2 Vejledning til formalia 1

2.1 Forside . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

2.2 Resumé/abstract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

2.3 Indholdsfortegnelse . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

### Kildehenvisninger og kildeliste

Dokumentation for twitchio kan findes på deres officielle hjemmeside: <https://twitchio.readthedocs.io/en/latest/>

Dokumentation for asyncio, som er en del af standardbiblioteket i Python, kan findes på Python's officielle dokumentations hjemmeside: <https://docs.python.org/3/library/asyncio.html>

Dokumentation for aiomysql kan findes på deres officielle GitHub-side: <https://github.com/aio-libs/aiomysql#documentation>

# Indledning

Formålet med dette projekt er at udvikle twitch-chatbot for at mindske mængden af chatbots i twitch streamers chat’s. Programmet er skevet og testet i python 3.9 og 3.10 (Foundation, Python Software Foundation, 2020), (Foundation, Python Software Foundation, 2021) på Windows 11 og gør brug af disse libaries twitchio.ext [commands, routines] (PythonistaGuild, 2021), aiomysql (Revision, 2015)

# Funktionsbeskrivelse

## chatbot bootloader

Chatbot bootloader funktionen bruges til at starte chatbotten op hvor efter den specificere hvilken konto den bruger og hvilke kanaler som den er aktiv på og så definere den prefixet til at trigger vores 5 commands som er lavet til dette projekt

## giveaway

funktionen til at lave giveaways som består af en join start og stop command

* !Join
* !Start
* !Stop

## shoutout

funktion til at shoutout en streamer hvor man foreksempel bliver "raided" på twitch

## discord

en funktion til at gøre det let for seeren at komme ind på en streamers discord for blandt andet at være med i spil etc

# Problemformulering

Streaming af videoindhold er blevet utroligt populært i dagens digitale verden, og millioner af mennesker over hele verden følger nu dagligt med på platforme som Twitch, YouTube og Facebook Gaming. For streamere kan interaktion og engagement med seerne være afgørende for deres succes, men det kan også være en udfordring at håndtere den store mængde af kommentarer og spørgsmål, som kan strømme ind i realtid.

Traditionelt set har streamere svaret på seernes spørgsmål i chatten, men dette kan være en tidskrævende proces, og det kan være svært at svare på alle spørgsmål. Dette kan føre til en nedgang i engagementet med seerne og i sidste ende en negativ påvirkning af deres vækst som en streamer.

En mulig løsning på dette problem kan være at implementere en chatbot, som kan besvare spørgsmål og give information til seerne. Chatbots kan også tilbyde yderligere funktionalitet, såsom at skabe interaktive spil og konkurrencer, der kan engagere seerne og skabe en mere spændende seeroplevelse. Ved at undersøge, hvordan chatbots kan hjælpe streamere med at forbedre deres interaktion og engagement med seerne, kan vi finde en løsning på udfordringerne og skabe en mere tilfredsstillende seeroplevelse for alle parter.

derfor ville følgende krav være interessante

* køre stabilt
* identificere commands
* respondere til commands
* hjælpe streameren

# Dokumentation

Probot - init() + event\_ready()

Koden er en del af en klasse for at oprette en chatbot.

\_\_init\_\_-metoden initialiserer bot'en med adgangskoden og prefixet til dens kommandoer. Den opretter også en variabel giveaway bool og initialiserer den til False. Endelig kører den en asynkron funktion setup\_database(), som opretter forbindelse til en database og opretter en tabel for deltagere, hvis den ikke allerede eksisterer.

event\_ready-metoden kaldes, når bot'en er klar til at chatte og bruge kommandoer. Den printer bot'ens navn og bruger-id.

shoutout()

Denne kode er en del af en chatbot, der er udviklet til at interagere med seerne på en streamingplatform. Koden indeholder en kommando, der aktiveres, når en seer skriver '!so' efterfulgt af et brugernavn. Når denne kommando udføres, sender chatbotten en besked til chatten, der opfordrer seerne til at tjekke ud den streamer, der er nævnt. Beskeden indeholder en tekst, der inkluderer navnet på den nævnte streamer og et link til deres kanal.

Denne funktion kan hjælpe streamere med at fremme andre streamere, de ønsker at støtte, og skabe et mere samarbejdsvilligt og inklusivt streaming-miljø. Ved at tilbyde en hurtig og nem måde at dele information om andre streamere, kan chatbotten også bidrage til at øge seernes engagement og interesse for streamingplatformen som helhed.

Det er vigtigt at bemærke, at denne kode kun er en del af en større chatbot-applikation og ikke kan fungere selvstændigt.

discord()

Dette er et forslag på kode, der kan bruges til at implementere en Twitch chatbot, der svarer på en bestemt kommando. Når en bruger skriver "!discord" i Twitch chatten, vil chatbotten reagere ved at sende en besked tilbage med en invitation til en Discord-kanal.

Funktionen defineres ved hjælp af "@commands.command()", som er en dekorator, der fortæller Twitch-biblioteket, at denne funktion skal aktiveres, når der modtages en bestemt kommando i chatten. I dette tilfælde vil kommandoen være "discord".

