DISCORD BOT

Major project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Harshit Gautam (181426)

Advait Agnihotri (181282)

UNDER THE SUPERVISION OF

Dr. Hari Singh

Department of Computer Science & Engineering and Information Technology

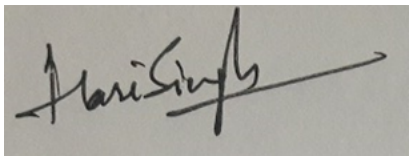
Jaypee University of Information Technology,

Waknaghat, 173234, Himachal Pradesh, INDIA

DECLARATION

I hereby declare that this project has been done by me under the supervision of **Dr. Hari Singh,** Assistant Professor (SG), Computer Science & Engineering and Information Technology, Jaypee University of Information Technology. I also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:



Dr. Hari Singh

Assistant Professor (SG)

Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat,

Submitted by:

Harshit Gautam

(181426)

Advait Agnihotri

(181282)

Computer Science & Engineering Department

Jaypee University of Information Technology

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to Almighty God for His divine blessing makes it possible to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor **Dr. Hari Singh,** Assistant Professor (SG), Department of CSE, Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of my supervisor in the field of "Task-Oriented Bots" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr. Hari Singh,** Assistant Professor (SG), Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Harshit Gautam

(181426)

Advait Agnihotri

(181282)

TABLE OF CONTENT

LIST OF FIGURES

ABSTRACT

Discord is ViOP(voice over internet), a messaging platform designed in order to develop communities and provide them with all the basic necessary features like (text, video, and audio). Along with low latency in audio and video quality.

The main highlight of this app is its ability to compete and exceed top streaming platforms which is handy. The structure of discord i.e. division in terms of server, channels, and voice channel makes it very user-friendly. Top esports organization streamers, gamers, and top influencers are making their servers to make their engagement more strong.

Bots are an integral part of discord which provide users to enhance and maintain their servers and perform various tasks assigned by the admins. Bots are created with the help of registering for the bot at the discord developer portal and developing the bot accordingly for their specific use.

The Discord API is based on two separate APIs which are WebSocket API and Rest API. The Websocket API is used to receive events from the discord in real-time. The bot uses this API to request a connection, manage voice connections and perform fundamental tasks, whereas the Rest API is responsible to perform actions in discord.

It acts as a query for information and is responsible to perform various actions like sending messages, kicking/banning the users, and updating user information. Our bot (HAM) is a discord bot that comes with multiple functions in terms of moderation, chat messaging, music playing, etc. It can perform multiple tasks in the discord server.

Along with that, it recognizes when a user enters the server and authenticates via user info whether the user is another bot or human. HAM is currently equipped with multiple functions like getting sentiment analysis on users' messages to playing songs from Youtube, all directly from within the discord. These all commands are based on various python libraries including youtube_dl, urllib, etc.

CHAPTER 01 : INTRODUCTION

1.1 Introduction

To understand the Discord Bot first we need to understand what discord is

"A complete Voice and text chat for gamers that's free, secure, and works on both desktop and phone at ultra-low latency rate. Save ur bucks on TeamSpeak servers and hassling with Skype. Simplify your life."

"Almost every top influencer around the globe now has their own discord servers and channels for their engagement and endorsement ."

F.1 Discord logo illustration

Discord is a **chat app**?

Yes, it is but it's more convenient than any chat application and well structured.

It provides us the functionality to build communities, groups, or anything else on a mass level. You can connect to each other via a text/voice/video chat and also can have a one-to-one interaction.

Discord is becoming more and more popular. That way automation prohibition of inappropriate users responding to user demands is essential for the community to prosper and grow. Appearance and acts like a user react automatically to events, discord commands are called bot users. discord for bot users (or bots only), there are almost endless possibilities of applications. For example, let's say you manage a new Discord Guild and Users Join for the first time. Excited, you can contact it personally to welcome them to users and your community. You can also talk to them or ask them about your channel image. Users feel welcome and Enjoy the discussions taking place in the guild and they in turn invite their friends. Over time, your community has grown so large that it is no longer feasible to personally approach each new member other than you, still want to send something to recognize them as a new guild member. In the case of bots, It is possible to automatically respond to new members joining the guild. You can customize it to control contextual behavior and how it works and Interact with each new user. Users can now copy or paste other image Videos, but have no ability to fetch them automatically.

The one on the text channel has been implemented yet. Discord provides a connection with Spotify, YouTube, Steam, etc., but not Reddit. is not A program to help you find all the news, votes, and facts,

Images are automatically displayed in various text channels. A System of additional public messages to be triggered Getting in and out of the server helps with the personalization system. We don't offer as much discord work on that text channel. The bot

is active everywhere on Discord, but all Discord bots are made for you

Play music only for specific reasons, such as Generate YouTube, Spotify, tokens, or RPGs, or text game. In a world where communication is important For many, it's important for the platform

Providing social and communicative functions Interesting and interactive. Discord is a group chat platform originally developed for gamers. Since then, it has become a popular platform for many species.

Of the community. Discord is divided into servers Unique members, themes, rules,

and channels. Discord also allows users to talk and Not only video chat, but also live stream games, etc. A program from their computer. Basically, Discord is set up to allow members to do this. Message to each other. Each community If you are familiar with "server." Slack, you can think of it as follows: Of it is a less formal version of this app. The server is Full of text channels (you can type and speak To others) and audio channels (where you can) Voice chat with others). You can also share videos, photos, internet links, music, etc. Any server There are usually several channels, each of which is Dedicated to another topic or has different rules. For example, there may be a channel to talk to About another for games and general chat The other is a picture of a cat. The possibility is infinite. It's free to start your own server and free to join Other servers. There are thousands of different discord servers, each one dedicated to a different topic. If you have an interest in something, there's a good chance you can find a Discord server for it. This is especially true for video games, which make up the bulk of Discord's most popular servers Discord is a chat app for gamers with features such as text, images, video, and audio communication.

The more formal definition of Discord is a freeware internet telephony application and digital distribution platform for the video gaming community with the features mentioned above.

Today, Discord bots are one of Discord's best features, allowing you to create automated users to perform all sorts of functions in Discord. This is another unique feature that sets Discord apart from other chat apps. The bot helps Discord get specific data, deliver notifications, schedule events, and play music for users on the server.

What is a **Discord Server**?

*A server is basically the main community group

*A channel is where you can chat and communicate with the other members. You can drop your messages in a channel like a normal chat app. In addition to that, there are voice channels where you can

1. talk like a **normal phone conversation**. :)

2. Listen to the music using the **Discord Bot.**

3. You can use voice channel overlay in any application at **ultra-low latency.**

Discord Server is a PC program that oversees admittance to unified assets inside an organization of local gatherings with rooms and channels. This channel is where you can begin visiting Discord. Very much like some other visit application, you can put a message, picture, or voice talk into your Discord channel.

What is a **Discord bot**?

A discord bot is an automated program coded in order to perform various functions around the discord servers. It provides you with more functionality than a normal user can along with that it helps to reduce the workload from human moderators and perform repetitive tasks.

A **discord bot** can be a:

1. **Chat Bot** (a bot that chats & tries to behave like a human)

2. **Moderation Bot** (it will moderate your server; kick, mute or ban members, if someone tries to create spam or anything similar likewise)

3. **Music Bot** (It plays music from the youtube or any other media platform)

Bots are generally used for:

1. Management of Servers (Kick, Ban Mute Members) via bot commands.

2. Control the levels of spam in the servers using sentiment analysis on users' messages.

3. Play music right from within the discord server.

4. View information regarding the discord servers or users.

Discord bots are one of the most useful features of communication platforms. These are AIs that marketers and business owners can use to perform various automated tasks on their servers.

Building a community is never easy, but doing it on Discord is a great way to give people a sense of exclusivity. If you already have an audience, you can benefit from this strategy.

In a secure and closed community environment, you can provide tips, record videos, host your life, and teach others everything you know. When people feel they're getting something they can't get anywhere else, they're more likely to share it with others and pay more for information in return.

Discord bots make it easy to build an avid community. You can use them to welcome new members to the server, relax communication between members, and ban people who

refuse to follow the rules.

You can use them to add memes, games, music, and other fun content to your server to make it more fun for your members.

A Discord bot is something that will assist you with making getting things done in your organization more straightforward.

It gives different capacities going from playing music to sending youtube links and playing fun games like tictactoe.

It can likewise help you in administrating your disagreement society when you're away.

You can either utilize an outsider bot or make yourself one by coding it with any programming dialects, for example, Java, Python, C++, and so forth.

A Discord server also known as an organization resembles a gathering however with additional functionalities that make it in excess of a gathering.

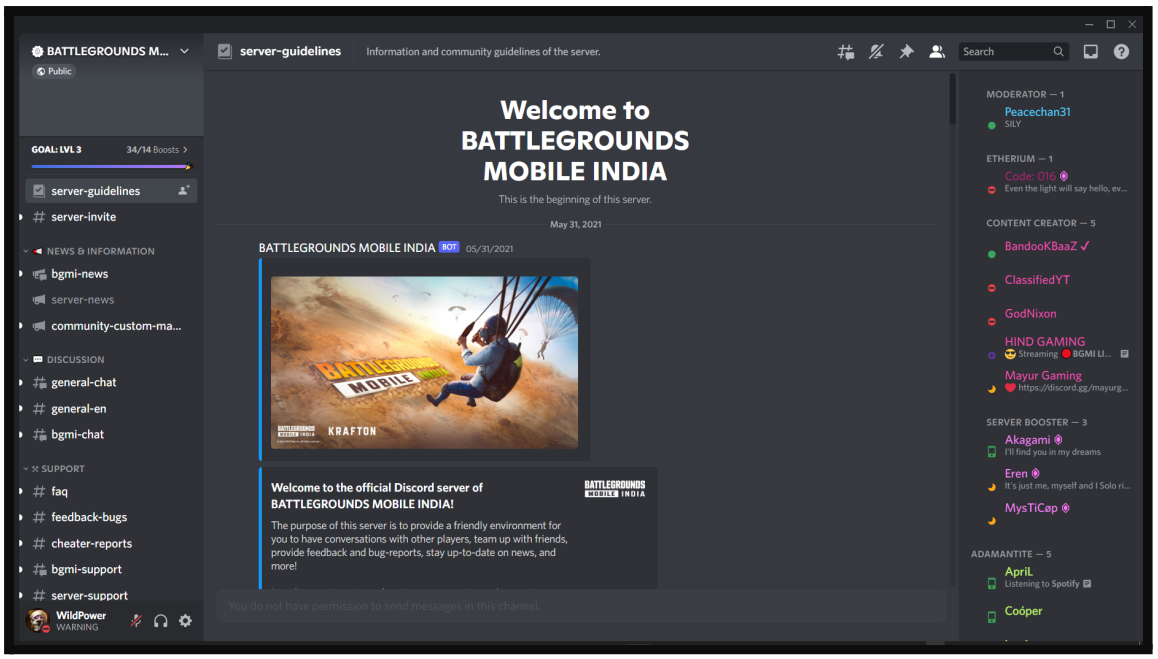
It plays parts, classifications, custom emoticon choices, and different choices!

Discord has its own vocabulary. You may hear teens and students use these words when speaking on Discord.

Server: The server is Discord's space.

They are created by a specific community or group of friends. Most of the servers are small and invitation-only. Some large servers are open to the public.

All users can start a new server for free and invite their friends.



F.2 Discord server example image

Channels: Discord servers are organized into text and audio channels. These are usually specific to a particular topic and can have different rules.

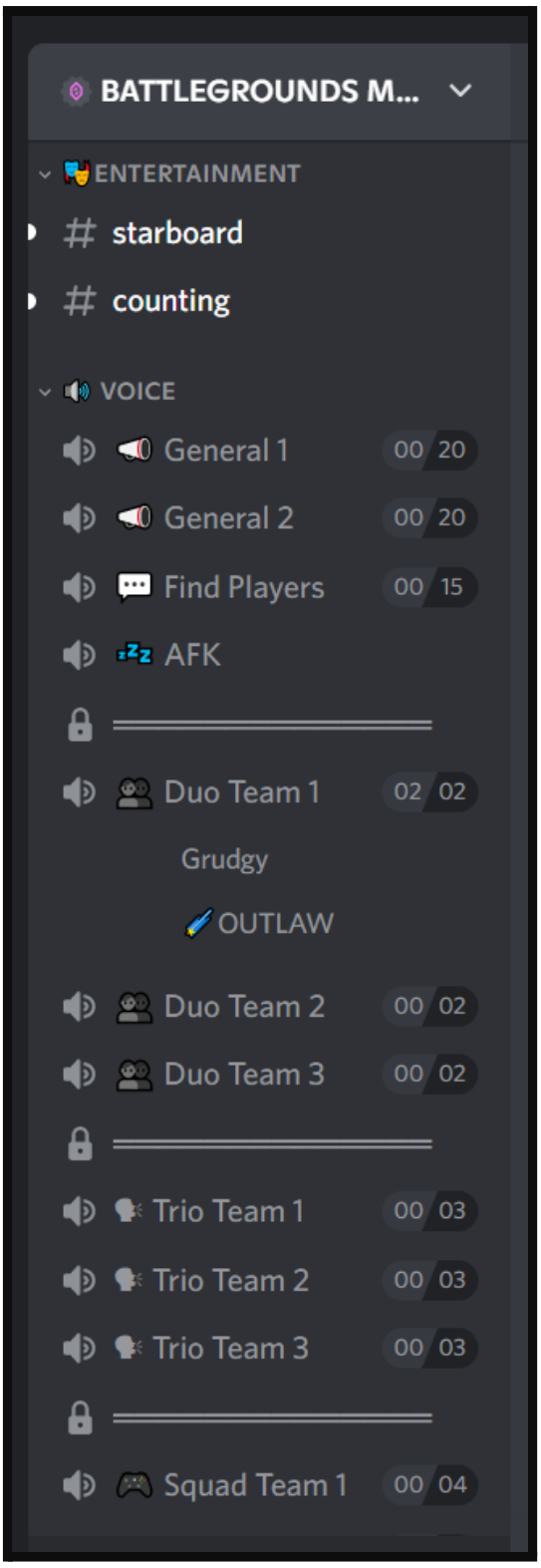
Channels in discord are of mainly two types :

1. Text channel

2. Voice channel

Text channels: allow users to post messages, upload files, and share images for others to see at any time.

Voice channels allow users to connect via real-time voice or video calls and share their screens with friends. This is called GoLive. DM and GDM.



