



TRƯỜNG ĐẠI HỌC SÀI GÒN  
KHOA CÔNG NGHỆ THÔNG TIN



## PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

Phát triển ứng dụng phát nhạc bằng hình thức stream file sử dụng Socket Programming và thư viện Discord.py

GVHD: Từ Lăng Phiêu

SV: Triệu Phùng Tân Nguyên – 3118410291

TP. HỒ CHÍ MINH, THÁNG 5/2024

## Mục lục

1. Giới thiệu về dự án. ....	2
1.1. Giới thiệu: ....	2
1.2. Các tiêu chí cần đạt được. ....	3
2. Các thư viện cần cài đặt. ....	3
3. Cài đặt Server. ....	3
4. Cấu hình ứng dụng trên Discord Developer Portal. ....	4
5. Cài đặt Client.....	6

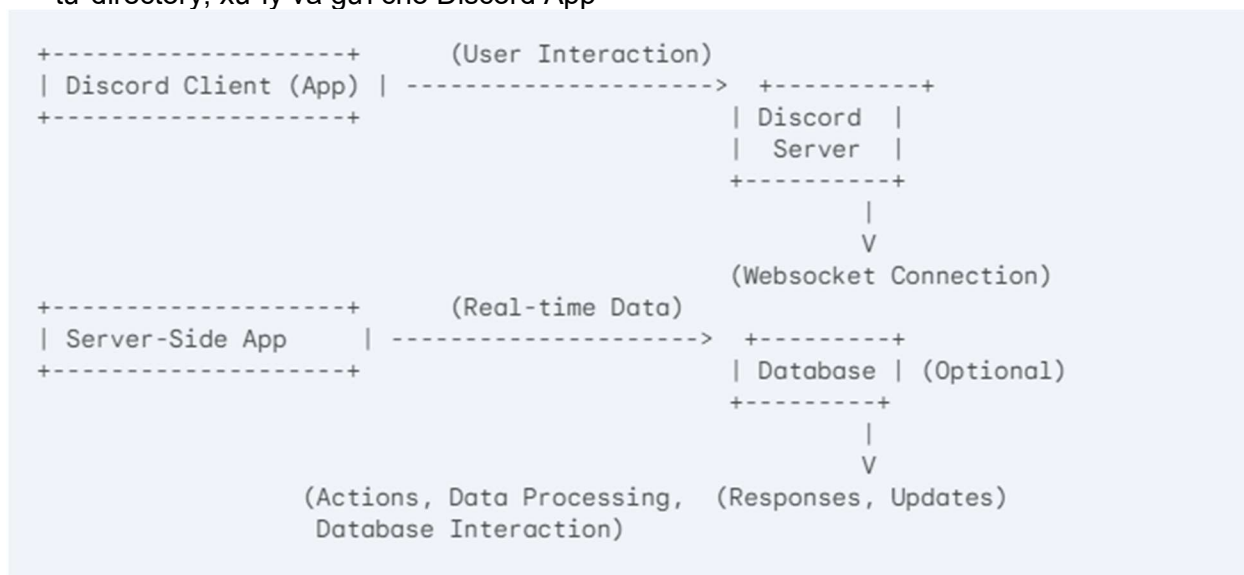
## 1. Giới thiệu về dự án.

### 1.1. Giới thiệu:

Hệ thống này được xây dựng theo kiến trúc Client-Server, dựa trên các thư mục và framework mã nguồn mở trên ngôn ngữ Python.

Các thành phần của hệ thống:

- Discord App: Đây là ứng dụng Client được viết để tương tác và phản hồi tới nền tảng Discord. Ứng dụng yêu cầu xác thực token và cung cấp các phân quyền cần thiết để thực hiện một số chức năng ( đọc nội dung chat, phát âm thanh, kết nối tới kênh)
- Server-Side App: Server giao tiếp với Discord App bằng cách sử dụng socket để chuyển giao dữ liệu (TCP Message). Thành phần này sẽ đảm nhiệm chức năng chính là đọc file từ directory, xử lý và gửi cho Discord App



## 1.2. Các tiêu chí cần đạt được.

- Server và Client có thể giao tiếp với nhau qua socket.
- Server đọc được file mp3.
- Client xử lý dữ liệu file và phát nhạc.
- Client có handle một số thao tác của user ( play, pause, stop, resume)

