#">Edit</a> |

## Coding

Please refer to our Github repository. Frontend and backend branches are built for pair-programming within each end and the main branch contains all updates toward the final product. Below are tools and coding language we used in this project.

## Frontend

For the front end, we decided to use HTML, CSS, JavaScript, Vue, Axios, and Bootstrap based on the following considerations:

- Bootstrap

Bootstrap is a powerful, extensible, and feature-packed frontend toolkit containing the most popular CSS Framework for developing websites, saving lots of time.

- Vue as a framework

Vue provides more built-in features and companion libraries from the core team. Vue framework is more friendly for a newer developer and not as familiar with advanced JavaScript concepts. This makes development faster and smoother.

- Axios

Axios is a promised-based HTTP request library. It can combine with the Vue framework to make HTTP requests from vue.js and handle the transformation of request and response data because of Vue's response-based data property.

## Backend

For the backend, we decided to use a combination of node.js, Express, and Mongo DB according to the following consideration:

- node.js. as environment

- Front and back-end unification.
- Easy to Learn and Quick to adapt. As new learners, developers can quickly study it and make it easy to understand.
- Improves App Response Time and Boosts Performance

- Express as a framework.

- Using Express.JS would be able to do the code of both frontend and backend with the help of using JavaScript
- Express makes the process easier
- Using Express.JS for backend development could be able to scale applications quickly

- Mongo DB as a database

- Full cloud-based developer data platform.
- Flexible document schemas.

## **Testing**

Our tests are part of the agile development. We test manually after each sprint release. Our testing focuses on three main areas: quality, defects, and feedback. We use agile development, so we communicate directly with our client during Sprint planning and work together to develop quality requirements for product features from the client's needs. The agile process emphasizes communication within the team. We often focus on the important features and concentrate on the details. In addition, we use unit testing to ensure the basic quality of the product, and we meet with our client to validate the consistency between client requirements and actual results. Finally, in terms of feedback, the agile process emphasizes simplicity and efficiency. So we need timely feedback on the product's current quality issues. This way, the team can immediately work to resolve them. In our projects, any team member who finds a defect will immediately report it to the other team members and then discuss how to fix it. Therefore, any defects can be immediately detected and modified.

## Backend

- Sign up - To check if user can sign up with invalid inputs (empty inputs, wrong email addresses, special characters).
- Sign in - To check if user can sign in with correct details matching with the database's.
- Reset password - To check if user can reset password and log in with new password.
- Search user - To check if only relative results will be returned.
- Search video - To check if only relative results will be returned.
- Show all users - To present all users on admin management page.
- Delete user - To check if an admin can permanently delete an user from the database.

- Upload video - To check if user can successfully upload videos under restrictions (mp4 file, <45 mb).
- Like/dislike video - To check if function is working, and if like/dislike could be cancelled if clicked by accident.

| No. | Test case                                      | Test steps  | Test data   | Expected result                                   | Actual result                                     |
|-----|--|---|---|---|---|
| 1   | Verify sign-up with invalid email and password | 1. Go to Sign-Up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Click Sign up | email: cc<br>username: cc<br>password:123                 | "invalid email"                                   | "invalid email"                                   |
| 2   | Verify sign-up with valid email and password   | 1. Go to Sign-Up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Click Sign up | email: cc@cc<br>username: cc<br>password:123              | Sign up successfully                              | Sign up successfully                              |
| 3   | Verify sign-up with valid email and password   | 1. Go to Sign-Up page<br>2. Enter email<br>3. Enter username<br>4. Enter password<br>5. Click Sign up | email: user2@gmail.com<br>username: user2<br>password:123 | Sign up unsuccessfully,<br>account already exist. | Sign up unsuccessfully,<br>account already exist. |

Partial sample of sign-up testing

## Frontend

Sign up testing

Sign in testing

Reset password testing

Delete video testing

Upload video testing

Change profile testing

Search video testing

Delete user testing (admin)

Update user information testing (admin)

| No. | Test case                                     | Test steps  | Test data | Expected result | Actual result          |
|-----|---|---|-----------|-----------------|------------------------|
| 1   | Enter a new username                          | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update | hxx0329   | Successful      | Update successfully    |
| 2   | Enter a new username that has been registered | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update | user2     | Unsuccessful    | Username already exist |
| 3   | Enter updated age                             | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information<br>4. Click update | 18        | Successful      | Update successfully    |
| 4   | Enter an age out of the domain                | 1. Go to profile page<br>2. Click manage profile<br>3. Enter updated information                    | 151       | Unsuccessful    | Age is above 150       |

## Deployment

Deployment is not required by our client so our team took it as a challenging task. For deployment, we connected our GitHub repository to Heroku. Heroku will automatically build and deploy after every push and return a log to local files. It is convenient for us to analyze the error as soon as possible. Since we have a almost-to-none budget, our database cannot store too many videos and sometimes videos will failed to be played after deployed due to packed and slow operating speed.

## Performance

The overall performance of our product is stable. We currently do not have any bugs.

For an upload the size of the video cannot exceed 45 MB. All videos should not be stored over 500 MB.

In addition, the more videos you store, the slower the web pages will load.

In addition, for the search function, the search function of our product is very smart and accurate. Both "k" and "K" results will be returned if searched either "k" or "K", but results such as "friend", or "dog" will not be returned.

## Demonstration and delivery

We showed the VShare to the client on 22/10 and handed the link. We started from how to set up personal profile and access the webpage, and walked through the functions of this project. We demonstrated how to upload, like/dislike, search videos, and how to edit user information as an admin. The client is very satisfied with our product and believed all features have met her requirements. We also discussed about future extendibility to boost user experience and commercial value, such as bringing in advertisements and preference-based algorithms.