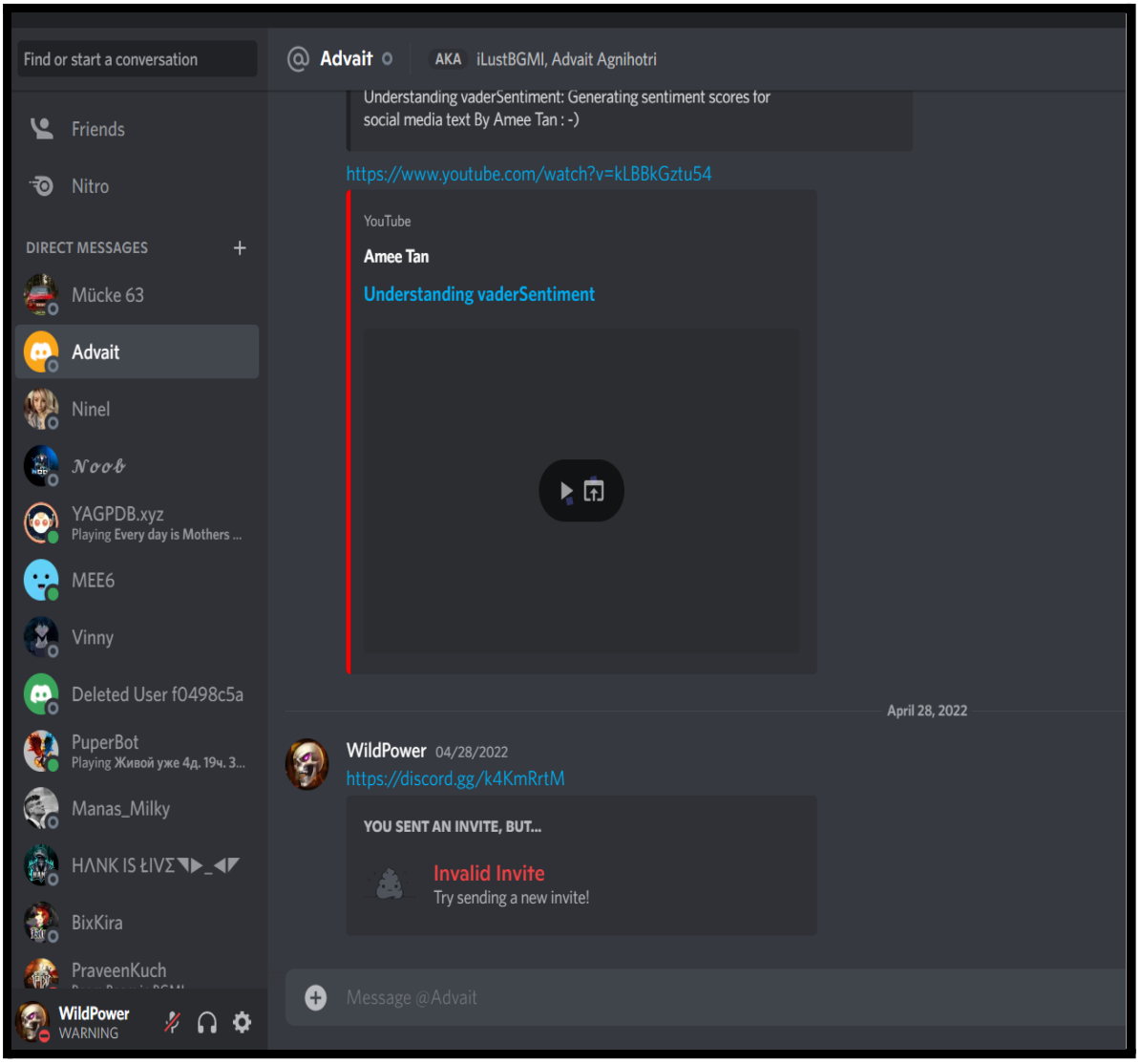
F.3 Discord voice channels example image

Users can send private messages to other users as direct messages **(DM)**, and initiate voice and video calls.

Most DMs are one-to-one, but users have the option of inviting up to 9 people to a conversation and creating a private group DM of up to 10 people.

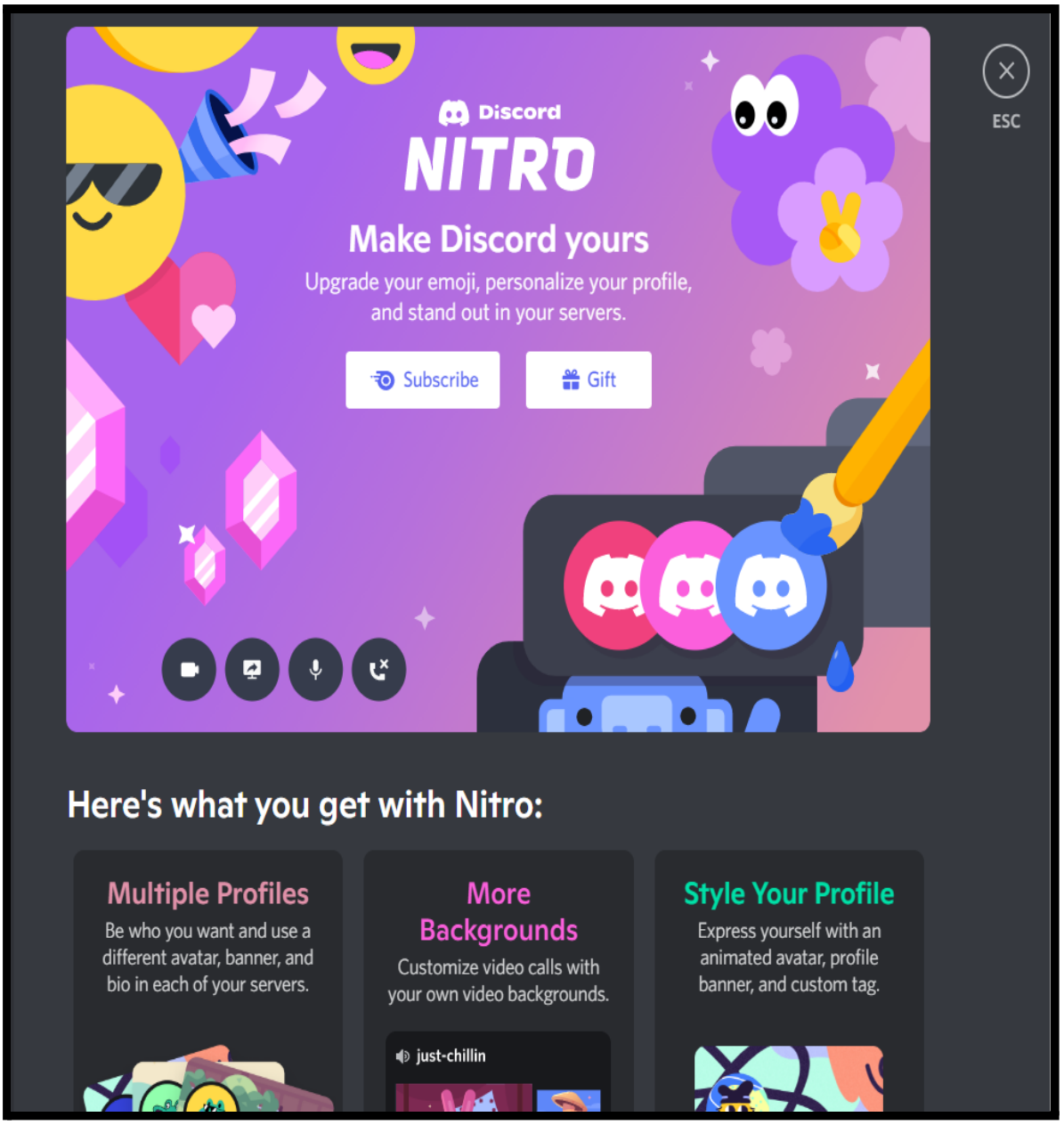
Group DM is not open to the public and requires an invitation from someone in the group to join.

Get up and running: Users can share their screens with anyone on the server or in the DM.



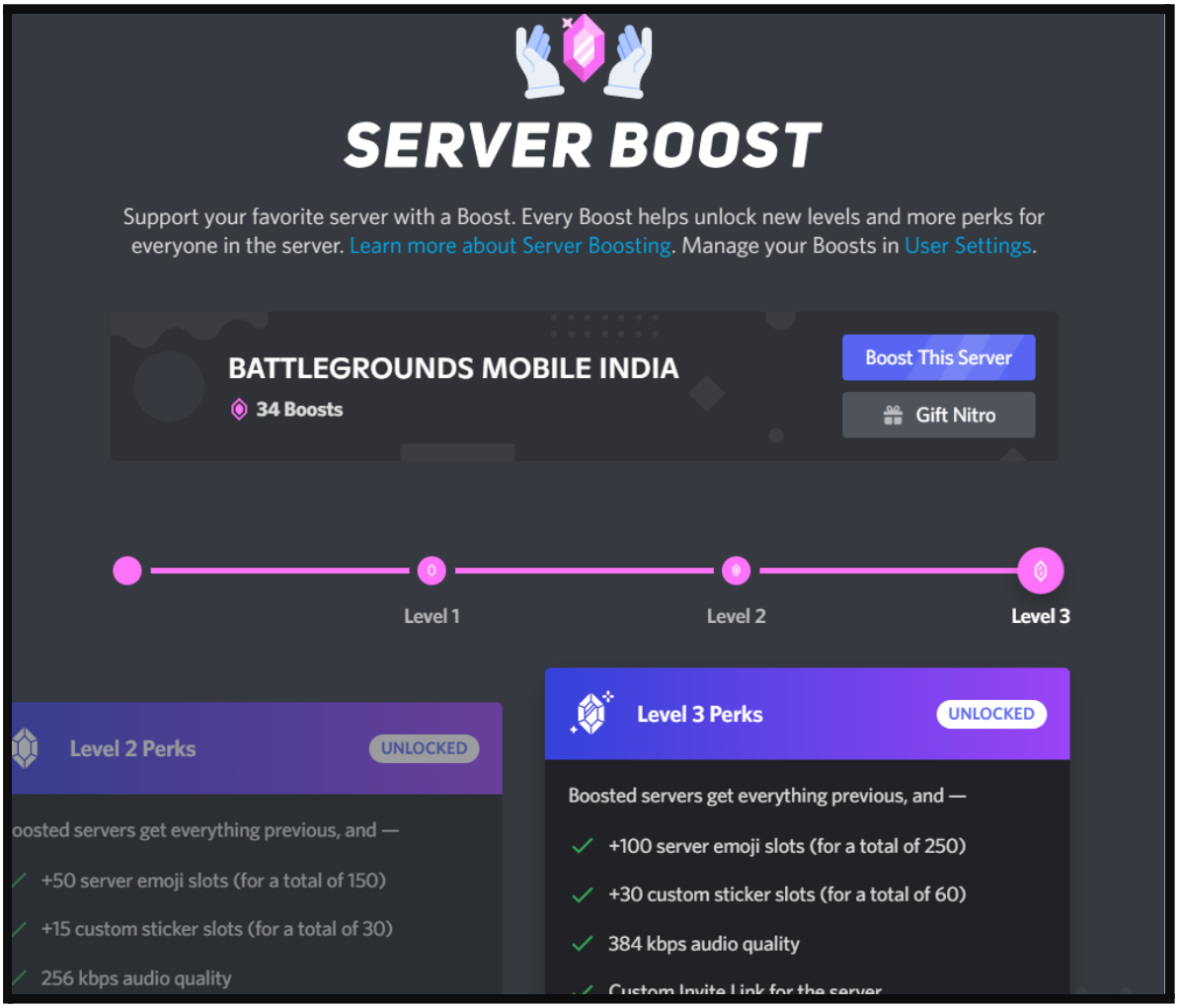
F.4 Discord DM example image

Nitro: Nitro is Discord's premium subscription service. Nitro offers subscribers the following special benefits: This includes options to customize Discord tags, the ability to use custom emotes on each server, higher file upload limits, and discounted server boosts.



F.5 Discord Nitro subscription page image

Server Boost: If your teen is a big fan of the community, they may want to boost the community's (or their own) servers. Like Nitro, Server Boost provides servers with special benefits such as more custom emotes, better video and audio quality, and the ability to set custom invitation links. Server Boost can be purchased with Nitro or individually.

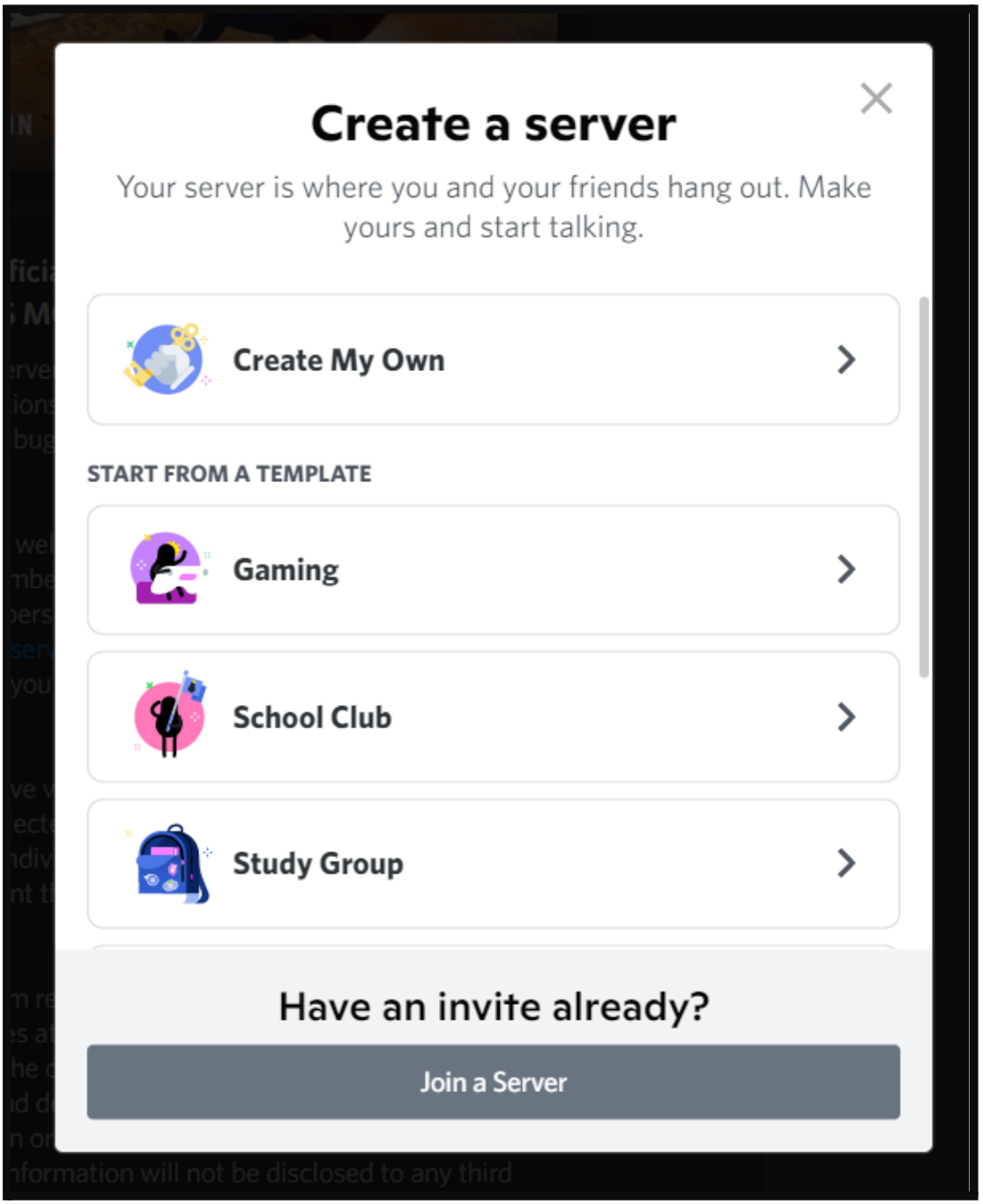


F.6 Discord Server boost page image

Student Hubs: Student Discord Hubs allow students to verify their Discord account using their official student email address and unlock access to their school's student-only hubs.

