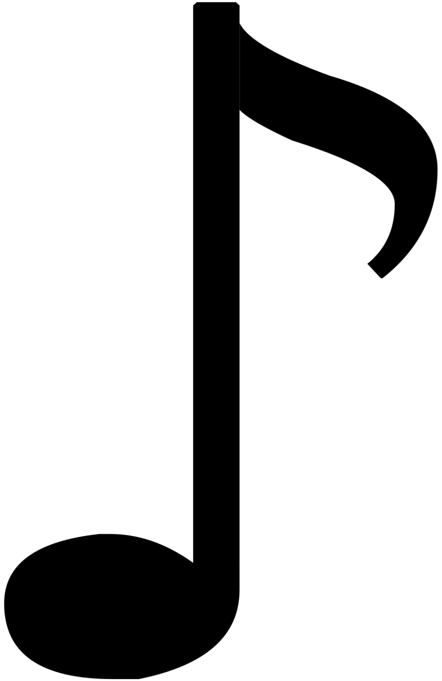
Popular Songs Database



Amira Muaket and Chris Sicle

Project's Description

- We have created a database that will print out popular songs of 2023
- By doing so, users will have an option of choosing a month they desire
- Once choosing the month it will print out the popular song of that month, as well as, the artist and genre
- Users also have the option of choosing and gaining the full list of popular songs where it will display the 12 months with the song, artist and genre

Project's Goal

- The goal of this project is to create an easy and effective way users can gain information.
- We have created a user friendly server that allows the option to gain insight about the top songs of each month while providing details of each song
- We also provided the user to have a choice in the matter. They can either choose a month for a quick look or obtain a full list of top songs for all months
- This allows for flexibility and personal preference

The details about the hardware

- We used a MacBook Air: Retina, 13-inch 2020
- The Processor: 1.1 GHz Quad-Core Intel Core i5
- The Graphics: Intel Iris Plus Graphics 1536 MB
- The Memory: 16 GB 3733 MHz LPDDR4X

