

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 jjalding@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

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.

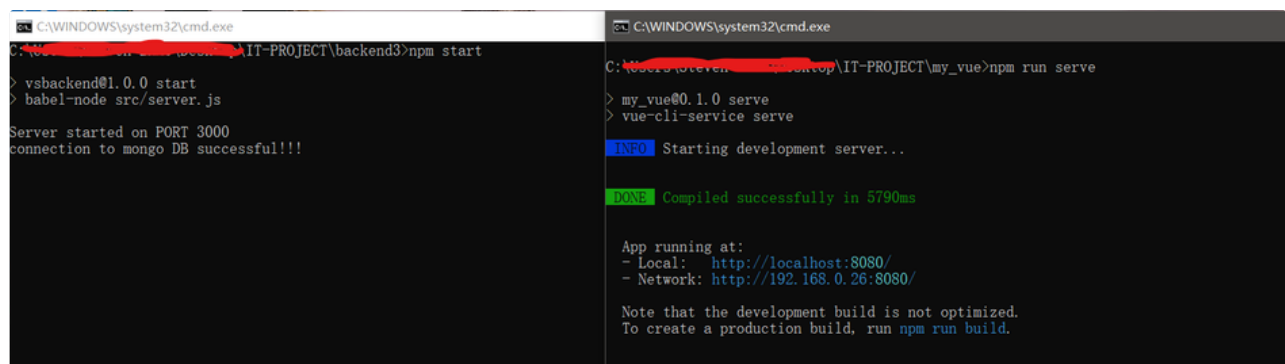
The followings are the process to run on the local machine:

- 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.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\jalding> cd IT-PROJECT\backend3 & npm start

> vsbackend@1.0.0 start
> babel-node src/server.js

Server started on PORT 3000
connection to mongo DB successful!!!

C:\WINDOWS\system32\cmd.exe
C:\Users\jalding> cd IT-PROJECT\my_vue & npm run serve

> 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 <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 tittle 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.

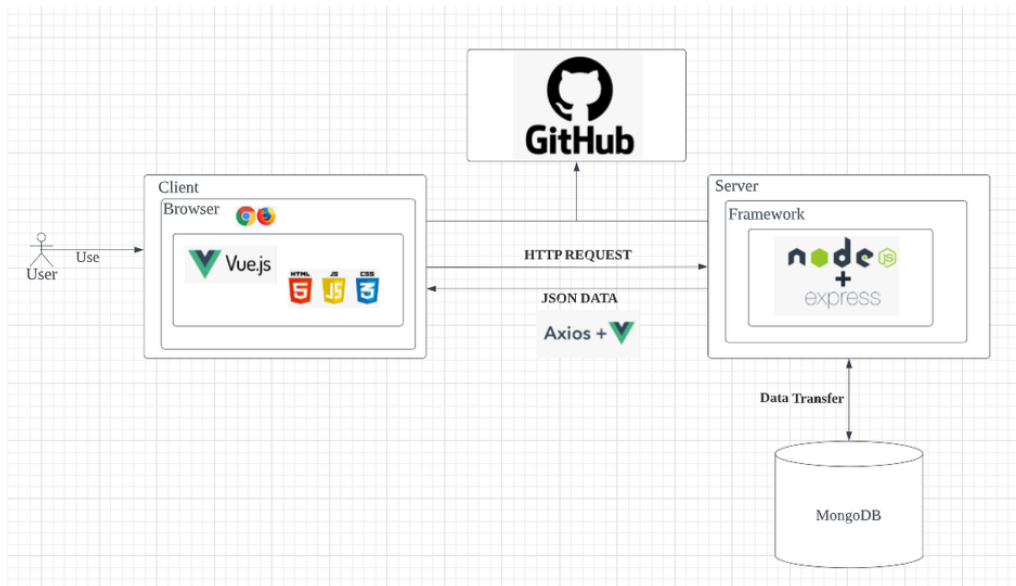
Other tools

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.