Within the hub, you can connect with other validated students, find servers for research groups and classes, and share your server with other students.

Hub does not partner with or manage the school or school staff. The servers in the hub are run by students, but may also include non-student servers.



F.7 Discord Server creation hub

It's important to know which Discord bot you want to add. Some bots can improve the server and provide a better community so that users are with you and want to interact with you. The wrong ones could turn people away or create a poor environment.

1.2 Problem Statement :

Discord is being used as the new social media platform for top content creators and YouTubers for its easy and interactive UI.

Bots act as the backbone as they help in performing various tasks assigned by the admins with much precision.

Bots are costly and require subscriptions for use on large servers. Most of the bots offer only moderation or entertainment services, so we have created our bot which includes a range of features.

F.8 Discord bot illustration

1.3 Objective of the Major Project

Discord bot is the most salient feature of discord where you can create and develop your own bot according to your need to perform various functions.

This is also another unique feature that outclasses Discord with the other chatting apps out there. Bots are useful for representing certain data in a Discord server, delivering the notifications, scheduling an event, playing music for users on a server, etc.

The important thing is that when any server is holding a community of more than 100's members then the discord structure and bots make it very easy for admins to manage the users in their server in a very structured manner and make it very easy for them to perform moderation tasks.

1.4 Motivation of the Major Project

We are users of discord for more than 5 years and we have experienced the needs and uses of discord bots firsthand. Discord bots offer services that make handling large-scale as well as small discord servers quite easy.

Discord is not a small app anymore and many companies are investing in it or connecting their services to Discord. e.g. Patreon has already waded its way into it. Discord bots are also sold at discord bot shops to the users to help them maintain or improve their server.

These days, every content creator , from a youtube streamer to a company like Gameloop has their personal discord servers to interact with their friends, customers, and users.

F.9 Discord news screenshot

1.5 Organization:

Our discord bot is an interconnected python script that can be run on Visual studio code, python idle, atom, Jupiter notebook, and all the other tools used for running a python program.

As soon as the script runs on the shell, the bot comes online in the Discord server wherever it is added as a bot along with all its functionalities. The bot can then be used within the server for all the functions provided in it.

For organizing those functionalities we have used certain modern world pythons grouping methods such as the usage of cogs and events:-

1. **Cogs:**
   Cogs are used to organize a set of commands, listeners, and states in one class.

```
Discord Bot Project , Under Construction

Emoji:
  cry
  heart
  hello
  hi
  joy
  namaste
  plead
  rofl
Utils:
  userinfo
images:
  image     searches for a image on google
music:
  leave     Leaves the Voice Channel
  play      Plays a selected song from youtube
  queue     Displays the current songs in queue
  skip      Skips the current song being played
```

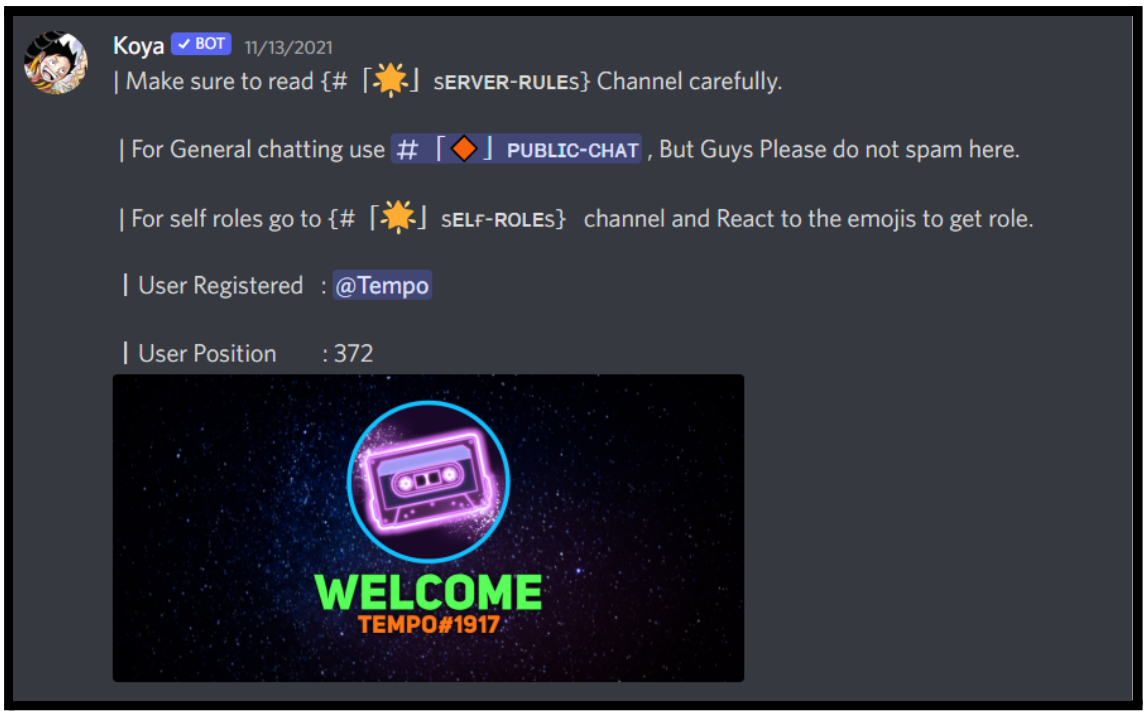
F.10 Cogs in our project

2. **Events:**

Events are like actions happening in the server that takes place, for instance when someone joins or leaves the server.

An event is something exact that happens (in our most memorable model code, the event is "a message is made").

At the point when an event/occasion is radiated, assuming there is a functional command connected to that occasion, this service will be executed.



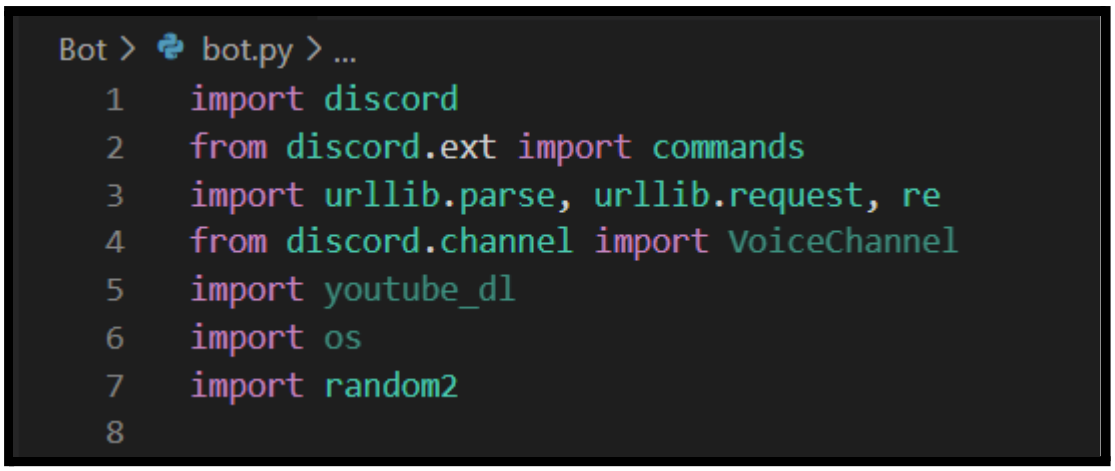
F.11 Discord Welcome channel screenshot

1.6 Language Used

The language used in this project is Python, Version 3.9.5. Python strives for readable codes and hence it is easier to read and write.

Libraries (python) used in this project are :

- discord
- urllib
- ffmpy
- youtube_dl
- os (generally pre-installed with Python)
- shutil
- random2
- FFmpeg
- vaderSentiment



F.12 Python libraries screenshot from our project

1.7 Technical and other Requirements

The main requirement for this project is **Bot Token**, which discord provides for submitting applications. Without a bot token, the bot cannot be run in discord.

- Windows (x86 or x64) pc or any other computer with Modern OS
- 4 GB RAM
- 1 GB free space
- Visual Studio Code (Version: 1.56.2 (user setup)) software or any other code editor
- Discord App for the live output.

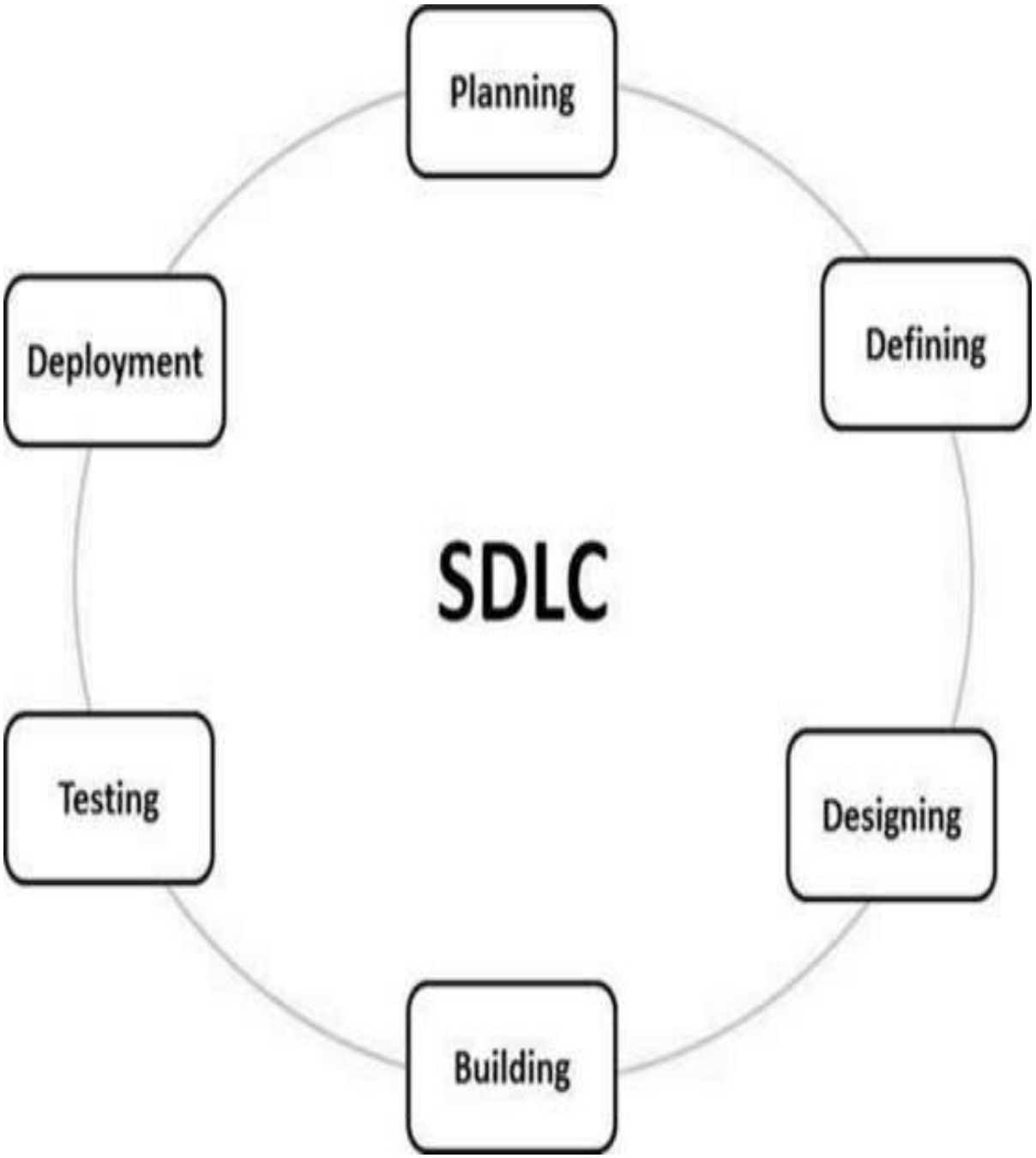
1.8 Deliverables of the Major Project

A discord bot is a bot that runs only on discord; it provides with you more functionality than a normal user has .

Our deliverables include :

- Audio Player (Connects to your voice channel and play music directly from youtube within discord)
- View information about any user in the discord server.
- Search for a video from youtube within discord .
- Automated greetings to the new user joining the server and farewell to the users leaving the server.
- Sentiment analysis on users' messages.
- Play fun games like tictactoe with other users .

CHAPTER 02: MAJOR PROJECT SDLC



F.13 Software Development Life Cycle (SDLC)

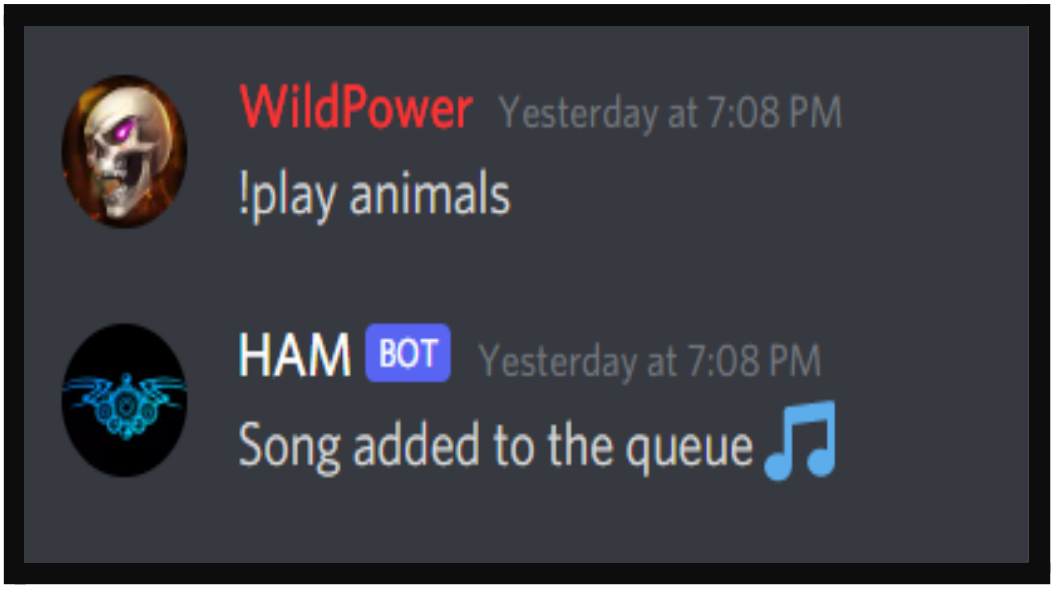
2.1 Feasibility Study on Major Project

Discord is a fast-growing app and discord bots are the backbone of discord servers. Discord bots offer many features to keep the server smooth and alive. So, having your own discord bot is very handy and safe.

