Team Project Report

Youtube system of a query with lucidness (U-sql)

Group Name:XiXHuZSunY
Group Members: Xueqi Xi, Zhijun Hu, Yuming Sun

Introduction.

As the administrator of a video website, it is essential to employ effective methods for tracking users' subscriptions to video bloggers, and comments for videos of them, as well as managing employees who review videos posted by video bloggers and provide feedback on user issues. Such measures are crucial for promoting website growth. A database management system (DBMS) is a software application that facilitates the organized and efficient storage and management of large amounts of data. In the case of video websites such as YouTube, a DBMS plays a vital role in managing data related to videos, users, employees, comments, and other content. Accordingly, our study of the YouTube platform has led to the development of a user-friendly DBMS that enables easy and intuitive data retrieval, display management, and query data acquisition. We call it the 'Youtube system of a query with lucidness(U-sql)'.

There are several security features to protect user data, including passwords, input validation and data sanitization in the database connected with a user interface constructed by Python. Only registered users/administrators can access/modify the system, and each user's data is isolated from other users and only edited by corresponding user or admin.

The database is designed for efficient performance, with appropriate indexing and optimized SQL queries. The database can be hosted on a dedicated server with adequate processing power and memory to handle multiple user requests.

Technical specification.

MySQL Workbench which is a popular graphical tool was used to create and manage MySQL databases.Python programming language was used to connect to a MySQL database via 'pymysql'

module as front-end user interface and modules such as ‘matplotlib’ and ‘pandas’ were aimed for data visualization.Numpy-based libraries of python were used to construct regression for necessary problems, as well as Sklearn, and Tensorflow to build Neural-Networks in case necessary analysis is needed. There are no machine restrictions and OS efficiency satisfies the need of our project.

The datasets were generated by supervised generation tools instead of acquiring real-world datasets because it is difficult and time consuming to write network acquisition data programs and additional steps for data cleaning. The generated datasets can meet the format requirements of the database, and can be used to test the validity of the stored procedure and the accuracy of the interaction with the front-end program. This may affect the reliability and analysis part of our project, more details are included in the future work part.

The user interface was implemented using a Jupyter Notebook hosted on a web server. A prompt interface was provided by Notebook to access the database and perform CRUD operations by users. The user interface was divided into several sections, each of which allowed users to perform a specific set of operations. The detailed information is included in the User Flow part.