Funktionen starter med at udskrive "test" i konsollen ved hjælp af "print()" funktionen. Dette er en simpel måde at teste, om funktionen udføres korrekt under kodningsprocessen.

Derefter bruger funktionen "ctx.send()" til at sende en besked tilbage til chatten med invitationen til Discord-kanalen. "ctx" henviser til "Context" objektet, som indeholder information om den modtagne besked, herunder afsenderen og beskedteksten. Funktionen "send()" bruges til at sende en besked tilbage til chatten, og teksten "https://discord.gg/a3jHXau" er det faktiske indhold af beskeden.

giveaway()

Denne kodeblok er en del af Twitch chatbotten, der er designet til at håndtere giveaways under streaming sessioner. Koden indeholder en funktion, der håndterer kommandoen "!giveaway" i Twitch chat. Når funktionen kaldes, vil den først sørge for at oprette forbindelse til en database, der opbevarer information om giveaways.

Herefter vil koden tjekke en boolean værdi kaldet "giveaway\_bool", som indikerer, om der allerede er en giveaway i gang. Hvis værdien er sand, vil chatbotten sende en besked tilbage i chatten, der fortæller, at der allerede er en giveaway i gang. Hvis værdien er falsk, vil koden tjekke, om personen, der har skrevet kommandoen i chatten, er en administrator ved at sammenligne deres brugernavn med listen "cfg.admin\_n\_list".

Hvis personen er en administrator, vil chatbotten sende en besked i chatten, der fortæller, at der er startet en giveaway. Koden vil også ændre værdien af "giveaway\_bool" til sand, således at andre brugere ikke kan starte en giveaway på samme tid.

Alt i alt viser denne kodeblok en simpel måde at implementere en Twitch chatbot, der kan håndtere giveaways på en effektiv måde.

endgiveaway()

Koden du har delt, ser ud til at være en del af en Twitch-bot, der kan hjælpe streamere med at håndtere en giveaway. Koden indeholder en funktion, der kaldes "endgiveaway", som ser ud til at være en kommando, som bot'en vil reagere på, hvis den modtager den fra en bruger på Twitch.

Funktionen har nogle betingelser for at kunne udføres. Først og fremmest skal personen, der kalder kommandoen, være på en liste over admin-brugere i en konfigurationsfil. Hvis personen ikke er på listen over admin-brugere, vil bot'en ikke udføre kommandoen.

Derefter tjekker funktionen, om der er en giveaway i gang. Hvis der ikke er en aktiv giveaway, vil bot'en sende en besked til chat, der informerer om dette. Hvis der er en aktiv giveaway, vil bot'en vælge en vinder fra listen over deltagere og sende en besked i chat, der annoncerer vinderen og gratulerer dem.

Til sidst nulstiller funktionen giveaway-tilstanden og nulstiller listen over vindere, så bot'en er klar til at håndtere en ny giveaway.

Alt i alt ser koden simpel ud, men en effektiv løsning til at håndtere giveaways for streamere på Twitch, og den kan hjælpe med at automatisere processen og spare tid for streameren.

join\_giveaway()

Funktionen join\_giveaway er en kommando, som en bruger kan aktivere ved at skrive !join i chatten på Twitch. Når denne kommando aktiveres, vil navnet på den bruger, der har aktiveret kommandoen, blive tilføjet som en deltager til en konkurrence eller en giveaway.

Funktionen add\_participant bliver kaldt af join\_giveaway-funktionen for at tilføje den aktuelle bruger til databasen med deltagere. Denne funktion åbner forbindelse til en MySQL-database og søger efter brugernavnet på den pågældende deltager i tabellen participants. Hvis deltageren allerede er til stede i tabellen, vil funktionen ikke gøre noget. Hvis deltageren ikke er til stede i tabellen, vil funktionen tilføje deltagernes navn til tabellen.

#### Test

Det er vigtigt for mig at understrege, at det er en god praksis at teste applikationen grundigt, inden jeg frigiver den til et større publikum. Ved at teste applikationen live, har jeg sikret, at applikationen fungerer korrekt i en realistisk situation. Dette har også hjulpet mig med at identificere og rette eventuelle fejl eller problemer, der måtte opstå under testen.

Når det kommer til udvikling af chatbot-applikationer, er det afgørende, at min applikation kan håndtere et stort antal brugere, som kan interagere med bot'en på samme tid. Derfor er det godt, at alle kommandoer fungerer, da dette sikrer, at applikationen kan reagere på brugernes anmodninger effektivt og korrekt.

Den rigtige funktionalitet af giveaway-funktionen er også vigtig, da det er en af de funktioner, som kan tiltrække mange brugere til min kanal. Hvis der opstår problemer med giveaway-funktionen, kan det resultere i tab af brugere og en negativ oplevelse for dem, som deltager i konkurrencen. Derfor er det godt, at min giveaway-funktion fungerede som den skulle, og at den trak en tilfældig vinder.

## Kildekode

Kildekode kan findes her <https://github.com/itsteutatas/Pro-eksamensprojekt>