The bots can help you detect spammers and fraud users. Out there, there are many discord bots for many purposes, but they are either paid or have subscription basis services . Every server must be maintained by discord bot to keep it clean and cozy . If the bot is good , it can gain enough charm and users will add the bot to mass servers .

The bot can also hold donation platforms or offer paid upgradeable services to farm revenue.Our discord bot is currently active in a few small servers and the user experience was quite seamless.

The bot can be easily created in a modern OS device in a code editor using various, programming languages and tools.

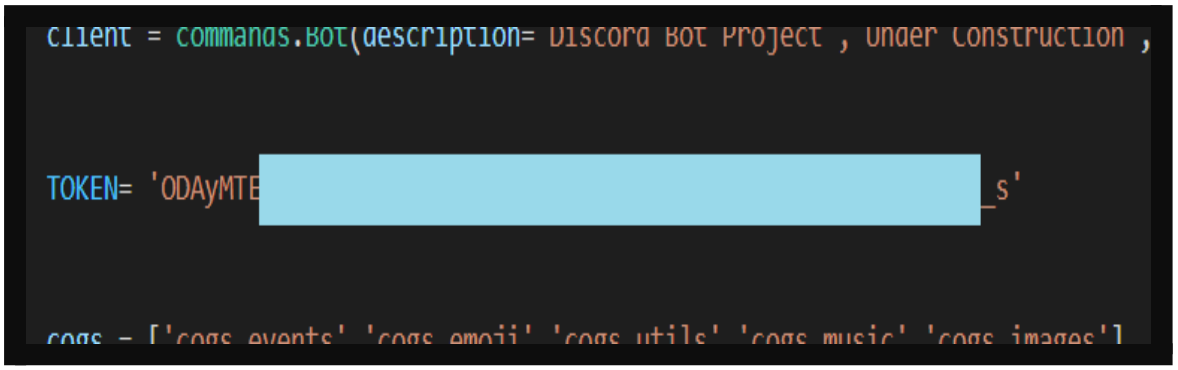


F.14 Screenshot of users using music commands

2.2 Requirements for Major Project

Discord's approval for the bot token on the submitted application is important for this project.

The bot token is like a password for the bot to use Discord API.



F.15 Bot Token used in the project, granted by discord

After the token is granted, we need our bot's code.

Our bot is written in Visual Studio Code software on a windows (x64) pc. The output was live observed on the discord app installed on the same pc. Apart from libraries (mentioned later), we used the basic structure of cog around various discord events to create various commands (bot services).

Cog is used to organize various functions to improve efficiency and readability.

```
cogs = ['cogs.events','cogs.emoji','cogs.utils','cogs.music','cogs.images']

for cog in cogs:
    try:
      client.load_extension(cog)
    except Exception as e:
      print(f'Could not load cog {cog}:|{str(e)}')
```

F.16 References to various cogs in the bot directory

2.2.1 Functional Requirements

- The system automatically detects new messages and distinguishes them as messages or commands.
-
- The system should be able to send messages in a discord channel.
-
- The system should be able to distinguish between other bots and human users.
-
- The system should have access to discord voice channels.
-
- The system should have permission to post links, files, and images.

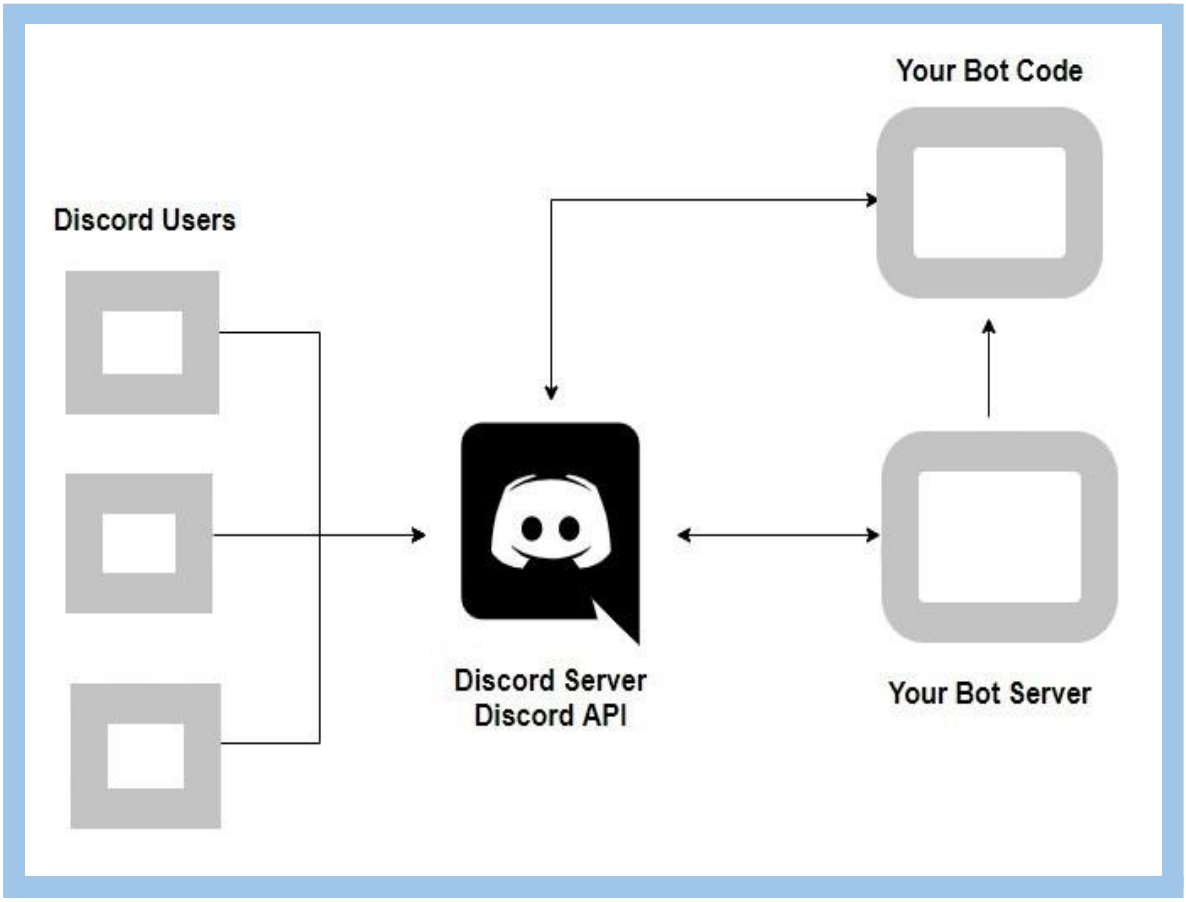
2.2.2 Non-Functional Requirements

- Users should not be allowed to use the bot in multiple voice channels on the same

server

- Every unsuccessful command by a user to access a service shall be recorded on the audit log.
- The system should be capable of holding simultaneous 100+ requests affecting its performance.
-
- The system should be portable so it can be deployed from one system or the other.
-
- The system shall not store private information about any user .

2.3 Use Case Diagram of the Major Project



F.17 Use a case diagram for the project

In the diagram above,

On the left are discord users which are connected to the server and are online to interact with other users and bots through the discord servers which can be via text channel or voice channels or DMs or groups ,etc..

In the middle is the official Discord server and API, which is maintained and run by discord itself.

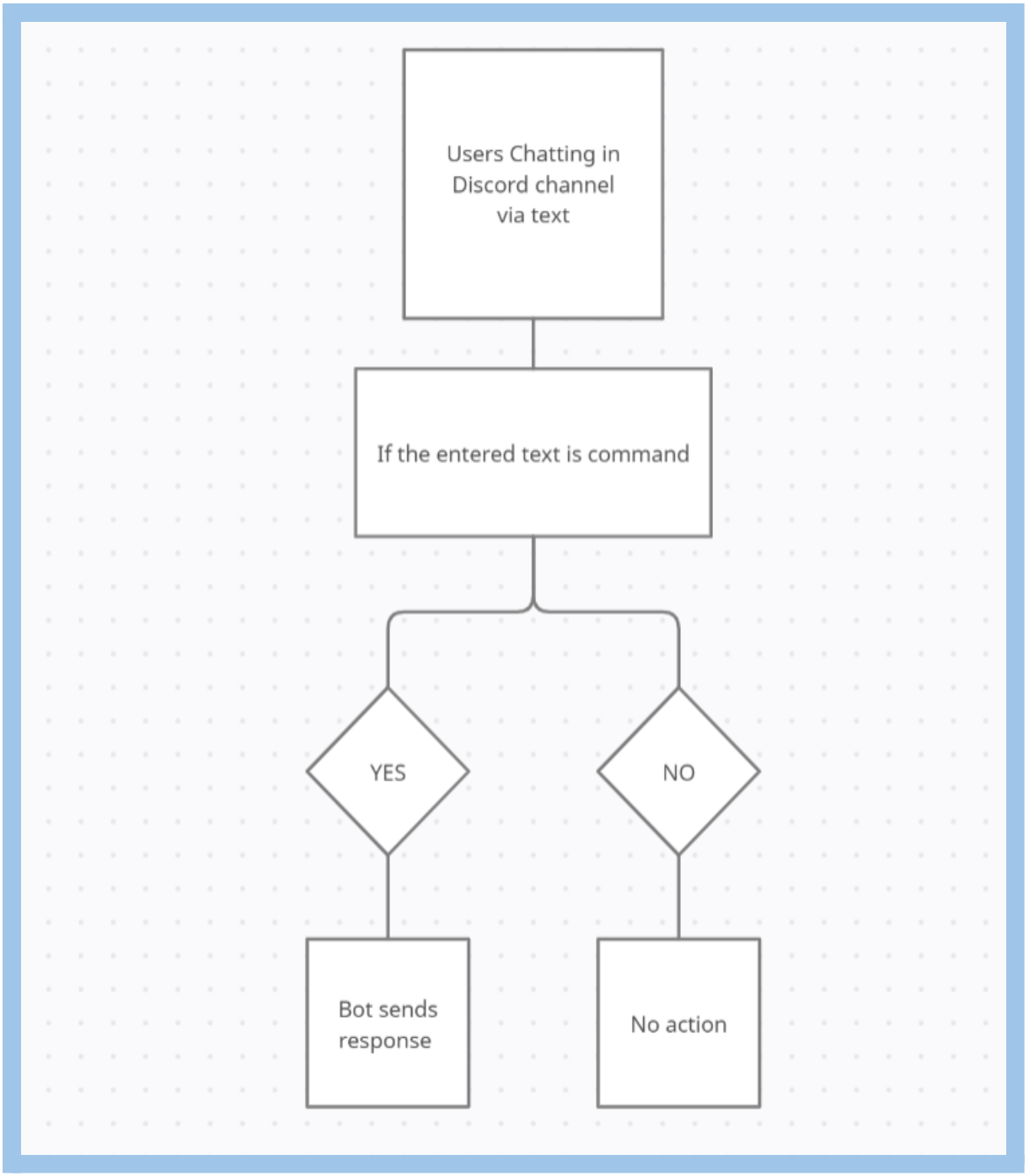
On the right top is our discord code which is connected to the discord API .

Discord bot code detects the commands ,events and responds with the function commands created in the code in our discord server.

Under our discord code is our server where all users and bots interact with each other.

Discord server is customizable by admin users according to their wishes and the server permissions can be set by the admin as per their choices.

2.4 DFD Diagram of the Major Project

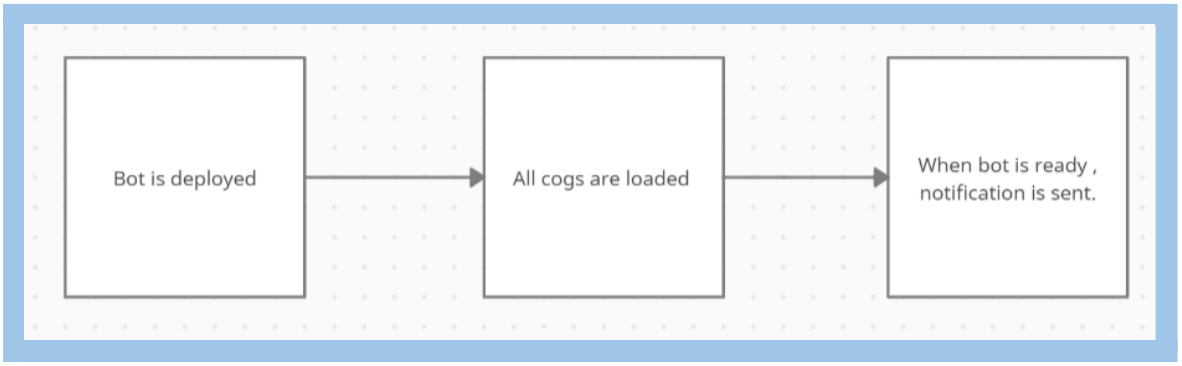


F.18 DFD diagram for the project

Our discord bot is a pre programmed bot . It detects the command and takes certain actions for an entered command.