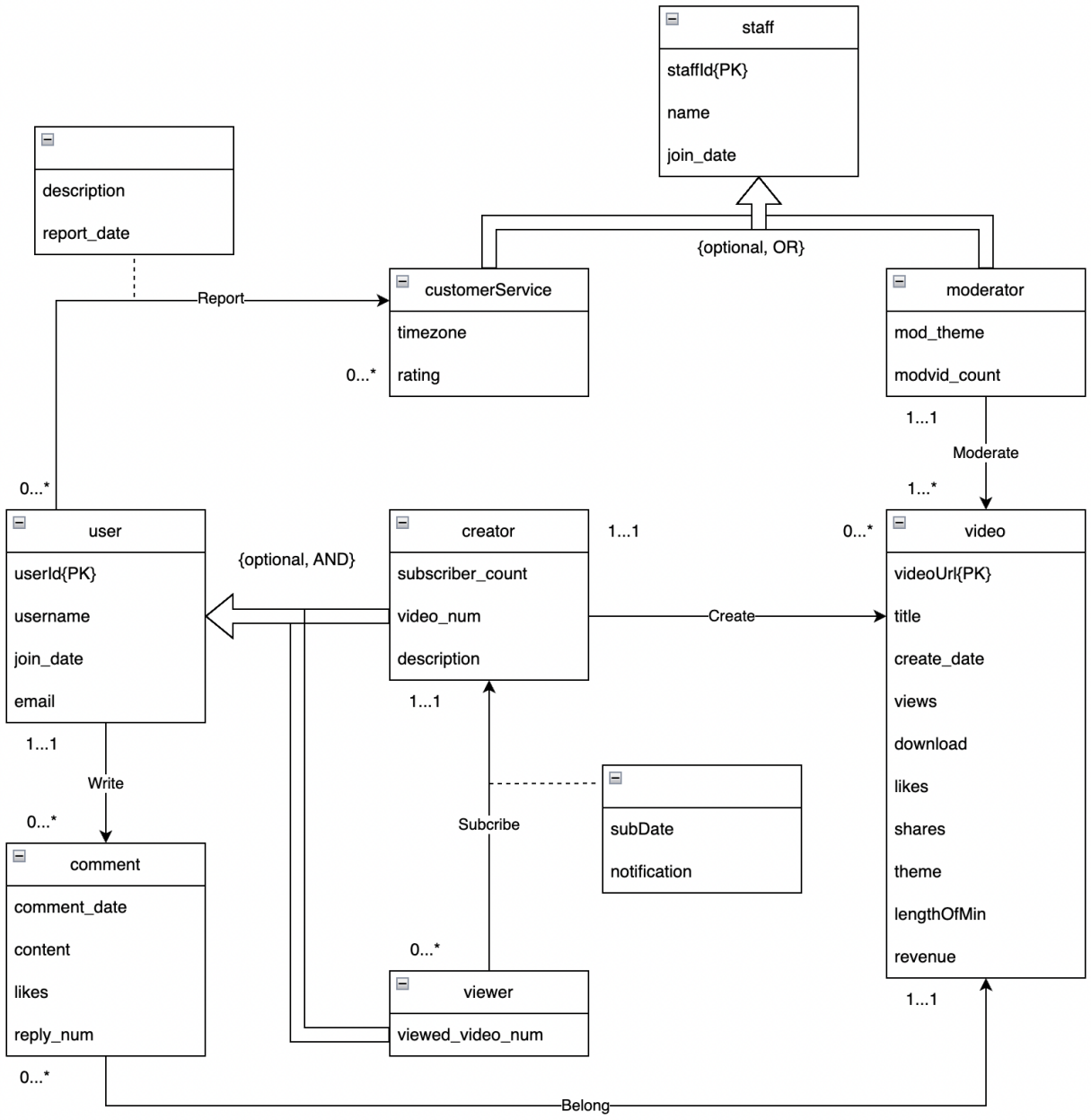
The database was implemented using MySQL as the DBMS. The database schema was designed to be in 3rd normal form including the following tables:

- Staff: Contains information about staff members, including their ID, name, and date of joining. It is a superclass of the table of CustomerService and Moderator.
- CustomerService: Contains information about customer service staff members, including their service ID (extends the Id in the table of Staff), timezone, and rating, with a foreign key reference to the Staff table.
- Moderator: Contains information about moderators, including their moderator ID, theme, and video count, with a foreign key reference to the Staff table.
- User: Contains information about users, including their ID, username, date of joining, and email. It is a superclass of the table of Creator and Viewer.
- Report: Contains information about user reports, including the user ID, customer service ID, description of the report, and report date, with foreign key references to the User and CustomerService tables.
- Creator: Contains information about content creators, including their ID, subscriber count, number of videos, and description of the creator, with a foreign key reference to the User table.
- Viewer: Contains information about viewers, including their ID and number of videos viewed, with a foreign key reference to the User table.
- Subscriber: Contains information about subscriptions, including the subscriber ID, subscription ID, subscription date, and notification preference, with foreign key references to the Viewer and Creator tables. **For a viewer to subscribe to a creator, a viewer has to press a button and can only subscribe to One and only One creator at a time, albeit this action can be repeated multiple times, it is still a 1..1 relationship.**
- Video: Contains information about videos, including the video URL, creator ID, title, creation date, number of views, downloads, likes, shares, theme, length in minutes, revenue, and moderator ID, with foreign key references to the Creator and Moderator tables.