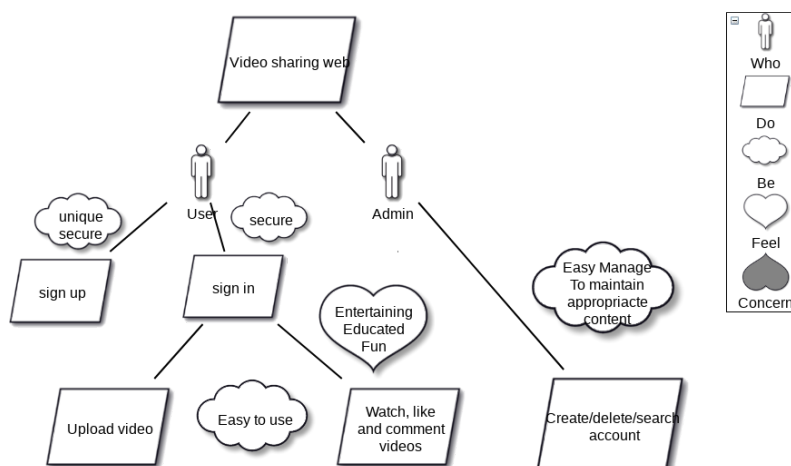
Diagrams

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

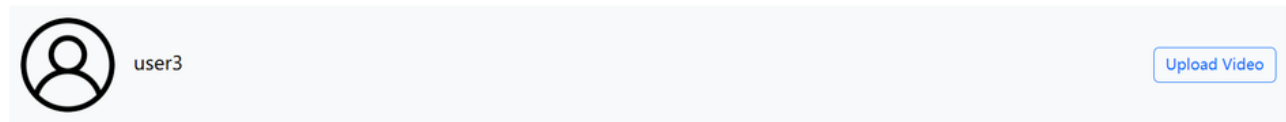
Profile page

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

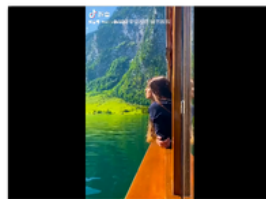
Click the avatar icon on any page to direct users to the profile page.

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

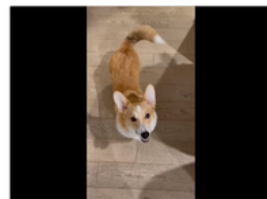
Click the "Manage Video" tab to delete videos, and the "Manage Profile" to view and change the profile details. Note that the e-mail cannot be changed.



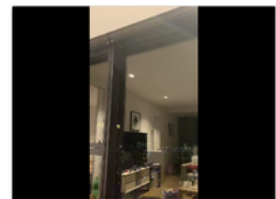
demo1
user3



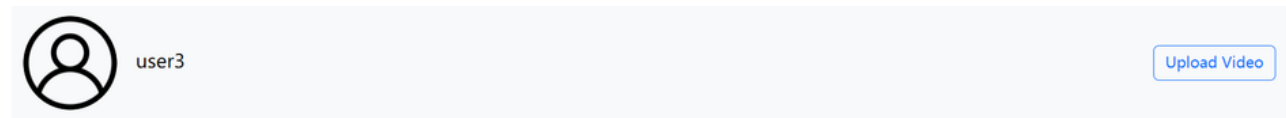
demo2
user3



demo3
user3

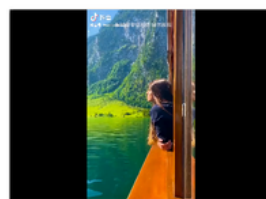


demo5
user3



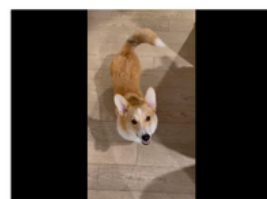
demo1
user3

Delete



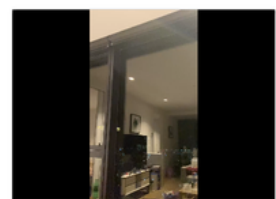
demo2
user3

Delete




demo3
user3

Delete



demo5
user3

Delete

 user3 Upload Video

My Video Manage Video Profile

E-Mail:
user3@gmail.com

Username:
user3

Age:
30

Gender:
Prefer not to say

Manange profile


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.