## 2. Các thư viện cần cài đặt.

- Discord.py
- Socket
- Pathlib
- Pickle
- Python-dotenv

## 3. Cài đặt Server.

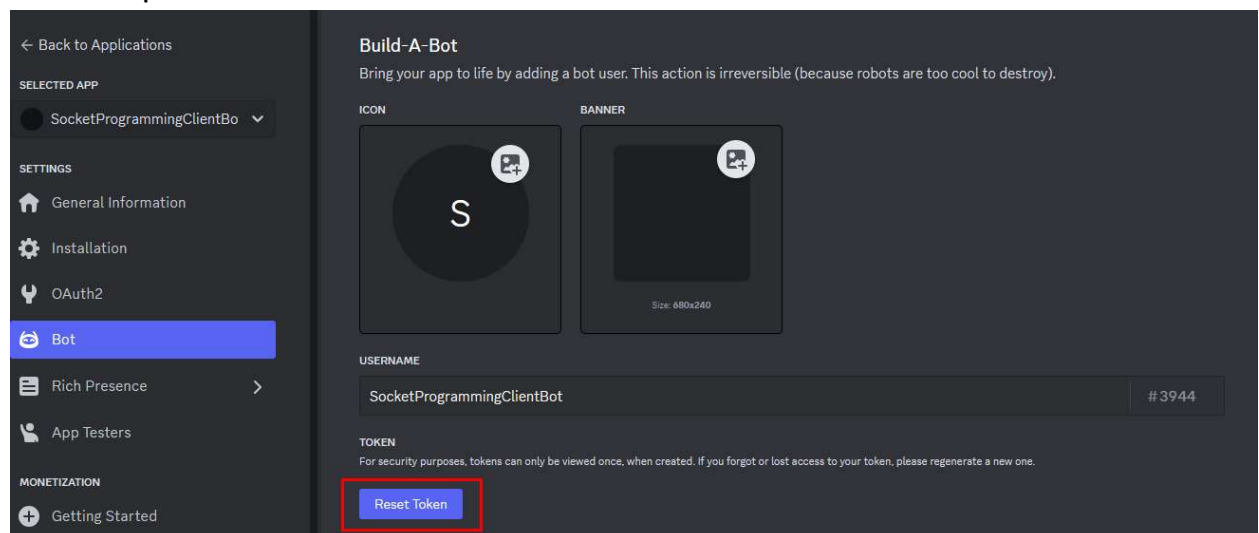
### server.py

```
1  # This is server code to send video and audio frames over TCP
2  import os.path
3  import pickle
4  import socket
5  import struct
6  from pathlib import Path
7
8
9  def server():
10     # khai báo host, port
11     host_name = socket.gethostname()
12     host_ip = socket.gethostbyname(host_name)
13     print(host_ip)
14     port = 9611
15     # chạy socket
16     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
17     s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
18     s.bind((socket.gethostname(), port))
19     s.listen(5)
20     # Server loop
21     while True:
22         clientsocket, address = s.accept()
23         print(f"Connection from {address} has been established")
24         clientsocket.send(bytes("Wellcome to the server!", "utf-8"))
25
26         msg = clientsocket.recv(1024) # nhận request từ client
27         song_name = clientsocket.recv(1024).decode("utf-8")
28         print(msg)
```

```
29         if msg == b"Search": # search request
30             os_path = './song/' + song_name + '.mp3'
31             print(os.path.exists(os_path))
32             if os.path.exists(os_path):
33                 clientsocket.send(bytes('200', "utf-8"))
34             else:
35                 clientsocket.send(bytes('404', "utf-8"))
36             clientsocket.close()
37
38         if msg == b"Play": #play request
39
40             os_path = './song/' + song_name + '.mp3'
41             print(os_path)
42             if os.path.exists(os_path):
43                 # xử lý file để gửi tới Client
44                 file_bytes = Path('./song/' + song_name +
45 '.mp3').read_bytes()
46                 data = file_bytes
47                 a = pickle.dumps(data)
48                 message = struct.pack("Q", len(a)) + a
49                 clientsocket.sendall(message)
50                 clientsocket.close()
51
52 if __name__ == '__main__':
53     server()
54
```

## 4. Cấu hình ứng dụng trên Discord Developer Portal.

### Bước 1: Tạo Token và lưu vào file .env



### Bước 2: Thêm bot vào Server với phân quyền phù hợp.

← Back to Applications

SELECTED APP

SocketProgrammingClientBo

SETTINGS

General Information

Installation

**OAuth2**

Bot

Rich Presence

App Testers

MONETIZATION

Getting Started

ACTIVITIES

Getting Started

OAuth2 URL Generator

Generate an invite link for your application by picking the scopes and permissions it needs to function. Then, share the URL to others!