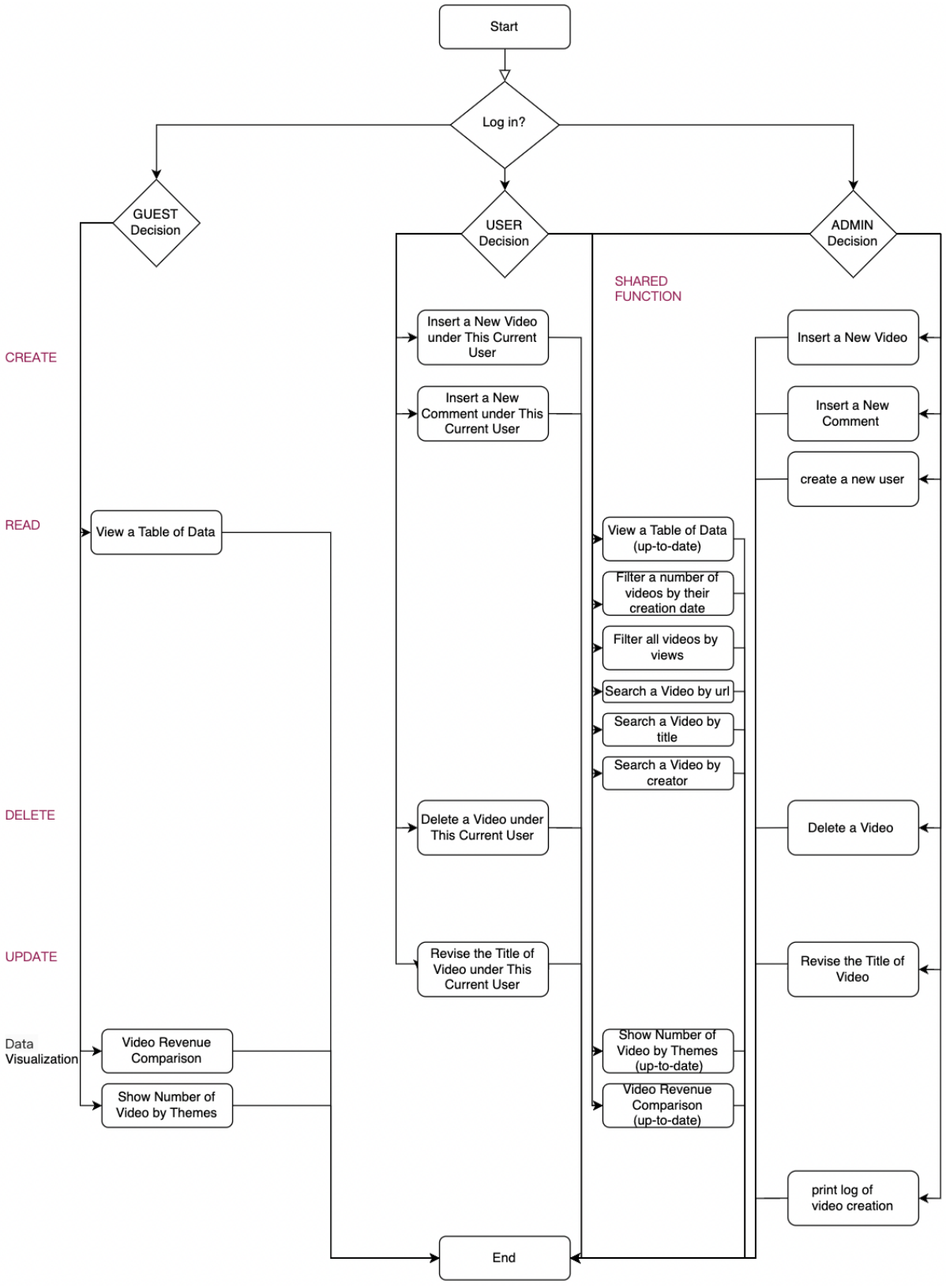
- Comment: Contains information about comments on videos, including the user ID, video URL, comment date, content, likes, and number of replies, with foreign key references to the User and Video tables.

The table properties of the database come from the entity attributes in the design of the uml diagram, and the two are not exactly the same.

UML Diagram



Activity Diagram.



README.

Prerequisites

Before getting started, you will need the following:

- MySQL installed on your local machine or a remote server
- MySQL connector for Python (`pymysql`) installed
- Jupyter Notebook (or `Python` terminal) installed

Setup

- Please load the dumped sql file on the MySQL server and keep the server active.