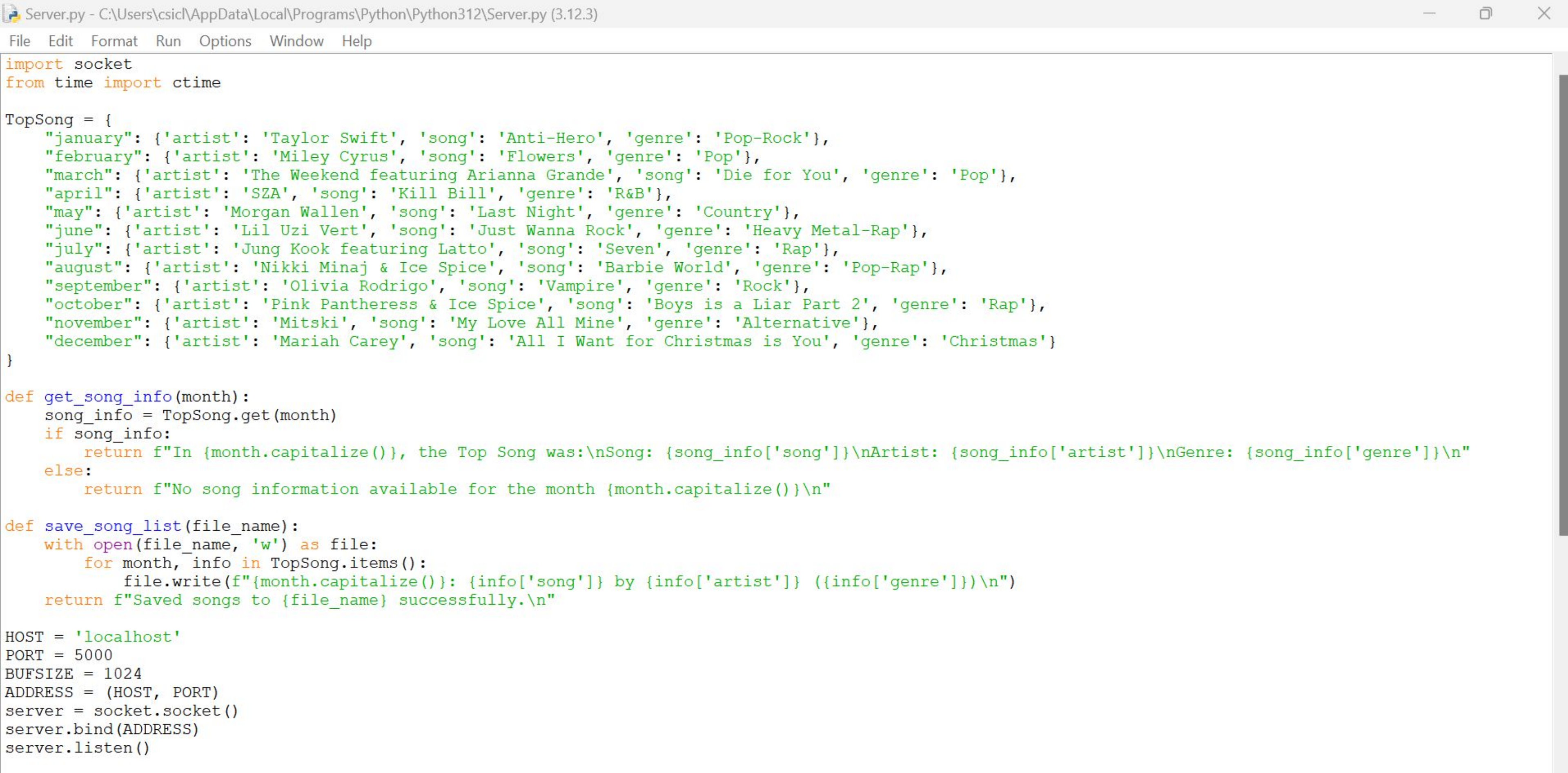
The details about the software

We used:

- IDLE Python 3.12 64
- Windows OS
- MacOS

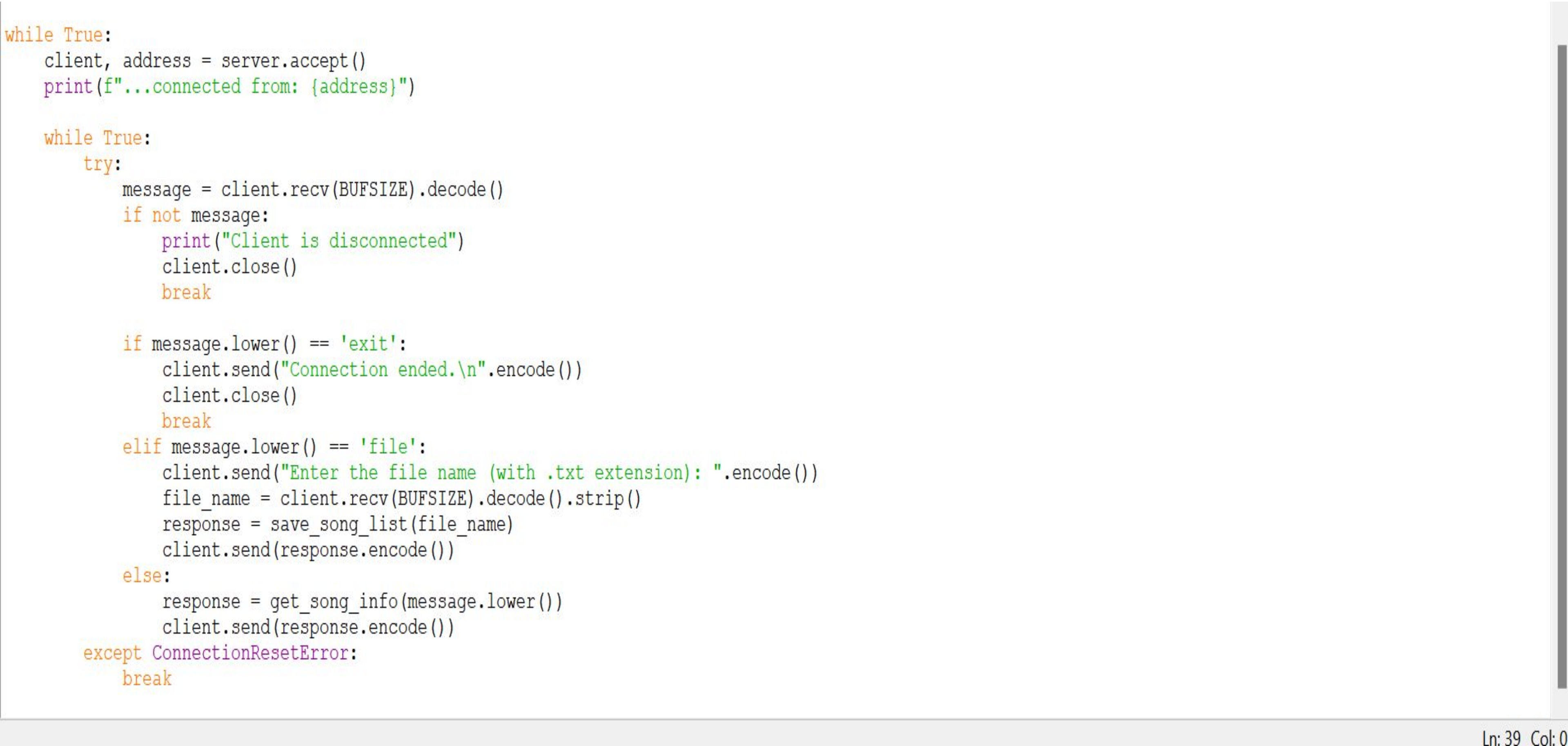
Our Full Code

Server code:


Server.py - C:\Users\csicl\AppData\Local\Programs\Python\Python312\Server.py (3.12.3)
File
Edit
Format
Run
Options
Window
Help