SCOPES

☐ identify

☐ connections

☐ guilds.members.read

☐ rpc.notifications.read

☐ rpc.video.read

☐ rpc.screenshare.write

☐ webhook.incoming

☐ applications.builds.read

☐ applications.entitlements

☐ relationships.read

☐ dm\_channels.read

☐ presences.write

☐ applications.commands.permissions.update

☐ openid

☐ guilds

☐ gdm.join

☐ rpc.voice.read

☐ rpc.video.write

☐ rpc.activities.write

☐ messages.read

☐ applications.commands

☐ activities.read

☐ relationships.write

☐ role\_connections.write

☐ dm\_channels.messages.read

☐ email

☐ guilds.join

☐ rpc

☐ rpc.voice.write

☐ rpc.screenshare.read

☒ bot

☐ applications.builds.upload

☐ applications.store.update

☐ activities.write

☐ voice

☐ presences.read

☐ dm\_channels.messages.write

← Back to Applications

SELECTED APP

SocketProgrammingClientBo

SETTINGS

General Information

Installation

**OAuth2**

Bot

Rich Presence

App Testers

MONETIZATION

Getting Started

ACTIVITIES

Getting Started

BOT PERMISSIONS

GENERAL PERMISSIONS

☐ Administrator

☐ View Audit Log

☐ Manage Server