- open the file with the extension of .ipynb by Jupter, you will see on the top of the file there are several modules to be installed.

```
!pip install pymysql

!pip install pandas

!pip install numpy

!pip install seaborn

!pip install matplotlib

!pip install cryptography
```

- In the [2] of the files, please set your username to `userg` and password to `passg` of MySQL server in order to connect to the database via Python.
- Be sure that in the [2] [5] [12]the variable of `database` is the name of the database that you load in the server.

- Please run the code from here to the bottom with the last block of code that works as a GUI system. Only the last block of code is supposed to be rerun during the whole process, no exceptions.

Make sure you've all the pre-requisite installed, if not, run the first block of code before running any other ones.

Usage

After you successfully run all the previous codes, you will be prompted with the following words `If you are a guest, enter 1; If you are a user, enter 2. Else, enter anything else and you will be redirected to login page.` when you run [26] or [*], which means you have successfully run the system.

- To enter 1 will get you access to the guest view. You can get the userId and username of a user for enter 1 - 1 - 8. (! the result of the userId is just to let you pair with the username to get access of its user options successfully, not to prove the program have the possibility of disclosing data information).

- To enter 2 will lead you to enter the userId, and its username to access the user option (userId=1 username = johndoe123)

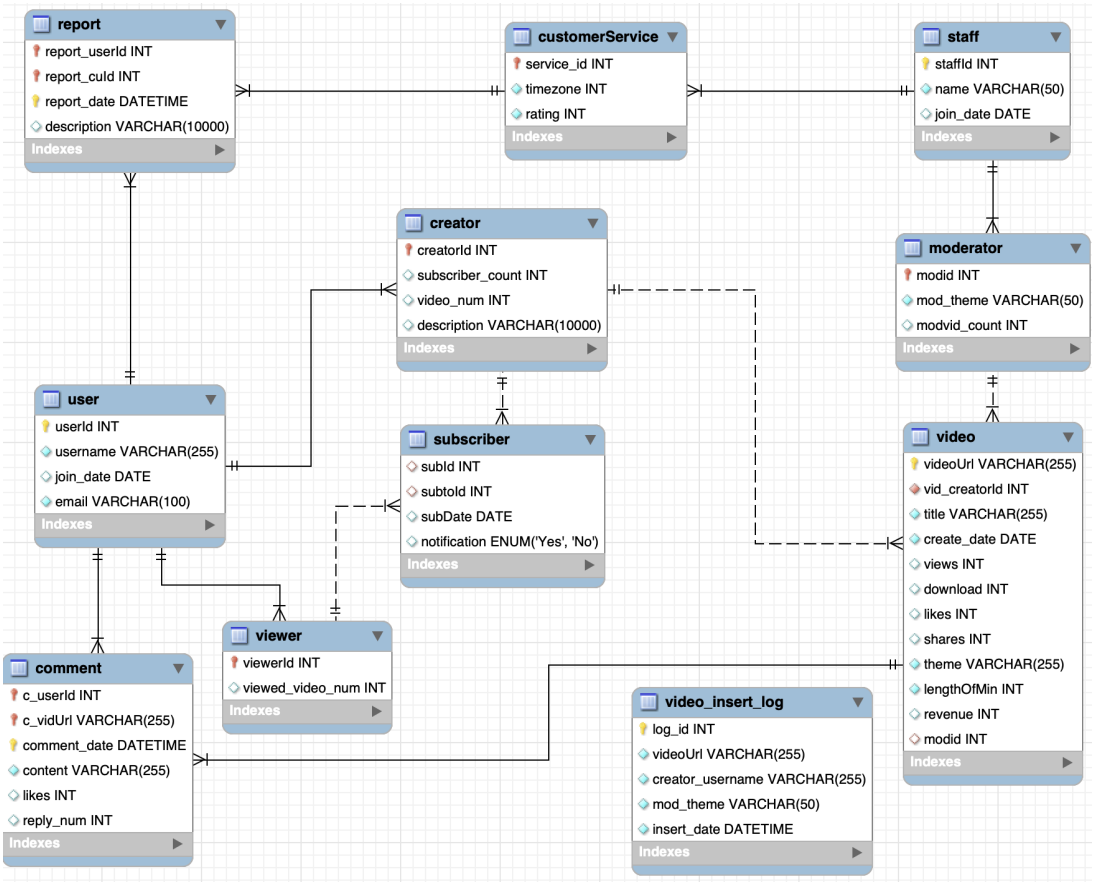
- To enter anything will lead you to enter the username and the password of the database to access the admin option.

