

Introduction

A movie interaction chatbot

 a platform where the users are able to find movie diversity and inclusion.

 obtains multiple functions e.g., input comments, rating, click on "like" or "dislike" button etc.

help us to consolidate the data and provide recommendation.

Cloud Services Involved



Google Cloud

Cloud hosting









Database hosting

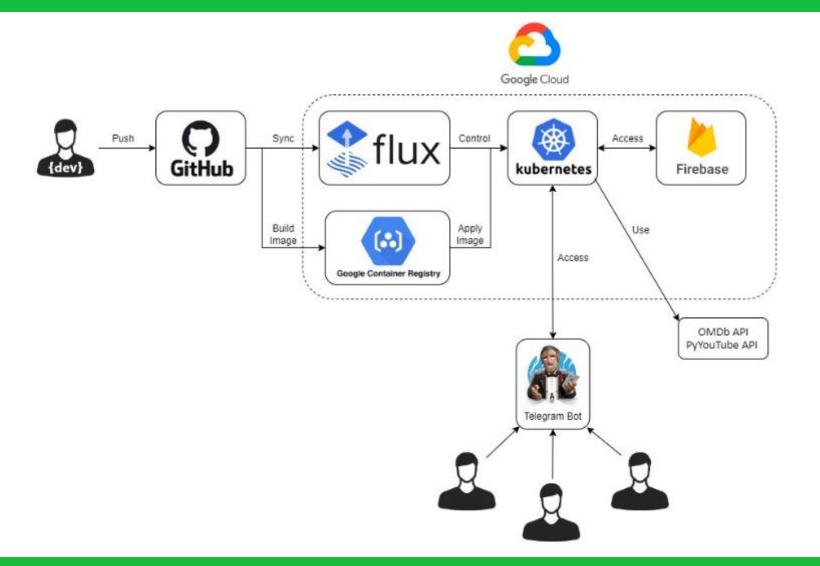


GitOps CICD

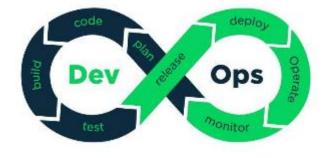


Coding repository

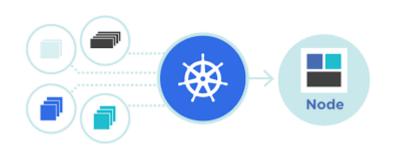
Cloud Infrastructure



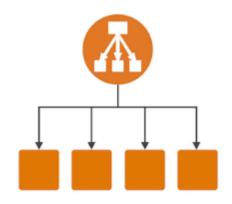
DevOps



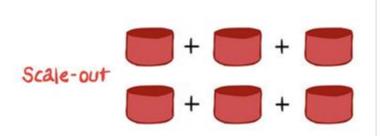
Container Orchestration



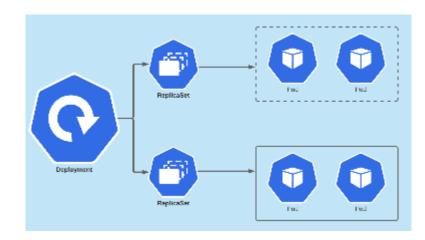
Load Balance



Scalability



Rolling update



Security Concern

1. Store Password and Key in Git Secret

- 2. Perform the vulnerability checking for Python package
- 3. Use NoSQL instead of SQL database
- 4. Clear the redundancy and legacy coding

Chatbot User Interface



Help information

Command: '/help'

Function: Provide a list of available commands and function description



Search movie

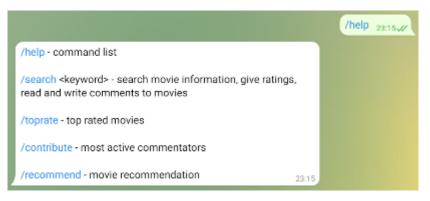
Command: '/search'

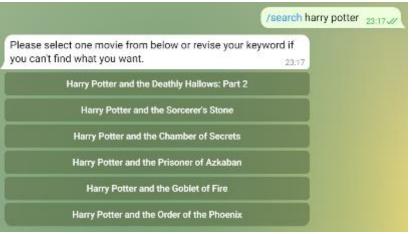
Function: Search with the name of the movie as a keyword and receive a list of six most relevant movies.

