Youtube Clone Backend Logic & Steps

1. **Setup Project**
   - Create new projects **ytprj**
   - Create new app (core)
   - Install Apps in Settings.py
   - Configure Templates
   - Configure Static and Media Files in settings and Urls.py
   - Create a new view, urls and template then runserver.

2. **Configure Admin Page, Superuser and Jazzmin**
   - Install Jazzmin (pip install django-jazzmin)
   - Add jazzmin in **INSTALLED_APPS**
   - Add Jazzmin Config Code in Settings.py
   - Create Superuser
   - Login To Admin Section

3. **Custom User Model**
   - Create custom "*Class User(models.Model)*:" in **models.py**
   - Add **AUTH_USER_MODEL = 'userauths.User'** in settings.py

4. **Create Video Model**
   - Create video models with all field
   - Create uuid using shortuuid package (pip install shortuuid)

5. **Create View to Display All Videos from DB**
   - Try filtering videos by only **public** to be displayed in the home page

6. **Create Video Detail Page**

- Use the UUID to get different videos

7. Create Channel Model

- Create channel models with all field
- Make sure it's oneToOneField

8. Show Related Video

- Show related videos based on what a user is watching

9. Comment Feature (Ajax)

- Create model for comment
- Install in the admin.py
- Get all model in view.py videoDetail
- Display comment in video
- Count comment *{{comment.count}}*
- Create form to add comment
- Create ajax function to get and prepend comment
- Create views to process comment and
- Create url to route comment

10. Delete Comment Feature (Ajax)

- Create view to delete comment
- Create URL to connect to delete_commet function
- Create ajax function to remove comment asynchronously without refreshing the page
- Make only the logged in user to be able to delete comment

11. Edit Comment Feature

- Create view to edit comment
- Create URL to connect to edit_commet function
- Create ajax function to edit comment asynchronously without refreshing the page
- Create new input field where previous comment would be shown then use javascript

- Make only the logged in user to be able to edit comment

12. Subscribe Feature

- Create function to add subscribers
- Create another function to load all channel subs
- Add views to urls.py file
- Create button in html with id to connect to axios
- Use axios to add sub and load sub in real time

13. Like Video Feature

- Create function to like video
- Create another function to load all video likes
- Add views to urls.py file
- Create button in html with id to connect to axios
- Use axios to add likes and load likes in real time

14. Channel Profile

- Create function to get channel profile
- Add the new function to **urls.py**
- Display channel Details
- Configure the subscribe button (import axios cdn)
- Get Featured Video and display in section

15. Channel Videos

- Create Channel in another section
- Create function to get channel profile
- Add the new function to **urls.py**
- Display Video in Video Section

16. Channel About

- Create new page for channel about
- Create logic to get channel total views

17. Channel Community

- Create Community Models
- Register model in Admin.py
- Create view, template and urls to Display Community Posts

18. Create Community Comment (No Ajax)

- Create community detail page
- Lists out all comment from DB to comment Section
- Create logic to add comment without using Ajax in the detail view

19. Delete Community Comment (No Ajax)

- Create new function to Delete Comment
- Create message alert

20. Like Community Comment (No Ajax)

- Create new function to like Comment
- Create message alert

21. Save Video for Later

- Create new view function to save Video
- Configure url
- Add url to save button

22. Register Feature

- Create new function to register using
- Create Forms.py (userCreationForm)

- Configure Template and URL

23. Login Feature

- Create new function to login using
- Create query in views.py
- Configure template to grab the login details

24. Logout Feature

- Create new function to logout using django logout package
- Create query in views.py
- Configure urls to view

25. Upload Video Feature

- Create forms in forms.py
- Create function to upload video in views.py
- Create URL
- Configure Forms in template
- Test

26. Edit Video Feature

- Create function to edit video in views.py
- Create URL
- Configure Forms in template
- Test

27. Create Community Post

- Create forms in forms.py
- Create function to upload video in views.py
- Create URL

- Configure Forms in template
- Test

28. Edit Community Post Feature

- Create function to edit video in views.py
- Create URL
- Test

29. Delete Community Post Feature

- Create function to edit video in views.py
- Create URL
- Test

30. Search Feature

- Create function to get URL parameter and query
- Create URL's py
- Add URL to search form

31. Tags Video Page

-