- In case of failure or no further input options, rerun the code block below and re-login to the database.

For all functions(options) of the program, please refer to `Activity diagram`

- In some query results, you may see the global `userId`, which is just to let you know the specific userId and username values, so that you can enter correct values among multiple users. This does not mean that the query results of our database are not safe/complete.

Logical Design.



Final User Flow.

Following the instruction of readme, you should be able to see the following guide under our python interactive Graphic-User-Interface(GUI):

As a guest, you can call upon the following functions after entering 1:
1. Obtain a table of Data, which will deliver a dataset
2. Video Revenue comparison, which will deliver an interactive visualization chart
3. show number of video themes, which will deliver the corresponding visualization chart of video themes.

Functions that guests call upon were completely isolated from the MySQL database, both code-wise and operation-wise. This prevented guests from making any modification inside the database; however, the datasets did not be updated with the database as well, unless the codes were run again entirely.

As a user, you can call upon the following functions once your login information has been verified by the database:
4. Obtain up-to-date Data
5. up-to-date Video Revenue Comparison
6. up-to-date video theme
7. Search a video by its url or 8. by its title or 9. by its creator
10. Filter a number of videos by their creation date')
11. Filter all videos by their views
12. insert a new Video using your current user
17. delete a Video under your name
19. make a comment under a designated video.
22. change your video title

Unlike the guests, the users have limited control of the MySQL database. However, they're still locked out of the admin functions with a different identity. To login as a user, you need to input the correct user ID and corresponding username, and the database will validate the information up-to-date. If your input is not recognized, your identity will reset as a guest and you'll need to re-run the GUI. This is to prevent the GUI logs being too long.

As a user, you can also only make changes to your own comment or video. Any other attempt will be refused by the detection system.

As an admin, you can call upon the following functions:
4. Obtain up-to-date Data
5. up-to-date Video Revenue Comparison
6. up-to-date video theme
7. Search videos by its url or 8. by its title or 9. by its creator
10. Filter a number of videos by their creation date
11. Filter all videos by their views
13. insert a new Video
14. create a new user
15. print log
16. delete a video by its url
18. Create a comment under an existing video and an existing user

20. Add a new user
21. Change a video title

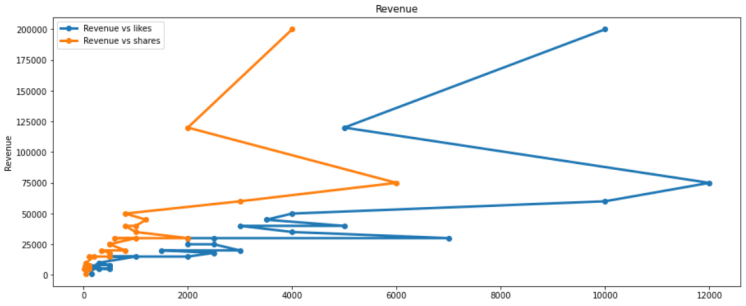
As an admin, you need to login using a specific admin name and password. However, login as an admin does not give you Omni-control of the entire GUI system. Some of the user functions are shared with admin, but some are limited to avoid wrongful input and confusion. Admin has the right to change all the video, comment and make new users in the MySQL database once the correct input is validated.