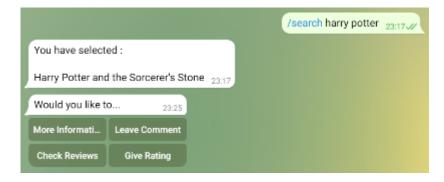


Action: Click on one of the movie for selection

Function: Prompt menu buttons to guide user to take further actions regarding to the selected movie



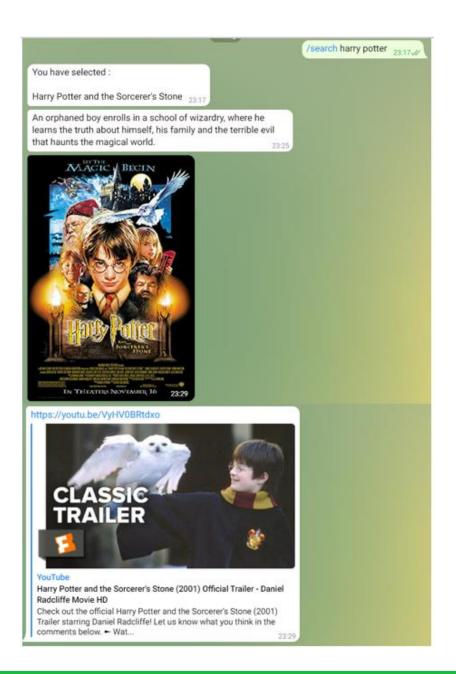






Get information of the movie

Once the user clicks on 'More Information', the bot will show the information of the movies, including plot, poster and trailer from YouTube.





Check comments of the movie from other users

Action: lick on the 'Check Reviews' button.

Function: List the latest 3 comments for users' review



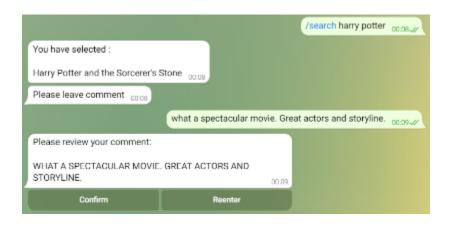




Leave comments

Action: Click on the 'Leave Comment' button

Function: Repeat the comment and ask the user for confirmation.





Action: Click on 'Re-enter' button.

Function: Allow the user to revise the comment.



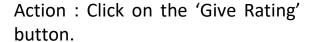


Action: Click on the 'Confirm' button.

Function: Upload the comment to the database.

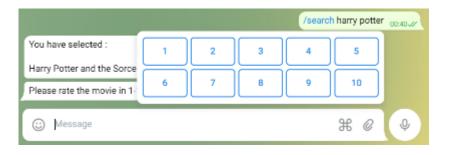
what a spectacular movie. Great actors, director and storyline Comment Shared! 00:14

Give Rating



Function: Prompt a number keyboard to let the user rate the movie on a scale from 1 to 10.

* Similar to the 'Leave Comment' function, users could reconfirm or re-enter.











Check the top rated movie

Command: '/toprate'

Function: Display the top 10 highest rating movies, based on the average ratings from other users.





Check the top contributions

Command: '/contribute'

Function: Display the top 3 most active users, based on their amount of ratings and comments provided.

Congratulation to our active users!

user_BRIAN: 3 comment & 4 rating

user_WOODY: 2 comment & 2 rating

O1:07

Recommend a movie



Command: '/recommend'

Function: Display the information, including plot, poster and trailer from YouTube, of a movie randomly picked from the top 10 highest rating list.



Database - Firebase RealTime Database

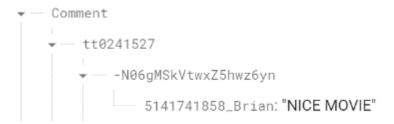
Two major tables: 1) Comment 2) Rating.

- First child layer
- Movie id (e.g. tt0241527) defined in OMDB, which is used for uniquely identifying each movie.

- Second child layer
- Database unique key (e.g. N06rKZZSCGUnfmz80ay) assigned by Firebase when pushing the data to the database.

- Third child layer
- Key: a composite string (telegram user id + first name of the user)
- Value: the comments or ratings of the users.

Comment Table



Rating Table



Future Development



Expand Database

expand our database and become more accurately.



Include non-mainstream movies



Collaborate with cinemas to provide a ticket booking system

can redirect our users to different cinema's websites