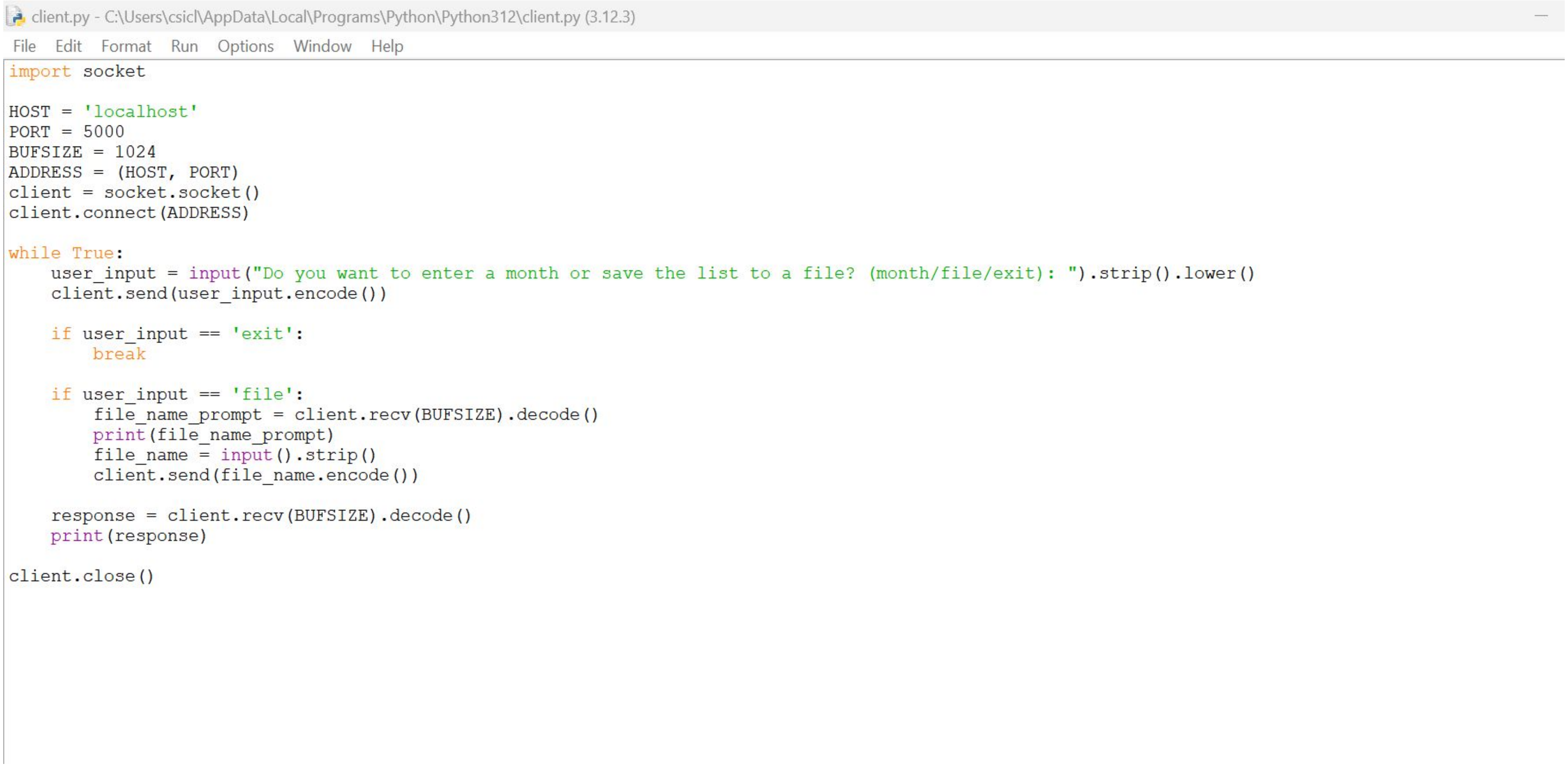
```python
import socket
from time import ctime

TopSong = {
    "january": {'artist': 'Taylor Swift', 'song': 'Anti-Hero', 'genre': 'Pop-Rock'},
    "february": {'artist': 'Miley Cyrus', 'song': 'Flowers', 'genre': 'Pop'},
    "march": {'artist': 'The Weekend featuring Arianna Grande', 'song': 'Die for You', 'genre': 'Pop'},
    "april": {'artist': 'SZA', 'song': 'Kill Bill', 'genre': 'R&B'},
    "may": {'artist': 'Morgan Wallen', 'song': 'Last Night', 'genre': 'Country'},
    "june": {'artist': 'Lil Uzi Vert', 'song': 'Just Wanna Rock', 'genre': 'Heavy Metal-Rap'},
    "july": {'artist': 'Jung Kook featuring Latto', 'song': 'Seven', 'genre': 'Rap'},
    "august": {'artist': 'Nikki Minaj & Ice Spice', 'song': 'Barbie World', 'genre': 'Pop-Rap'},
    "september": {'artist': 'Olivia Rodrigo', 'song': 'Vampire', 'genre': 'Rock'},
    "october": {'artist': 'Pink Pantheress & Ice Spice', 'song': 'Boys is a Liar Part 2', 'genre': 'Rap'},
    "november": {'artist': 'Mitski', 'song': 'My Love All Mine', 'genre': 'Alternative'},
    "december": {'artist': 'Mariah Carey', 'song': 'All I Want for Christmas is You', 'genre': 'Christmas'}
}

def get_song_info(month):
    song_info = TopSong.get(month)
    if song_info:
        return f"In {month.capitalize()}, the Top Song was:\nSong: {song_info['song']}\nArtist: {song_info['artist']}\nGenre: {song_info['genre']}\n"
    else:
        return f"No song information available for the month {month.capitalize()}\n"

def save_song_list(file_name):
    with open(file_name, 'w') as file:
        for month, info in TopSong.items():
            file.write(f"{month.capitalize()}: {info['song']} by {info['artist']} ({info['genre']})\n")
    return f"Saved songs to {file_name} successfully.\n"

HOST = 'localhost'
PORT = 5000
BUFSIZE = 1024
ADDRESS = (HOST, PORT)
server = socket.socket()
server.bind(ADDRESS)
server.listen()
```