 user3

Select file
Only video file in .mp4 format is accepted

Title
Please enter your video title

Category
Please select category

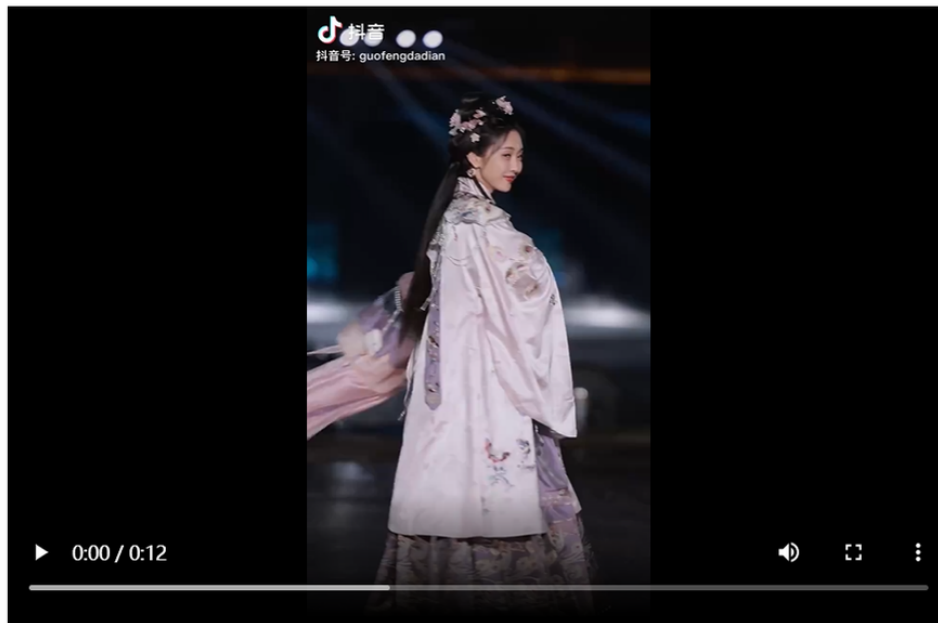
Description
Please enter your video description

Upload

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

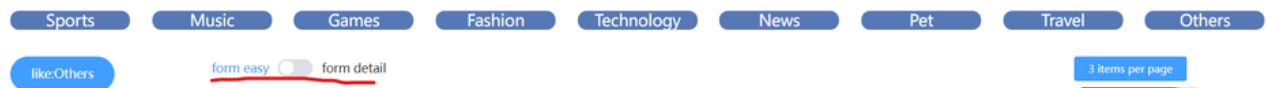
Chinese Hanfu

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:

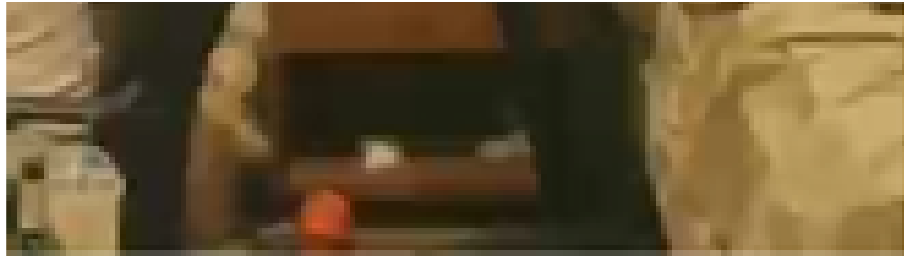


Search and sort

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

demo10





You can also sort the result the number of likes.

your search :

sort by like ▾

likes down

likes up

random



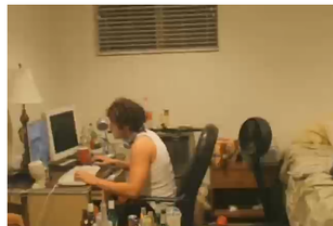
demo1

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)

VShare

Search Email/Username

Search

Sign out

Add Account

Email Address	User Name	Age	Gender		
admin@gmail.com	admin	30	Prefer not to say	Delete	Edit
user3@gmail.com	user3	30	Prefer not to say	Delete	Edit
user4@gmail.com	user4	30	Prefer not to say	Delete	Edit
sb@123456	user5	20	Prefer not to say	Delete	Edit

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

user3@gmail.com

User name

user3

Password

Leave blank if not want to change (Length 6-20 range)

Gender

Prefer not to say

Age

30

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	Delete	Edit

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 2. Enter email 3. Enter username 4. Enter password 5. Click Sign up	email: cc username: cc password:123	"invalid email"	"invalid email"
2	Verify sign-up with valid email and password	1. Go to Sign-Up page 2. Enter email 3. Enter username 4. Enter password 5. Click Sign up	email: cc@cc username: cc password:123	Sign up successfully	Sign up successfully
3	Verify sign-up with valid email and password	1. Go to Sign-Up page 2. Enter email 3. Enter username 4. Enter password 5. Click Sign up	email: user2@gmail.com username: user2 password:123	Sign up unsuccessfully	Sign up unsuccessfully, 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 2. Click manage profile 3. Enter updated information 4. Click update	hxx0329	Successful	Update successfully
2	Enter a new username that has been registered	1. Go to profile page 2. Click manage profile 3. Enter updated information 4. Click update	user2	Unsuccessful	Username already exist
3	Enter updated age	1. Go to profile page 2. Click manage profile 3. Enter updated information 4. Click update	18	Successful	Update successfully
4	Enter an age out of the domain	1. Go to profile page 2. Click manage profile 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.