An illustration of the Graphic-User-Interface(GUI) is shown below:

```
If you are a guest, enter 1; If you are a user, enter 2. Else, enter anything else and you will be redirected to login page.3
Enter username:root
Enter password:abcabc2004
Username:  root Password:  abcabc2004
Welcome, Admin
As an admin, you can call upon the following functions:
 4. Obtain up-to-date Data
 5. up-to-date Video Revenue Comparison
6. up-to-date video theme
 7. Search videos by its url or 8. by its title or 9. by its creator
 10. Filter a number of videos by their creation date
11. Filter all videos by their views
 13. insert a new Video
 14. create a new user
 15. print log
 16. delete a video by its url
18. Create a comment under an existing video and an existing user
 20. Add a new user
 21. Change a video title
Select the number of the functions you want to run from the list above: 5
```

	videoUrl	vid_creatorId	title	create_date	views	download	likes	shares	theme	lengthOfMin	revenue	modid
26	https://www.youtube.com/watch?v=wQXZptfRUsE	13	Interview with a Politician	2022-03-15	5000	3000	150	50	Politics	30	1000	19
4	https://www.youtube.com/watch?v=7dqJTP6l9y0	2	Guitar Lessons for Beginners	2022-04-12	4000	2000	100	20	Music	45	5000	8
6	https://www.youtube.com/watch?v=Az_-GHsX-h0	4	Virtual Reality Gaming	2022-01-10	15000	10000	300	50	Technology	30	5000	10

```
Please enter the name of the first NUMERIC column you want to add to the revenue compare plot: likes
Please enter the name of the second NUMERIC column you want to add to the revenue compare plot: shares
```




```
In [ ]:
```

Lesson Learned.

Technical Expertise Gained:

During the development of this project, we gained valuable technical expertise in working with databases and developing applications that interact with both. Specifically, we became more proficient in SQL programming, including creating tables, views, and stored procedures, and calling them based on Python code.

Insights and Time Management:

Throughout the project, we learned the importance of time management and planning. It is crucial to spend enough time upfront on designing the database schema and planning the application architecture. This helped to avoid potential roadblocks and issues later on during development. we also realized the importance of breaking down large tasks into smaller, more manageable ones, and setting realistic deadlines for each task.

Realized or Contemplated Alternative Designs/ Approaches:
During the development process, we realized that there were several alternative approaches that could have been taken. For example, we could have used a different web framework, such as Django, to build the front-end of the application. We also considered using a NoSQL database, such as MongoDB, instead of MySQL, but ultimately decided to stick with MySQL for consistency with the project requirements.

Code Not Working:
Fortunately, we were able to complete the project without any major issues or code problems. However, there were several minor bugs and issues that we encountered throughout the developing process, which were addressed through careful testing and debugging. Overall, we are proud of the quality of the code we were able to produce and the functionality and features of the final product.

Future work.

Planned uses of the database: The database can be used for additional features such as video recommendations, user analytics, and user preferences. The collections of data can be used to generate insights into user behavior and preferences, which can be used to improve overall user experience of the platform. In addition, the platform can also implement a notification system to notify users of new video releases, comments, and other platform activities.
Selection of dataset: Our current dataset is a generated dummy dataset, however, in the future, we can use a better dataset or a Python-based web scraper to implement a better practical meaning to our database. With a better dataset, we can also implement Machine Learning and/or Neural Network learnings into our Python Front End and extract data from our SQL database, to create an up-to-date tactical Youtube analysis database which will update its data and gives Youtube career advices corresponding to new trends on Youtube.
Potential areas for added functionality: The platform can benefit from additional features such as user profile customization, playlist creation or some more generous machine learning algorithm. The user profile customization will allow users to personalize their profile pages with backgrounds, profile pictures, and other features. Playlist creation will enable users to create personalized playlists of videos they like, which can be shared with other users. Furthermore, the platform can implement a feature to allow users to rate videos, which can be used to rank videos based on popularity and interest.
Another potential area for added functionality is the implementation of recommendation systems. The recommendation system will analyze users' viewing history and preferences to recommend other relevant videos they might be interested in. This can increase user engagement and retention on the platform.

Another feature that can be added is the integration of social media sharing, which will allow users to share videos they like on their social media profiles, leading to increased visibility and traffic to the platform.

Finally, the platform can benefit from implementing more advanced machine learning algorithms such as image recognition and natural language processing. Image recognition can be used to automatically tag and categorize videos, making it easier for users to search for videos they are interested in. Natural language processing can be used to automatically generate video descriptions and transcriptions, making it easier for users to understand and access the content.