☐ Manage Roles

☐ Manage Channels

☐ Kick Members

☐ Ban Members

☐ Create Instant Invite

☐ Change Nickname

☐ Manage Nicknames

☐ Manage Expressions

☐ Create Expressions

☐ Manage Webhooks

☐ Read Messages/View Channels

☐ Manage Events

☐ Create Events

TEXT PERMISSIONS

☒ Send Messages

☐ Create Public Threads

☐ Create Private Threads

☐ Send Messages in Threads

☐ Send TTS Messages

☒ Manage Messages

☐ Manage Threads

☐ Embed Links

☐ Attach Files

☒ Read Message History

☐ Mention Everyone

☐ Use External Emojis

☐ Use External Stickers

☐ Add Reactions

☐ Use Slash Commands

☐ Use Embedded Activities

VOICE PERMISSIONS

☒ Connect

☒ Speak

☐ Video

☐ Mute Members

☐ Deafen Members

☐ Move Members

☒ Use Voice Activity

☐ Priority Speaker

☐ Request To Speak

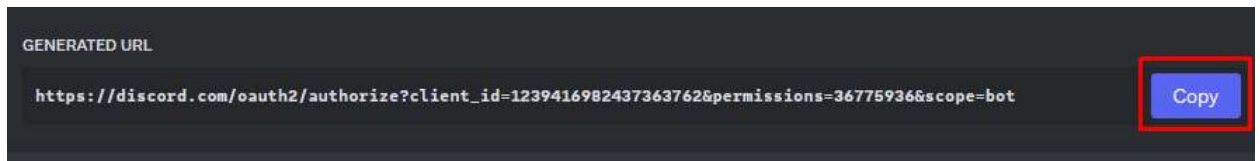
☐ Use Embedded Activities

☐ Use Soundboard

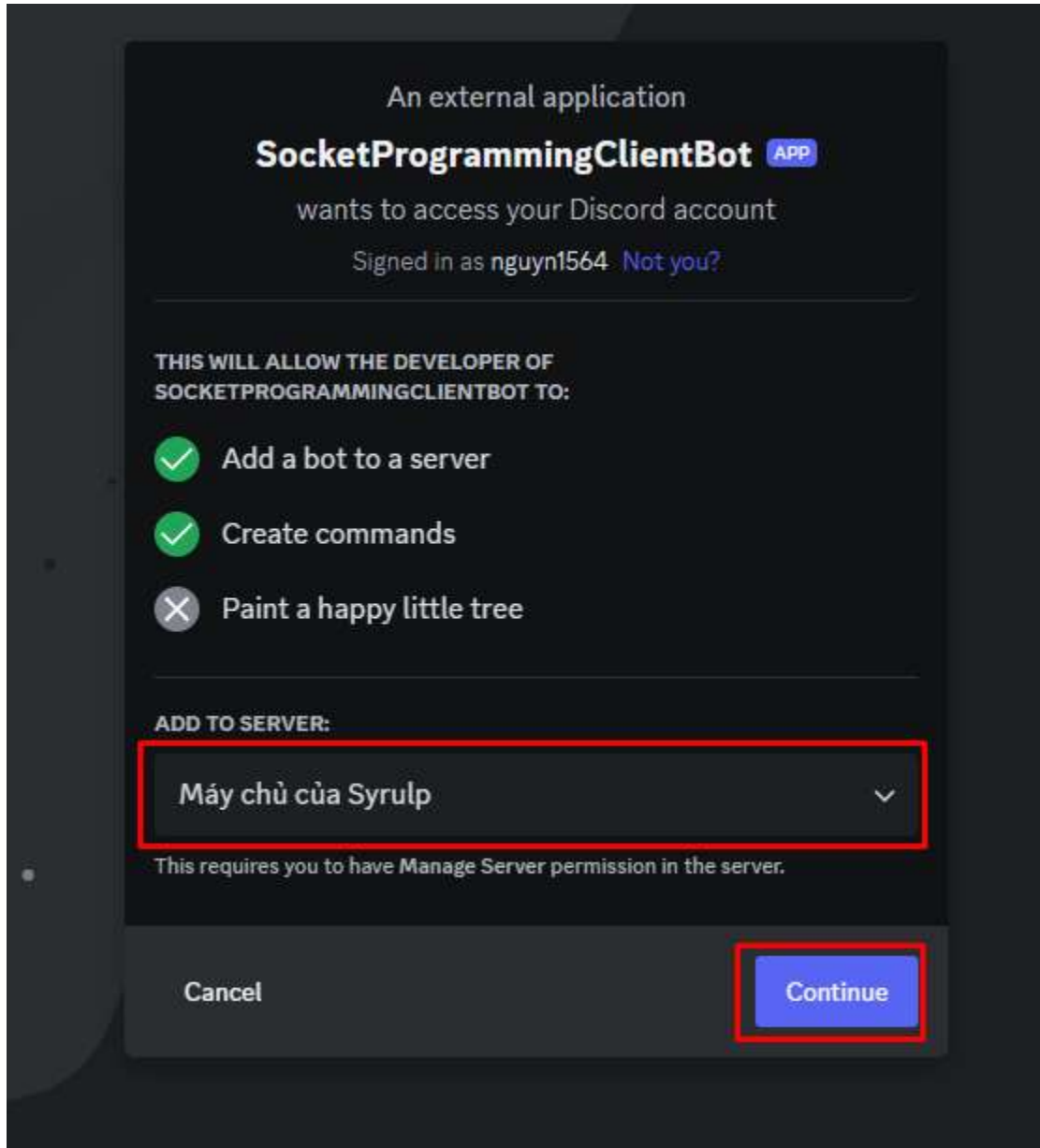
☐ Use External Sounds

☐ Create Polls

Sau đó copy invite url vào trình duyệt



Chọn Discord Server muốn thêm bot vào ( user phải có quyền Server Admin )