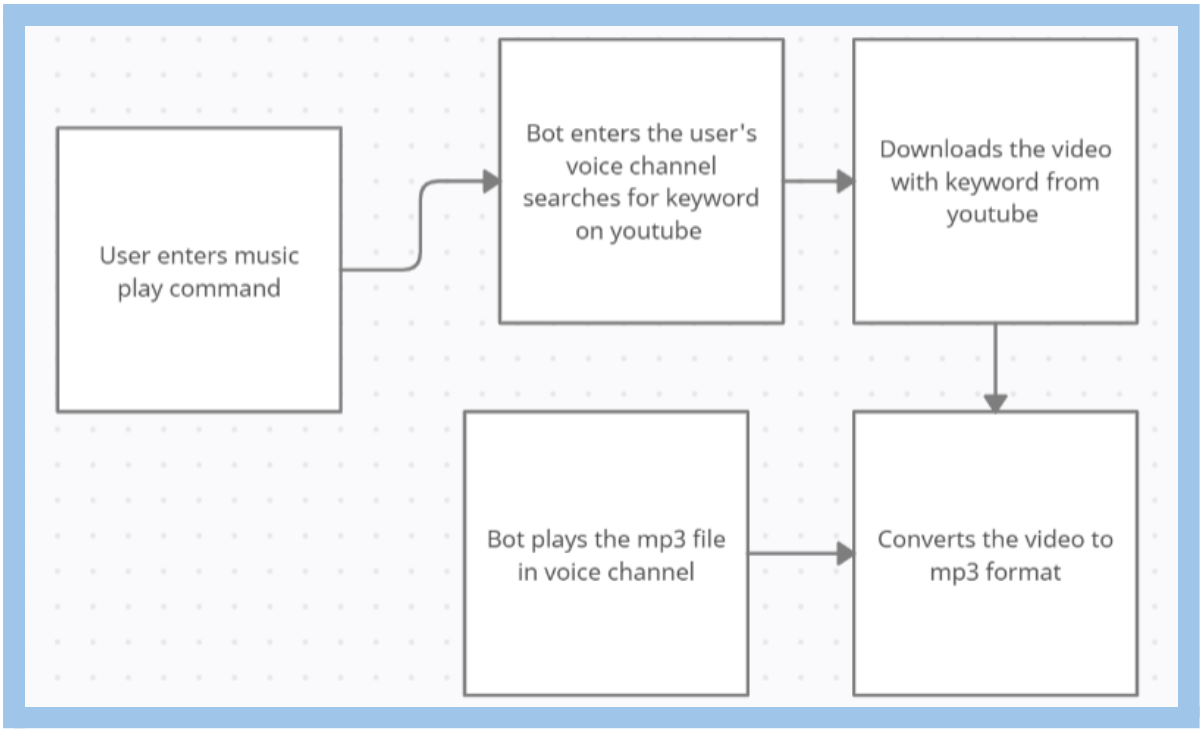
2.5 State Transition Diagram of the Major Project

- On running the bot



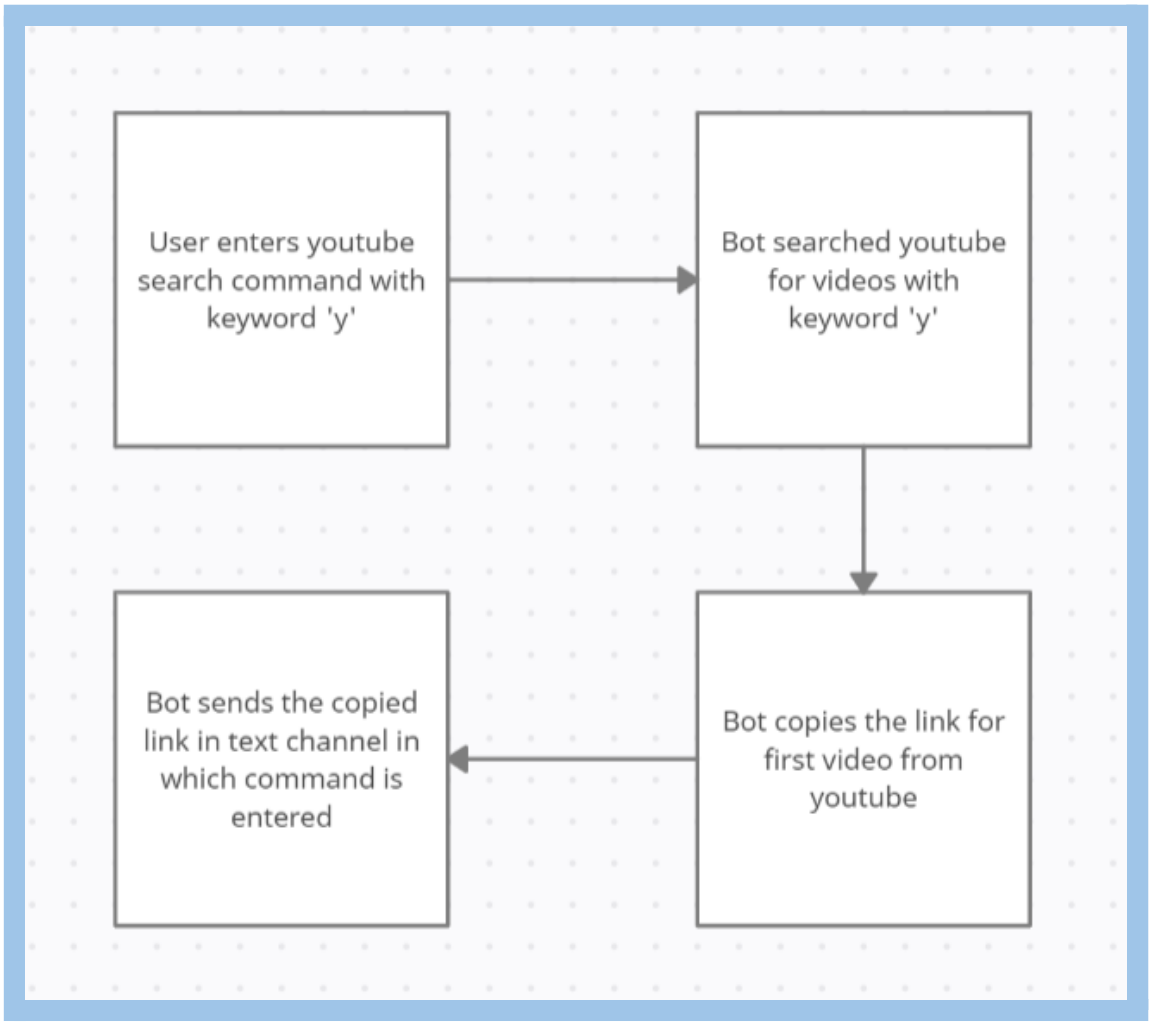
F.19 Bot running State Transition Diagram

- Music Command



F.20 Music command State Transition Diagram

- Youtube search command



F.21 Youtube search State Transition Diagram

- Tictactoe game command

F.10 Tictactoe discord illustration

Our bot can start a game of tictactoe in between two users . Although the game has to be commanded by any user , however the arbiter of the game is the discord bot which oversees for any fouls or misturn during the game and declares the winner after the game ends.

CHAPTER 03: IMPLEMENTATION OF THE MAJOR PROJECT

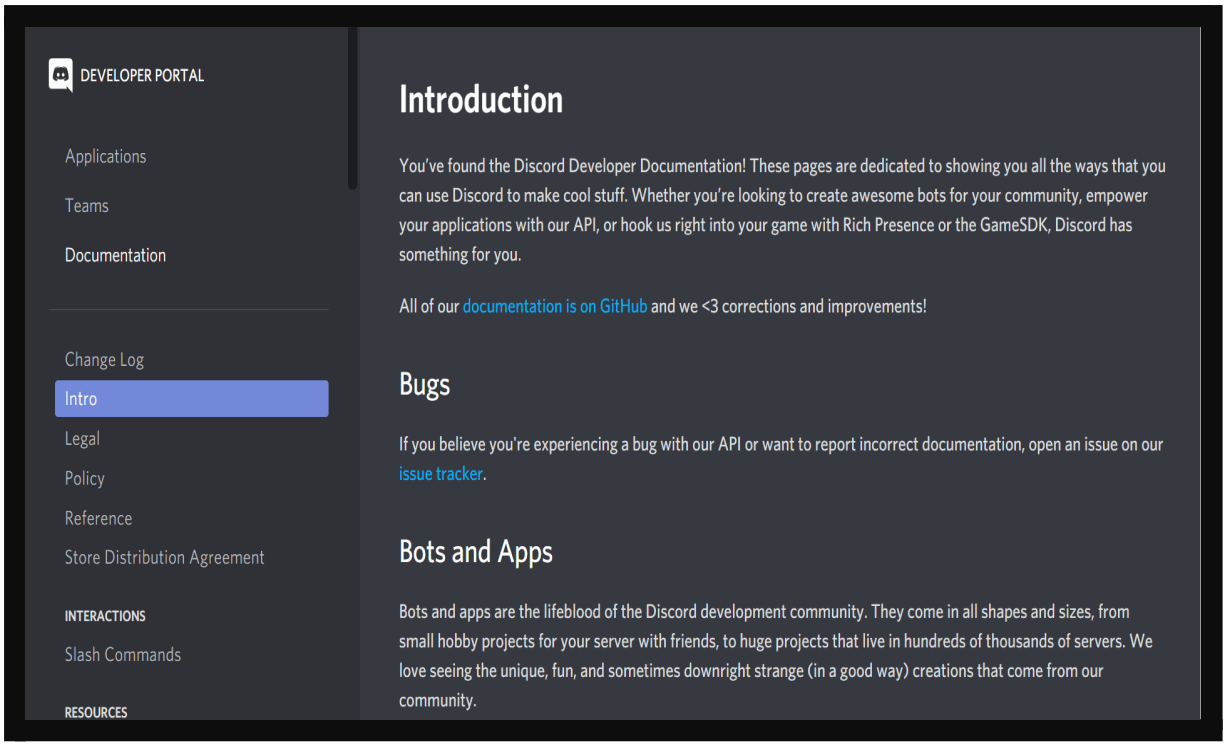
3.1 Discord Bot Account

In order to use the python library and discord API, we created a Discord bot account. First, log in to the discord developer portal web page ( ) and navigate to the application page.

There, create a “New Application”. Fill in the required details and submit the application. The discord dev will survey your application form and a token is granted on acceptance for the bot.

The Token is like a password for the bot. Once granted, the token can be regenerated if lost. The next step was to add a bot to your discord server.

We created a new discord server for this project and added the bot from the developer portal to our discord server.



F.23 Screenshot from the discord developer portal

3.2 Discord bot Code

We used the discord.py library to write the code for our bot.We also used other small py to create various services in discord for our server. Discord bot code mainly uses two components :

(a) Cogs

Cogs are used to organize a collection of bot commands into classes.

This helps in keeping one type of command together from the other. With commands, different cogs can be loaded and unloaded to maximize efficiency.

When cogs are created, they are needed to be registered with the main bot to be used. We have created separate cogs for entertainment commands, moderation, commands and chatbot commands.

```
4   class Utils(commands.Cog):
5
6       def __init__(self, client):
7
8           self.client = client
9
10      @commands.command()
11      async def userinfo(self,ctx,user:discord.User=None):
12          if user is None:
13              await ctx.send('Please mention a user to get info on ! ')
14              return
15          embed =discord.Embed(title='Userinfo', description=f'Here is
16          embed.add_field(name=user,value=f'- User\'s name: {user.name}
17          embed.set_thumbnail(url = user.avatar_url)
18
19          await ctx.send(':wave:', embed=embed)
20
```

F.24 Screenshot of cog “utils”

(b) Events

Events are like actions happening in the server which take place for instance when someone joins or leaves the server, some user sends a message or joins/leaves a voice channel.

These actions are called events. These events are used in discord bot code for the detection of commands to the bot and the action to be performed on giving a command.

3.3 Music Command Problem Statement

Creating a Discord music bot is not easy as discord itself does not offer music services .

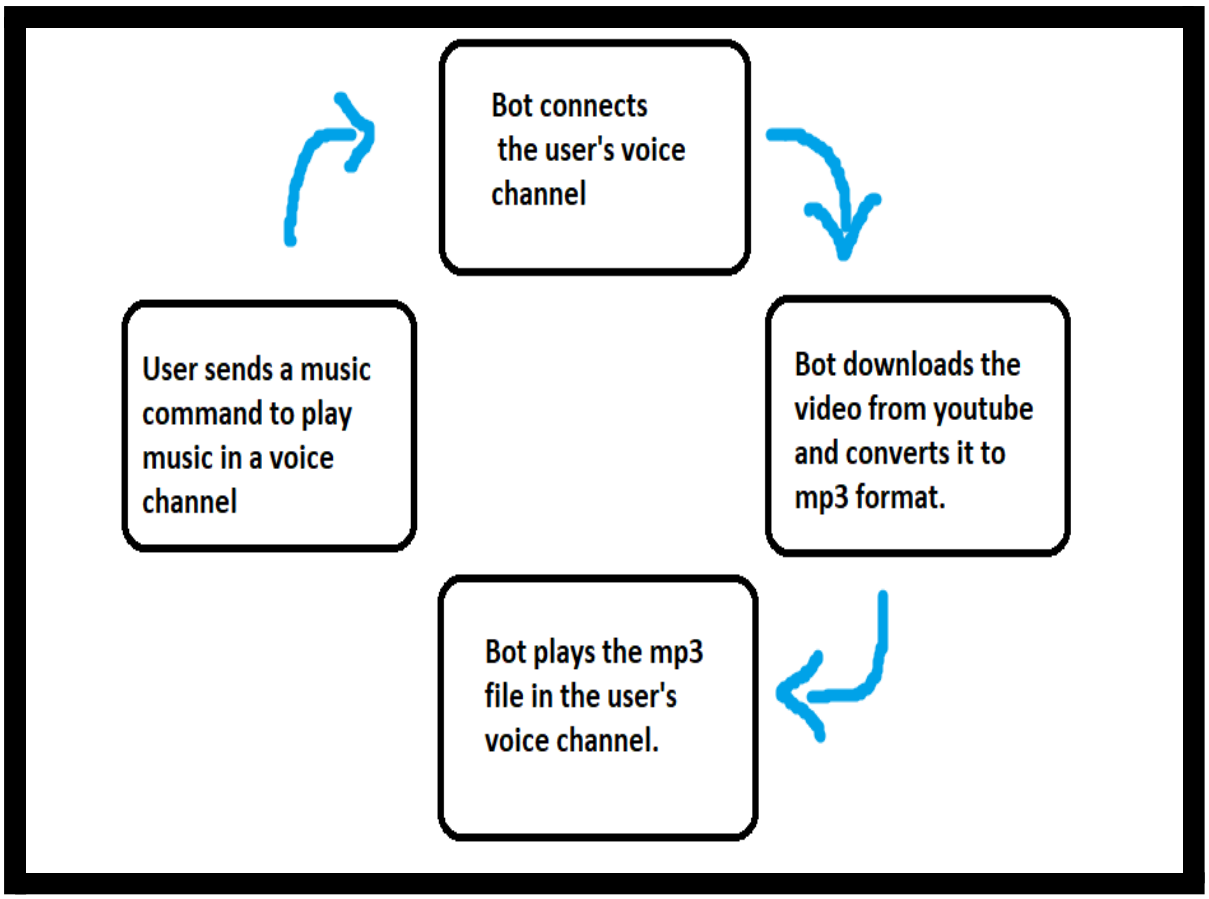
Therefore, we used youtube videos as a source for music in discord.

We used the youtube_dl python library to fetch videos from youtube within discord in our bot.

Then we converted the video into mp3 format using FFmpeg python library so the bot can play the mp3 file in a discord voice channel.

The bot first needs to connect to the same voice channel as the user and then download the video from youtube, convert it to mp3 format and play it on the voice channel.

If the user is not connected to any voice channel , the bot refuses to connect to a voice channel and play any music .

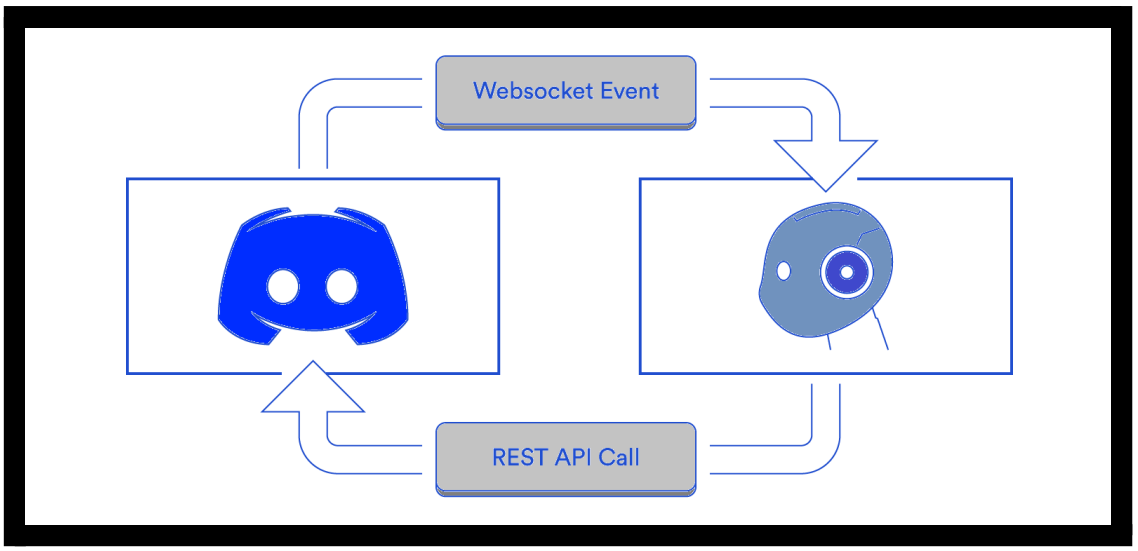


F.25 Music command problem statement diagram

3.4 Flow graph of the Major Project Problem

In the diagram below,

on the left is the discord server and on the right is the discord bot



F.26 Flow graph of the problem

Discord users interact with each other via sending messages in text channels or talking in voice channels .

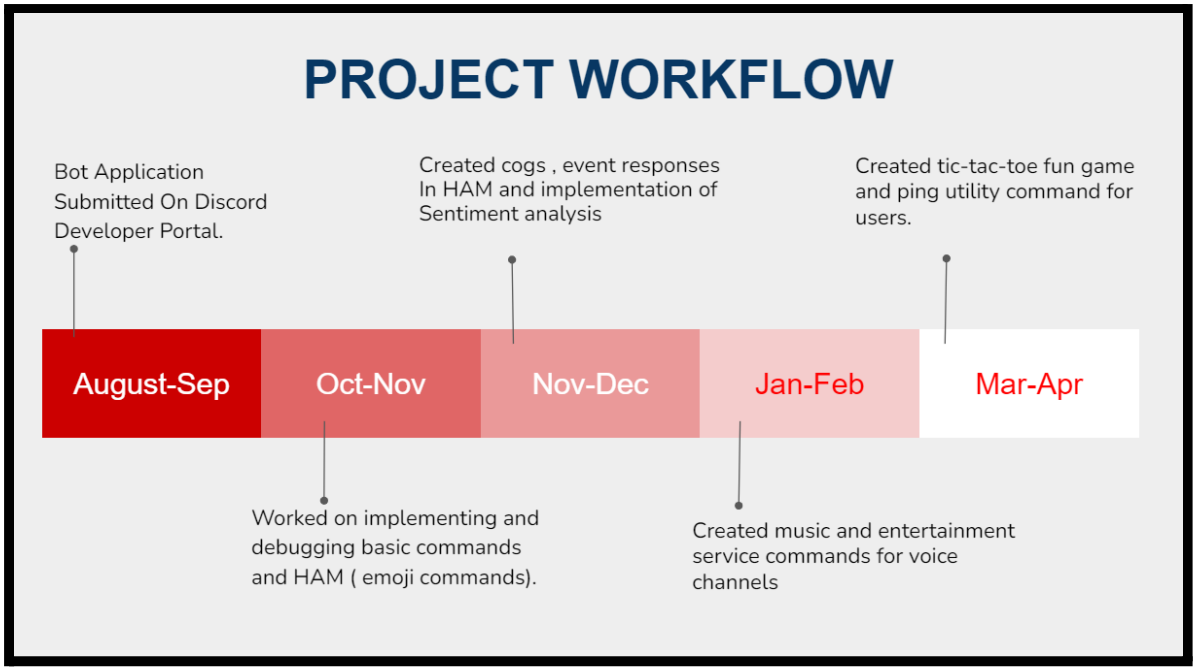
Commands to the bot can be sent via text in text channels on which the bot takes specified actions.

The bot interacts with Discord through discord API and when commands are triggered , the bot reacts in a pre-programmed way.

The Websocket API is used to receive events from discord,including message creation,deletion,user kick/ban events and permission updates etc.

The Discord REST API is used by bots to perform most action such as sending messages, updating user info etc.

3.5 Workflow of the Major Project Problem



F.27 Workflow diagram

Our recent project development includes the tictactoe game and shifting of our project from windows 10 to a windows 11 pc .

Shifting from one version of the operating system to another required updation of python libraries in our project which we performed in the last stages of our project.

3.6 Screenshots of the various stages of the Project

- The main body of code

  Below are the screenshots from our code of our project , which we coded in Microsoft visual studio code .

```python
import discord
from discord.ext import commands
import urllib.parse, urllib.request, re
from discord.channel import VoiceChannel
import youtube_dl
import os


intents = discord.Intents.default()
intents.members = True


client = commands.Bot(description="Discord Bot Project , Under Construction",command_prefix='!', intents=intents)




cogs = ['cogs.events','cogs.emoji','cogs.utils','cogs.music','cogs.images']

for cog in cogs:
    try:
      client.load_extension(cog)
    except Exception as e:
      print(f'Could not load cog {cog}:|{str(e)}')
```

F.28 Code Screenshot 1

```python
@client.event
async def on_ready():
    print('-----------------------------------------The bot is now ONLINE !--------------------------------')


@client.command()
async def loadcog(ctx, cogname = None):


    if cogname is None:
        return

    try:
        client.load_extension(cogname)
    except Exception as e:
        print(f'Could not load cog {cogname}: {str(e)}')
    else:
        print('Loaded Cog Successfully')

@client.command()
async def unloadcog(ctx, cogname = None):

    if cogname is None:
        return

    try:
        client.unload_extension(cogname)
    except Exception as e:
        print(f'Could not unload cog {cogname}: {str(e)}')
    else:
        print('Unloaded Cog Successfully')


```

F.29 Code Screenshot 2

```
190
191 def checkWinner(winningConditions, mark):
192     global gameOver
193     for condition in winningConditions:
194         if board[condition[0]] == mark and board[condition[1]] == mark and board[condition[2]] == mark:
195             gameOver = True
196
197 @tictactoe.error
198 async def tictactoe_error(ctx, error):
199     print(error)
200     if isinstance(error, commands.MissingRequiredArgument):
201         await ctx.send("Please mention 2 players for this command.")
202     elif isinstance(error, commands.BadArgument):
203         await ctx.send("Please make sure to mention/ping players (ie. <@688534433879556134>).")
204
205 @place.error
206 async def place_error(ctx, error):
207     if isinstance(error, commands.MissingRequiredArgument):
208         await ctx.send("Please enter a position you would like to mark.")
209     elif isinstance(error, commands.BadArgument):
210         await ctx.send("Please make sure to enter an integer.")
211
212 client.run(TOKEN)
```

F.30 Code Screenshot 3

- Cogs

```
1  import discord
2  from discord.ext import commands
3
4  class Utils(commands.Cog):
5
6      def __init__(self, client):
7
8          self.client = client
9
10     @commands.command()
11     async def userinfo(self,ctx,user:discord.User=None):
12         if user is None:
13             await ctx.send('Please mention a user to get info on ! ')
14             return
15         embed =discord.Embed(title='Userinfo', description=f'Here is some infor
16         embed.add_field(name=user,value=f'- User\'s name: {user.name}\n- User\'
17         embed.set_thumbnail(url = user.avatar_url)
18
19         await ctx.send(':wave:', embed=embed)
20
```

F.31 Code Screenshot 4

This is a sample screenshot from a cog “ Utils” which has some utility commands in it.

- Music Commands

```python
import discord
from discord.channel import VoiceChannel
from discord.ext import commands
from discord.voice_client import VoiceClient

from youtube_dl import YoutubeDL

class music(commands.Cog):
    def __init__(self, client):
        self.client = client
        self.is_playing = False

        self.music_queue = []
        self.YDL_OPTIONS = {'format': 'bestaudio', 'noplaylist':'True'}
        self.FFMPEG_OPTIONS = {'before_options': '-reconnect 1 -reconnect_streamed 1 -reconnect_delay_max 5', 'options': '-vn'}

        self.vc = ""

    def search_yt(self, item):
        with YoutubeDL(self.YDL_OPTIONS) as ydl:
            try:
                info = ydl.extract_info("ytsearch:%s" % item, download=False)['entries'][0]
            except Exception:
                return False

        return {'source': info['formats'][0]['url'], 'title': info['title']}

    def play_next(self):
        if len(self.music_queue) > 0:
            self.is_playing = True
```

F.32 Code Screenshot 5

This is a screenshot from the music cog, showing the play function.

Music cog has variety of codes ranging from vode for searching music keyword to connecting to a voice channel. Above figure is just a peek at the huge code of music cog.

- Sentiment analysis function command

```
73  def sentiment_analyzer_scores(text):
74
75      score = analyzer.polarity_scores(text)
76      lb = score['compound']
77      if lb >= 0.05:
78          return 'Good'
79      elif (lb > -0.05) and (lb < 0.05):
80          return 'neutral'
81      else:
82          return 'Bad'
83
84  @client.event
85  async def on_ready():
86      print('Logged in as {0.user}'.format(client))
87
88  @client.event
89  async def on_message(message):
90      if message.author == client.user:
91          return
92      sentiment = sentiment_analyzer_scores(message.content)
93      print('sentiment:' + str(sentiment))
94      await message.channel.send('The sentiment of your message is ' + str(sentiment))
95
```

F.33 Code Screenshot 6

Here, you can see the function to analyze the message entered by the user.

- TicTacToe game function code

  Our bot uses emoticons to create the board panel for playing tictactoe game and lets two users to enjoy the game .

```
player1 = ""
player2 = ""
turn = ""
gameOver = True

board = []

winningConditions = [
    [0, 1, 2],
    [3, 4, 5],
    [6, 7, 8],
    [0, 3, 6],
    [1, 4, 7],
    [2, 5, 8],
    [0, 4, 8],
    [2, 4, 6]
]

@client.command()
async def tictactoe(ctx, p1: discord.Member, p2: discord.Member):
    global count
    global player1
    global player2
    global turn
    global gameOver

    if gameOver:
        global board
        board = [":white_large_square:", ":white_large_square:", ":white_large_square:",
                 ":white_large_square:", ":white_large_square:", ":white_large_square:",
                 ":white_large_square:", ":white_large_square:", ":white_large_square:"]
        turn = ""
        gameOver = False
        count = 0

        player1 = p1
```

F.34 Code Screenshot 7

In above figure , we have crated the game board with white large square emoticons for default state of game and then replac them with cross emoticons or circle emoticons as per the choice of player's move .

The bot sends the updates board in the chat after a user has played one turn .

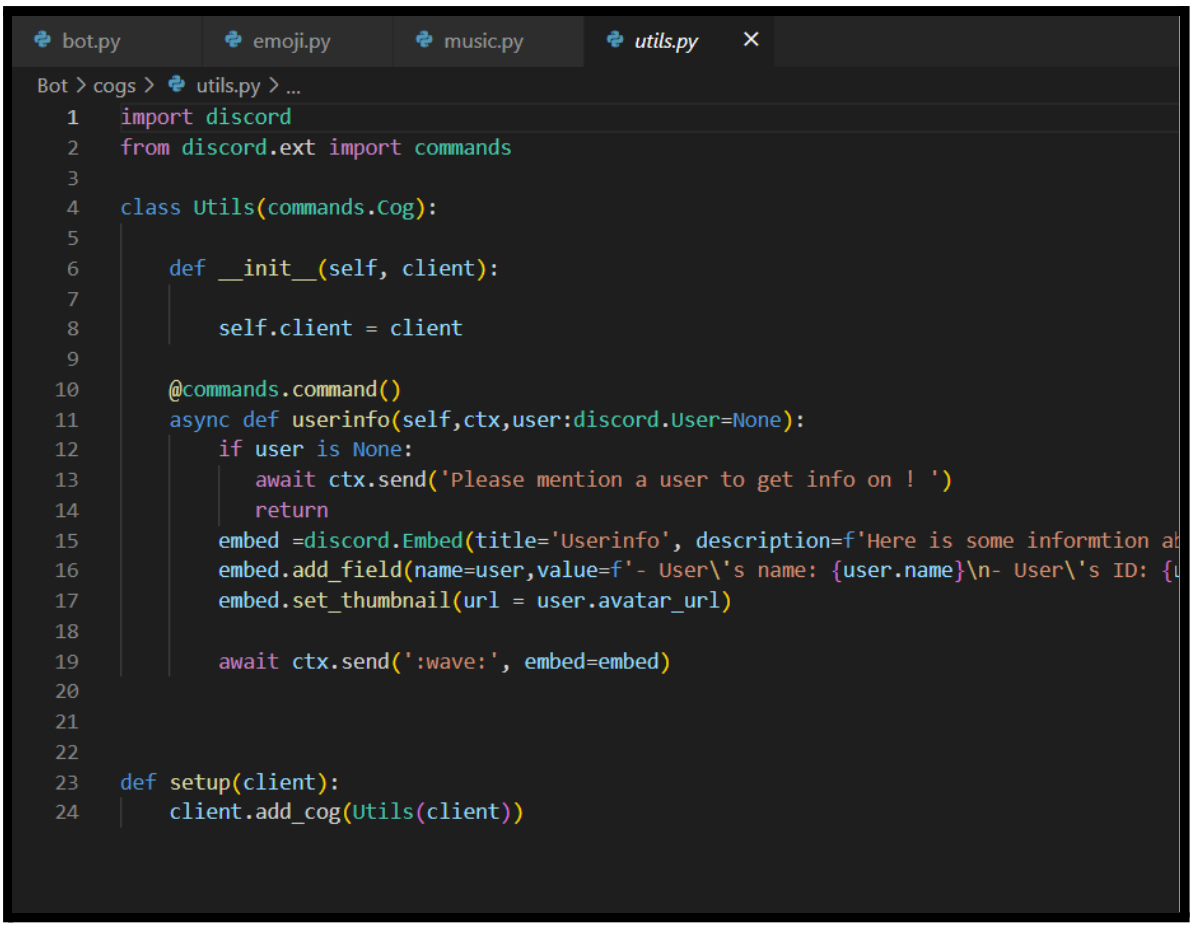
This continues until the game has reached deadlock state or a player has won.

```
@client.command()
async def place(ctx, pos: int):
    global turn
    global player1
    global player2
    global board
    global count
    global gameOver

    if not gameOver:
        mark = ""
        if turn == ctx.author:
            if turn == player1:
                mark = ":regional_indicator_x:"
            elif turn == player2:
                mark = ":o2:"
            if 0 < pos < 10 and board[pos - 1] == ":white_large_square:" :
                board[pos - 1] = mark
                count += 1

                # print the board
                line = ""
                for x in range(len(board)):
                    if x == 2 or x == 5 or x == 8:
                        line += " " + board[x]
                        await ctx.send(line)
                        line = ""
                    else:
                        line += " " + board[x]

                checkWinner(winningConditions, mark)
                print(count)
                if gameOver == True:
                    await ctx.send(mark + " wins!")
                elif count >= 9:
```

F.35 Code Screenshot 8

- Utility command code

Utility commands provide help to the admin to access the information regarding the discord users, servers, etc.