## 5. Cài đặt Client

DiscordBot.py

```
1 import os
2 import pickle
3 import socket
```

```
4     import struct
5     from pathlib import Path
6     from typing import Final
7
8     import discord
9     from discord.ext import commands
10    from dotenv import load_dotenv
11
12    host_name = socket.gethostname()
13    host_ip = socket.gethostbyname(host_name)
14    print(host_ip)
15    port = 9611
16
17    # Load Token from dot env
18
19    load_dotenv()
20    TOKEN: Final[str] = os.getenv('DISCORD_TOKEN')
21
22    # Bot Setup
23    intents = discord.Intents.default()
24    intents.message_content = True
25
26    bot = commands.Bot(command_prefix='$', intents=intents)
27    client = discord.Client(intents=intents)
28
29
30    def get_song_list():
31        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
32        s.connect((socket.gethostname(), port))
33        msg = s.recv(1024)
34        print(msg)
35        s.send(bytes("List", "utf-8"))
36        res = s.recv(1024)
37
38
39    def search_song(arg: str):
40        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
41        s.connect((socket.gethostname(), port))
42
43        msg = s.recv(1024)
44        print(msg)
45        s.send(bytes("Search", "utf-8"))
46        s.send(bytes(arg, "utf-8"))
47        res = s.recv(1024).decode("utf-8")
48        if res == '404':
49            return False
50        return True
51
52
53    def audio_stream(arg: str):
54        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
55        s.connect((socket.gethostname(), port))
56
57        msg = s.recv(1024)
58        print(msg)
59        s.send(bytes("Play", "utf-8"))
60        s.send(bytes(arg, "utf-8"))
```

```
61     data = b""
62     data2 = Path('./stream/' + 'streaming_file' + '.mp3')
63     payload_size = struct.calcsize("Q")
64     # stream = open("stream/test2.mp3", 'wb')
65     while True:
66         try:
67             while len(data) < payload_size:
68                 packet = s.recv(4 * 1024) # 4K
69                 if not packet:
70                     break
71                 data += packet
72             packed_msg_size = data[:payload_size]
73             data = data[payload_size:]
74             msg_size = struct.unpack("Q", packed_msg_size)[0]
75             while len(data) < msg_size:
76                 data += s.recv(4 * 1024)
77             frame_data = data[:msg_size]
78             data = data[msg_size:]
79             frame = pickle.loads(frame_data)
80             # stream.write(frame)
81             data2.write_bytes(frame)
82             s.send(b"close")
83         except:
84
85             break
86
87     s.close()
88     print('Audio Created')
89     return True
90
91
92 song_list = []
93
94
95 @bot.event
96 async def on_ready():
97     print('Bot is ready')
98
99
100 @bot.command()
101 async def test(ctx, arg):
102     await ctx.send(arg)
103
104
105 # play command
106 @bot.command(pass_context=True)
107 async def play(ctx, arg1: str):
108     print(f'You passed {arg1}')
109     voice_client = ctx.guild.voice_client
110     if ctx.author.voice:
111         channel = ctx.message.author.voice.channel
112         # voice_client = await channel.connect()
113         if voice_client.is_playing():
114             voice_client.stop()
115         if search_song(arg1):
116             audio_stream(arg1)
```



```
117
voice_client.play(discord.FFmpegPCMAudio("./stream/streaming_file.mp3"))
118     else:
119         await ctx.send(f"Không tìm thấy bài hát: {arg1}")
120     else:
121         await ctx.send("Bạn không trong kênh giọng nói nào cả")
122
123
124 @bot.command(pass_context=True)
125 async def connect(ctx):
126     if ctx.author.voice:
127         channel = ctx.message.author.voice.channel
128         await channel.connect()
129         await ctx.send("Đã kết nối")
130     else:
131         await ctx.send("Bạn không trong kênh giọng nói nào cả")
132
133
134 # disconnect command
135 @bot.command(pass_context=True)
136 async def disconnect(ctx):
137     await ctx.guild.voice_client.disconnect()
138     await ctx.send("Đã ngắt kết nối")
139     if os.path.exists("./stream/streaming_file.mp3"):
140         os.remove("./stream/streaming_file.mp3")
141
142
143 # pause command
144 @bot.command()
145 async def pause(ctx):
146     voice_client = ctx.guild.voice_client
147     if voice_client.is_playing():
148         voice_client.pause()
149
150     else:
151         await ctx.send("Không có bài hát nào đang được phát")
152
153
154 @bot.command()
155 async def resume(ctx):
156     voice_client = ctx.guild.voice_client
157     if voice_client.is_paused():
158         voice_client.resume()
159     else:
160         await ctx.send("Không có bài hát nào đang bị tạm dừng")
161
162
163 @bot.command()
164 async def stop(ctx):
165     voice_client = ctx.guild.voice_client
166     if voice_client.is_playing():
167         voice_client.stop()
168     else:
169         await ctx.send("Không có bài hát nào đang được phát")
170
171
172 bot.run(token=TOKEN)
```