F.36 Code Screenshot 9

In the above screenshot , we have created utility based command function userinfo for accessing public information of any discord account added in the bot's server .

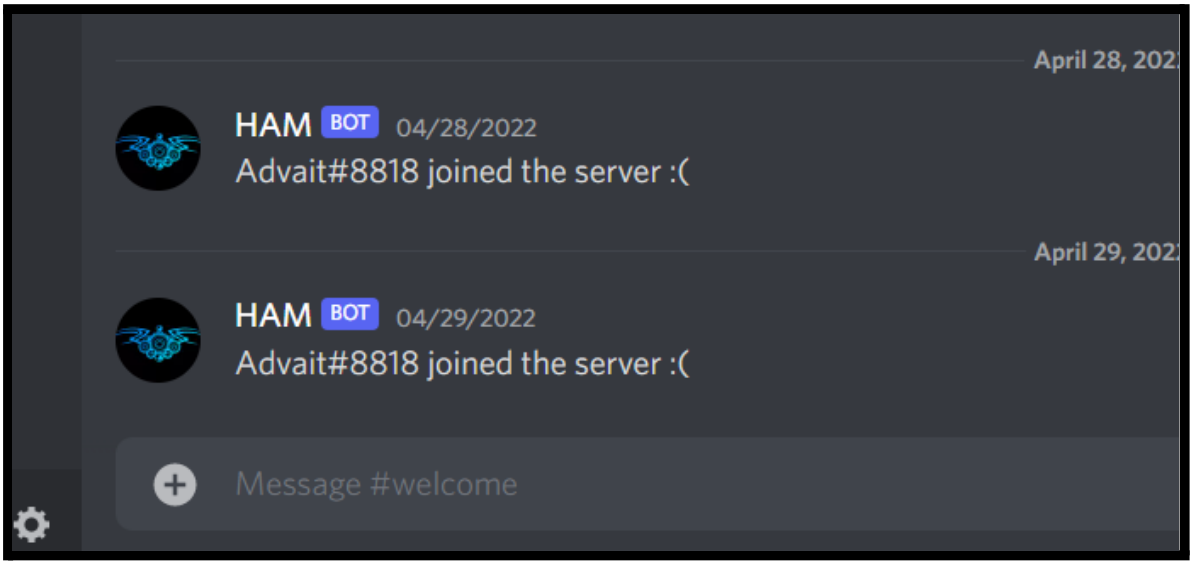
This helps the admins to detect fraud accounts and prevent spam in the server.

CHAPTER 04: RESULTS

4.1 Discussion on the Results Achieved

We created all the necessary commands and were able to achieve the following results:

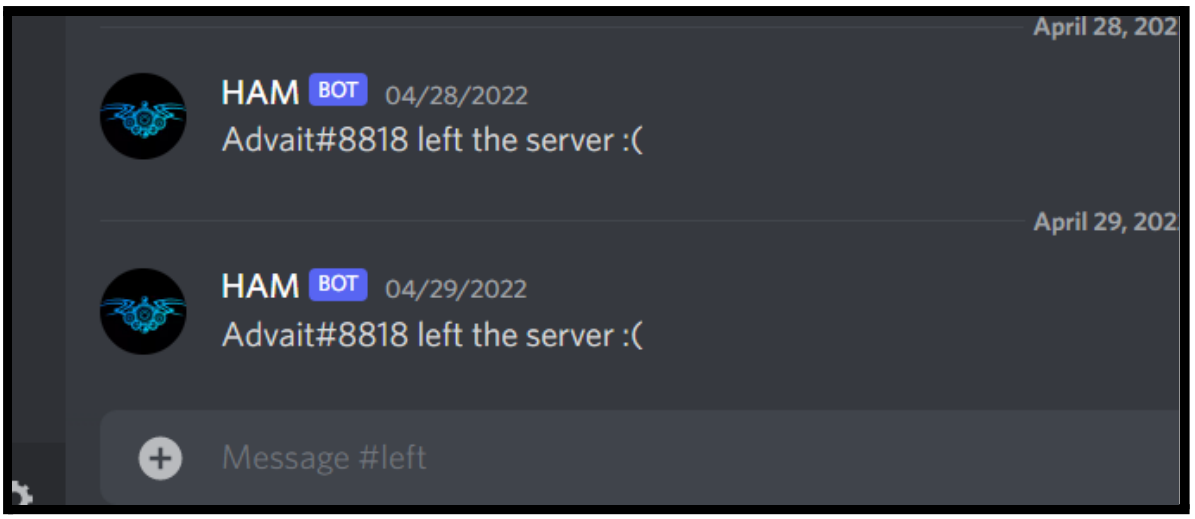
1. HAM is able to acknowledge and notify if a user enters the server.



F.37 Output Screenshot 1

When a user enters the server , the event of server entry is created in the discord API , and the bot is pre-programmed to push notification in a given channel on detecting such an event .

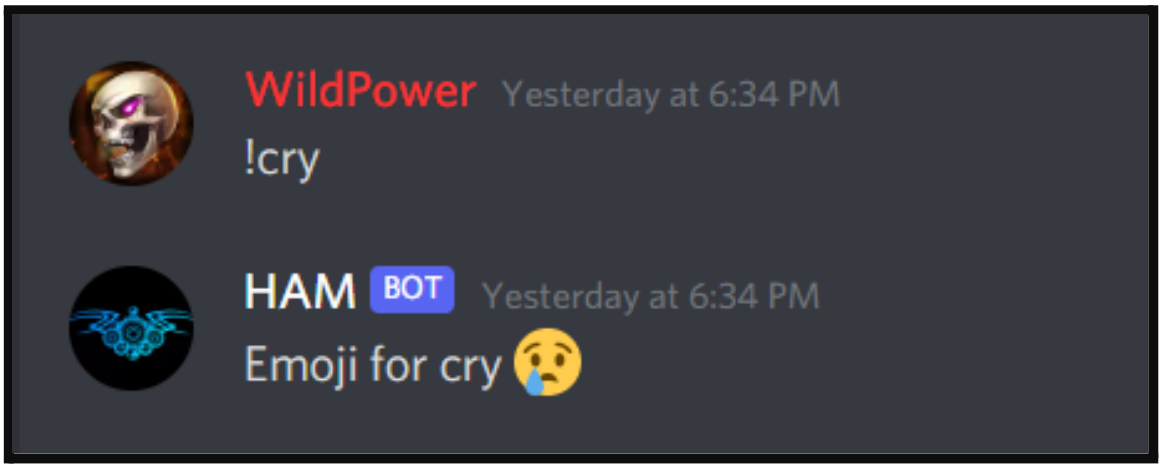
2. HAM is able to acknowledge and notify if a user exits the server.



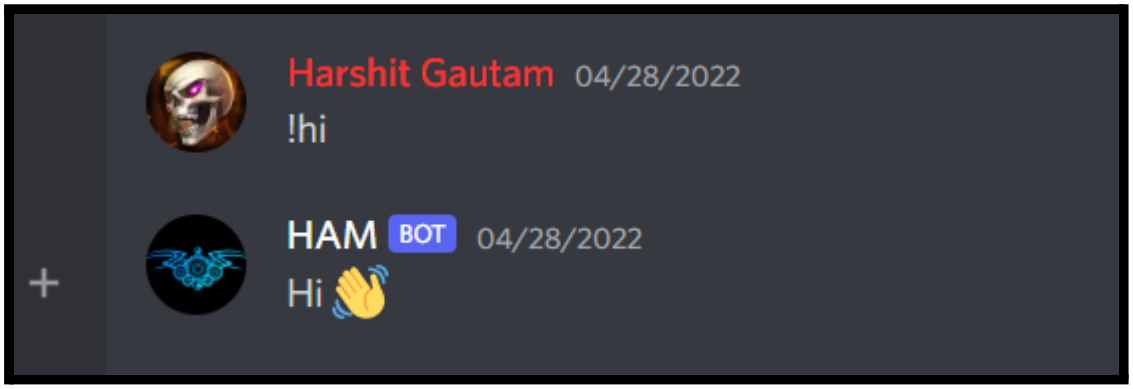
F.38 Output Screenshot 2

When a user leaves the server, the event of server leave is created in the discord API, and the bot is pre-programmed to push notifications in a given channel on detecting such an event.

3. HAM is able to understand emotions and is able to recognize that as created in its file.

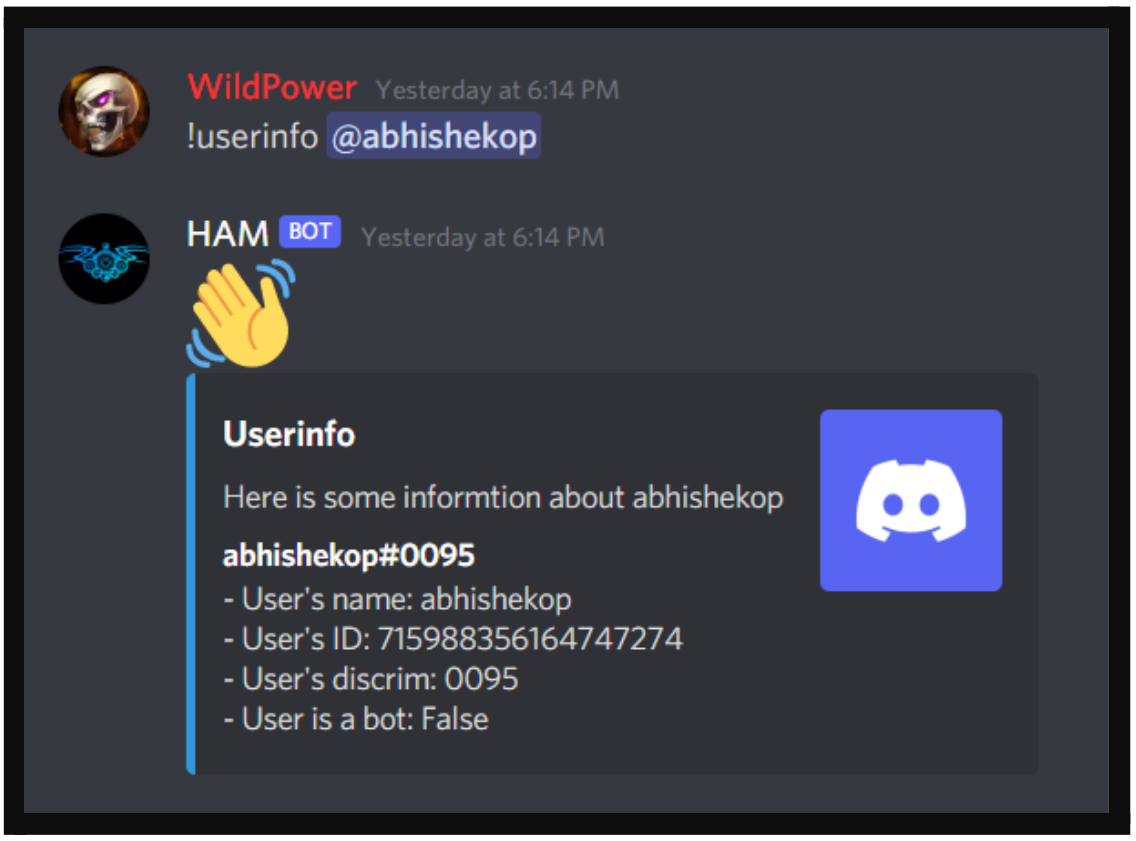


F.39 Output Screenshot 3



F.40 Output Screenshot 4

4 . HAM is able to identify the user and present necessary details about it in the form of embeds .

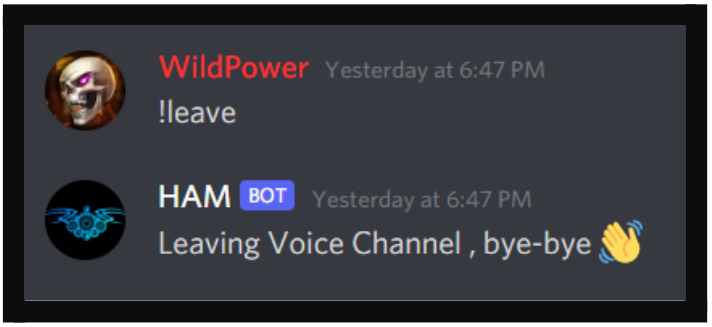


F.41 Output Screenshot 5

4. HAM is able to play music on the server.

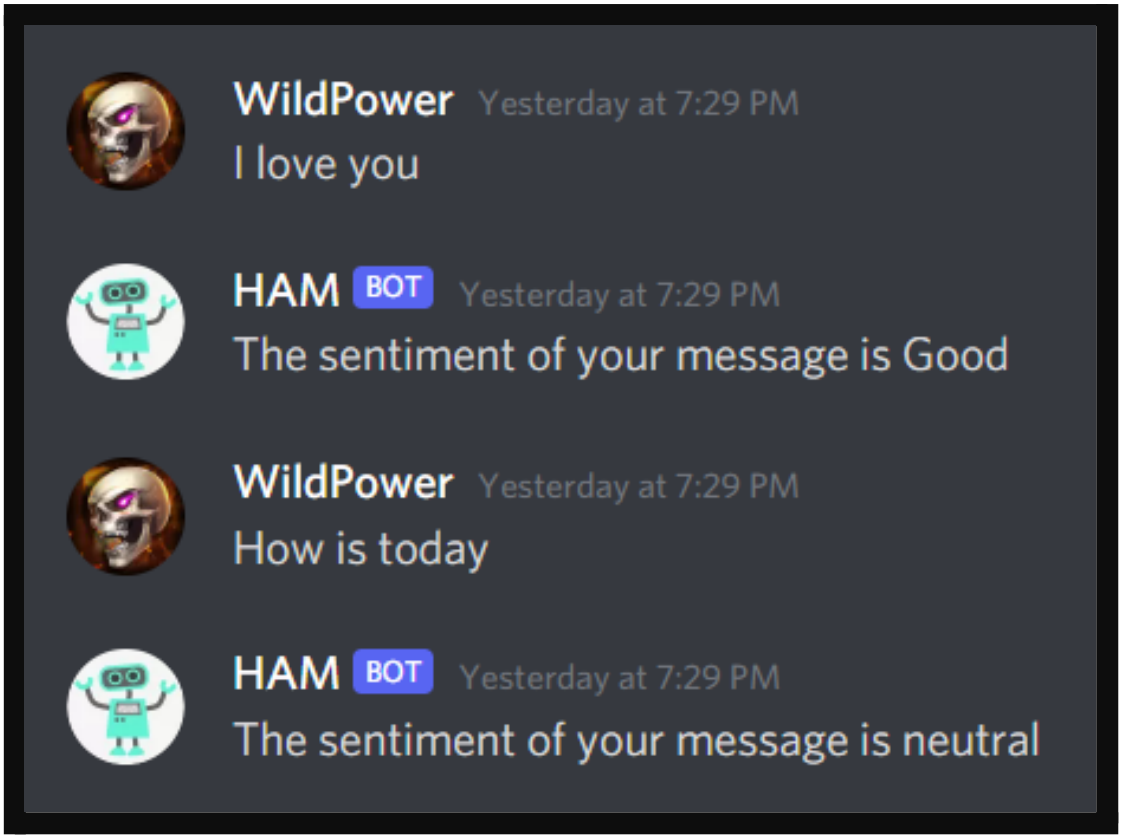
The bot plays music using the play command from youtube and also has commands like a queue, and stop for further control.

F.42 Output Screenshot 6



F.43 Output Screenshot 7

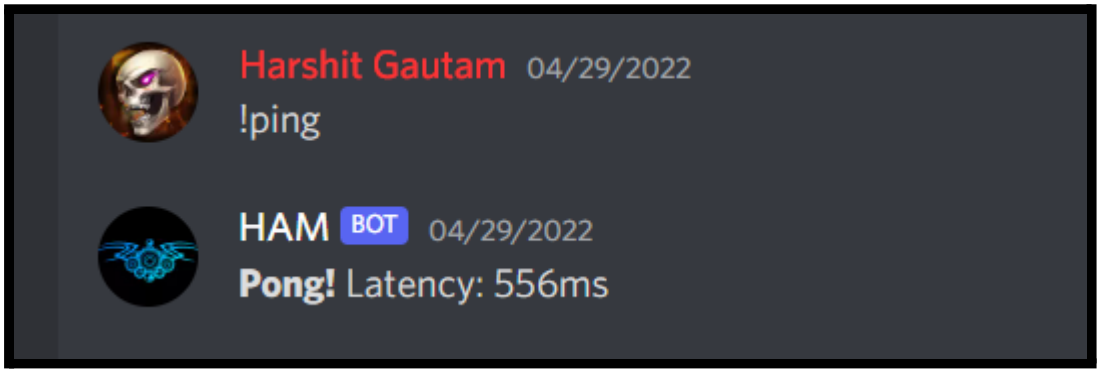
5. HAM is able to provide sentiment analysis on users' messages.



F.44 Output Screenshot 8

You just send a message and the bot will provide sentiment analysis on the message.

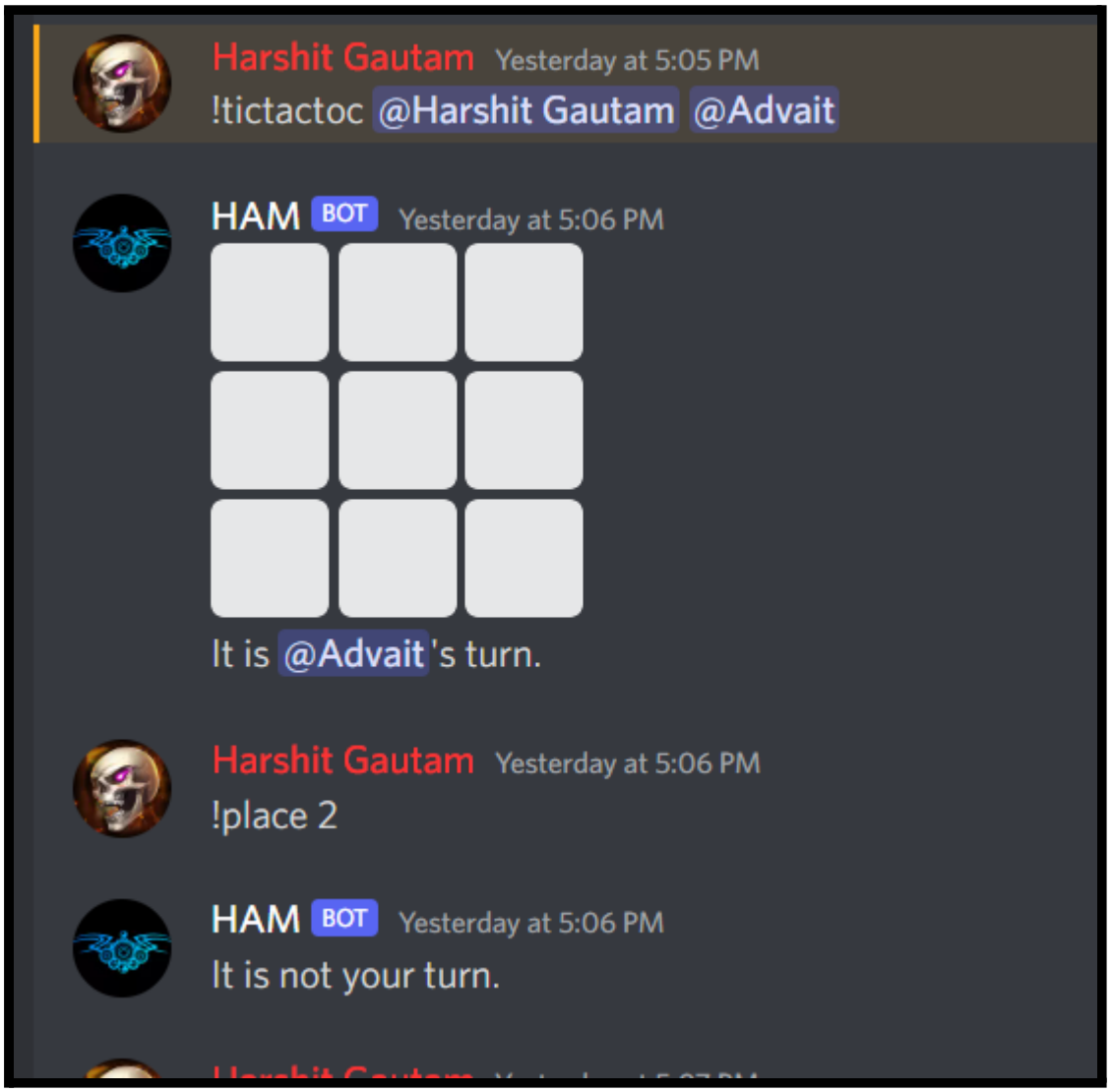
6. HAM is able to measure the ping of any user .



F.45 Output Screenshot 9

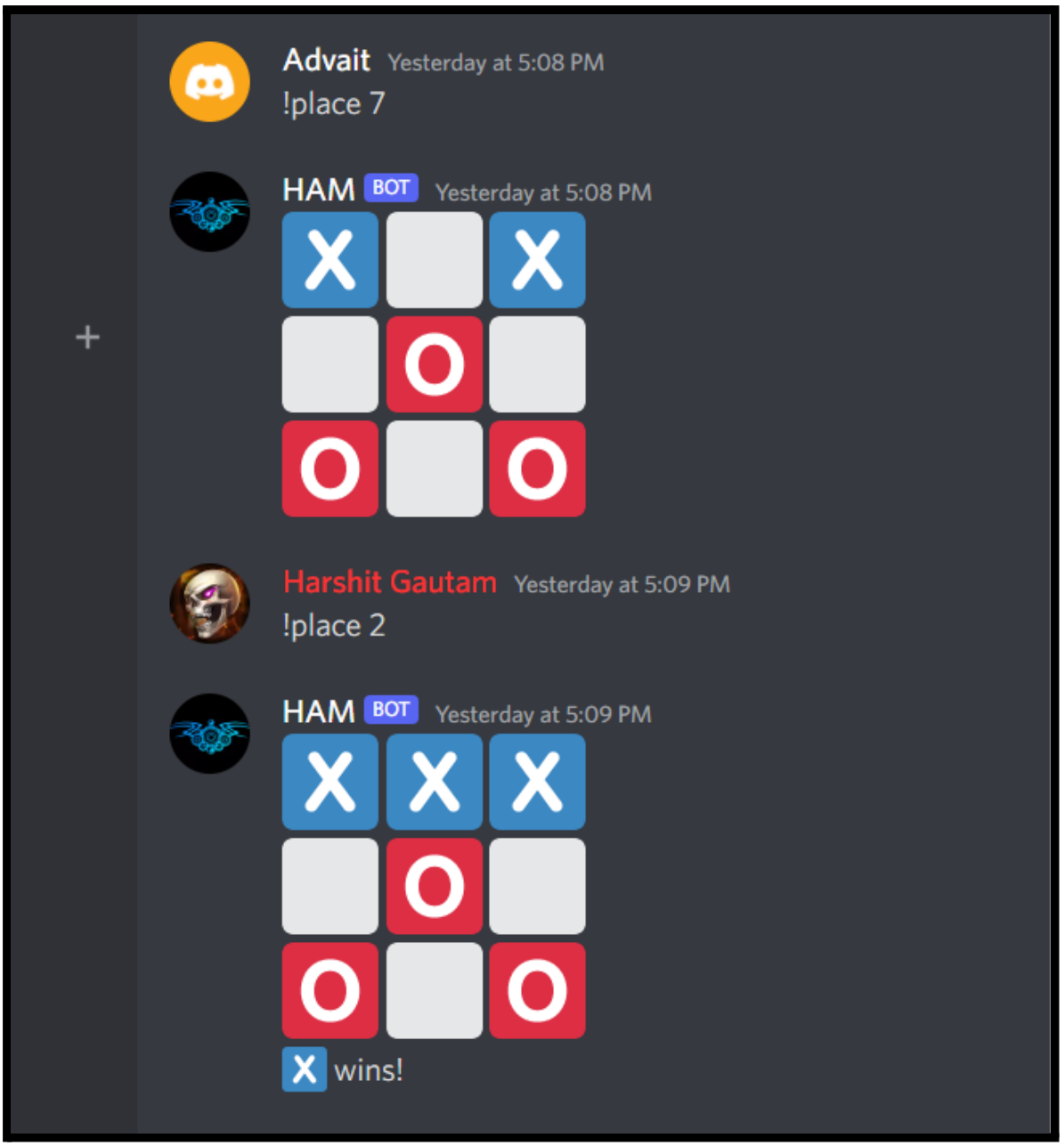
This helps the user to check his internet strength and speed.

7. HAM is able to initiate TicTacToe, a fun game between two users.



F.46 Output Screenshot 10

When the game ends, the bot announces the winner of the game and ends the game session.



F.47 Output Screenshot 11

In the above figure , the bot can be seen to end the game and declare the player winner with the ‘X” emoticon mark.

4.2 Application of the Major Project

The application of the project is to perform all the necessary functions on the discord server and perform specific tasks assigned to it. Provide help to admin as well as users in the server for their work and make their work more hassle-free.

The aim of a discord bot is to perform repetitive tasks that can be tedious for the admin as well as the user and having a bot on the server makes it very easy.

4.3 Limitation of the Major Project

- HAM is currently available on small servers where participants are less than 100.
- HAM's features are limited right now and don't have voice commands.
- HAM isn't available 24X7 as it is hosted on our pc, not on a cloud server.
- HAM efficiency is tested on small servers, not on large servers.

4.4 Future Work

- Make HAM available 24X7 by hosting it on cloud servers
- Provide HAM with more moderation commands for better user experience
- Make HAM equipped with voice commands for performing task
- Develop HAM to create its own currency for trading purpose

REFERENCES

- Anirudh Verma, S. T. G. M. (2021). A Comprehensive Review on Bot - Discord Bot. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *7*(2). https://doi.org/https://doi.org/10.32628/CSEIT2172100
- Rapptz. (2015). *Discord.py API Reference. (2015)*. https://discordpy.readthedocs.io/en/latest/api.html
- Python community. (2003). *PyPI · The Python Package Index.* https://pypi.org
- Alex Ronquillo. (2019). *How to Make a Discord Bot in Python*. https://realpython.com/how-to-make-a-discord-bot-python/
- Randall Schmidt. (2019). *How to Make a Discord Bot: an Overview and Tutorial*. https://www.toptal.com/chatbot/how-to-make-a-discord-bot
- Beau. (2019). *freeCodeCamp.org*. https://www.youtube.com/watch?v=SPTfmiYiuok&ab_channel=freeCodeCamp.org

Software download References :

- Discord App.

  https://discord.com/download

- Visual Studio Code .

  https://code.visualstudio.com/download

- Python

  https://www.python.org/downloads/

- Libraries :

discord : https://pypi.org/project/discord.py/

urllib : https://pypi.org/project/urllib3/

re : https://pypi.org/project/regex/

youtube_dl : https://pypi.org/project/youtube_dl/

shutil : https://pypi.org/project/pytest-shutil/

random : https://pypi.org/project/random2/

ffmpeg : https://pypi.org/project/ffmpeg-python/

vaderSentiment : https://pypi.org/project/vaderSentiment/