Continued:



```
while True:
    client, address = server.accept()
    print(f"...connected from: {address}")

    while True:
        try:
            message = client.recv(BUFSIZE).decode()
            if not message:
                print("Client is disconnected")
                client.close()
                break

            if message.lower() == 'exit':
                client.send("Connection ended.\n".encode())
                client.close()
                break
            elif message.lower() == 'file':
                client.send("Enter the file name (with .txt extension): ".encode())
                file_name = client.recv(BUFSIZE).decode().strip()
                response = save_song_list(file_name)
                client.send(response.encode())
            else:
                response = get_song_info(message.lower())
                client.send(response.encode())
        except ConnectionResetError:
            break
```

Client Code:


client.py - C:\Users\csicl\AppData\Local\Programs\Python\Python312\client.py (3.12.3)
File
Edit
Format
Run
Options
Window
Help

```python
import socket

HOST = 'localhost'
PORT = 5000
BUFSIZE = 1024
ADDRESS = (HOST, PORT)
client = socket.socket()
client.connect(ADDRESS)

while True:
    user_input = input("Do you want to enter a month or save the list to a file? (month/file/exit): ").strip().lower()
    client.send(user_input.encode())

    if user_input == 'exit':
        break

    if user_input == 'file':
        file_name_prompt = client.recv(BUFSIZE).decode()
        print(file_name_prompt)
        file_name = input().strip()
        client.send(file_name.encode())

    response = client.recv(BUFSIZE).decode()
    print(response)

client.close()
```

Break down of our code

Loops Used:

Server Code:

- while True:

      client, address = server.accept()
  (Line 40 -41)
- while True:
  (line 42-63)

Client Code:

- while True:
  (Line 10-21)

Functions used:

Server Code:

- def get_song_info(month):
  (Line 19)
- def save_song_list(file_name):
  (Line 26)

Lists and Dictionaries used:

Server Code:

-
  ```
  TopSong = {
      "january": {...},
      "february": {...},
  ```
  (Line 4-16)

Files used:

Server Code:

-
  ```
  def save_song_list(file_name):
      with open(file_name, 'w') as file:
  ```
  (Line 26-27)

Strings used:

Server Code:

-
  ```
  def get_song_info(month):
      song_info = TopSong.get(month)
      if song_info:
          return f"In {month.capitalize()}, the Top Song
  was:\nSong: {song_info['song']}\nArtist:
  {song_info['artist']}\nGenre: {song_info['genre']}\n"
      else:
          return f"No song information available for the month
  {month.capitalize()}\n"
  ```

  (Line 19-24)

Strings Continued:

Client Code:

- user_input = input("Do you want to enter a month or save the list to a file? (month/file/exit): ").strip().lower()
    (Line 11)

Socket Programming Used:

Server Code:

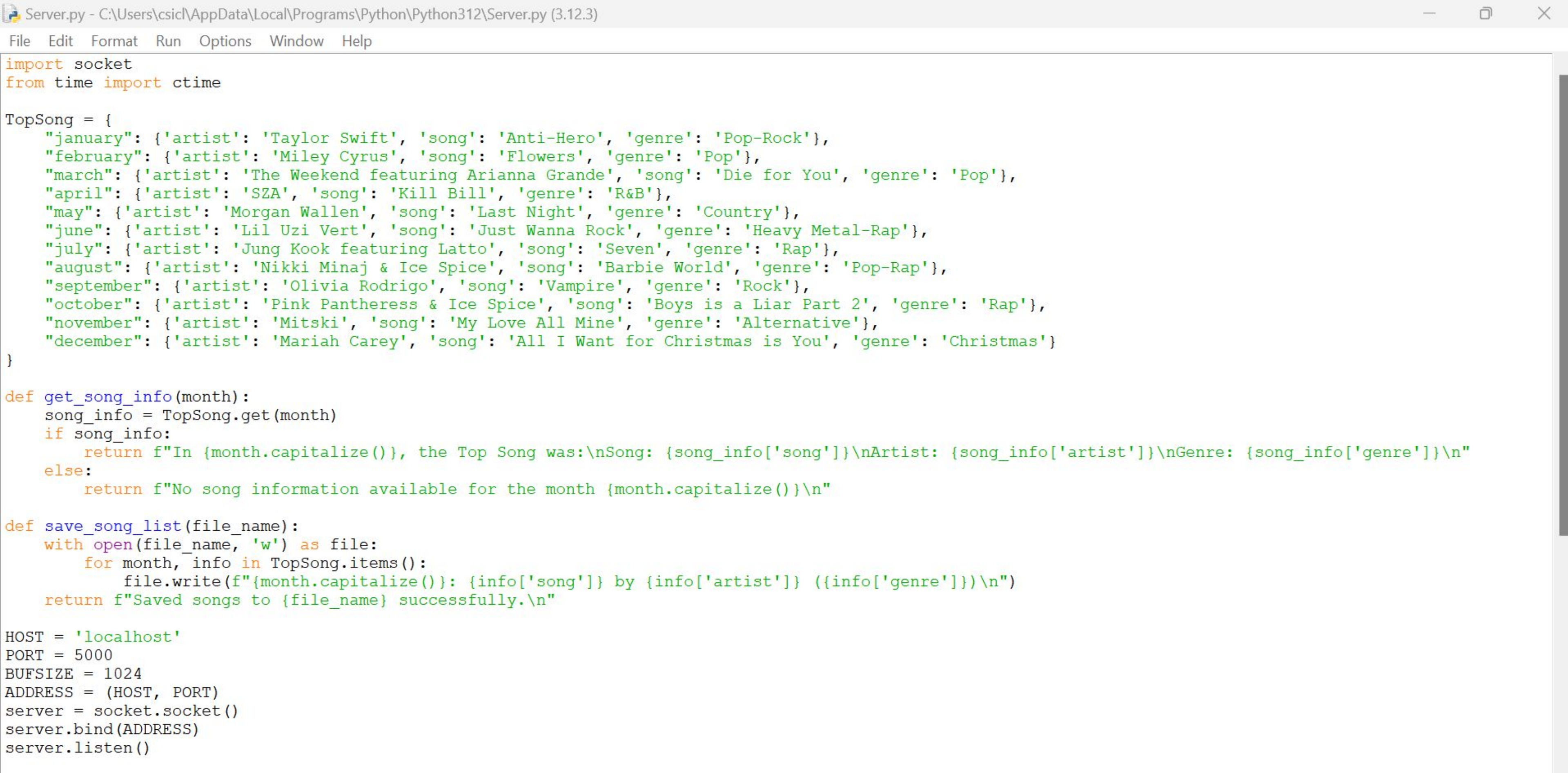
- server = socket.socket()
  server.bind(ADDRESS)
  server.listen()
  (Line 36-38)

- client, address = server.accept()
  (Line 41)

Client Code:

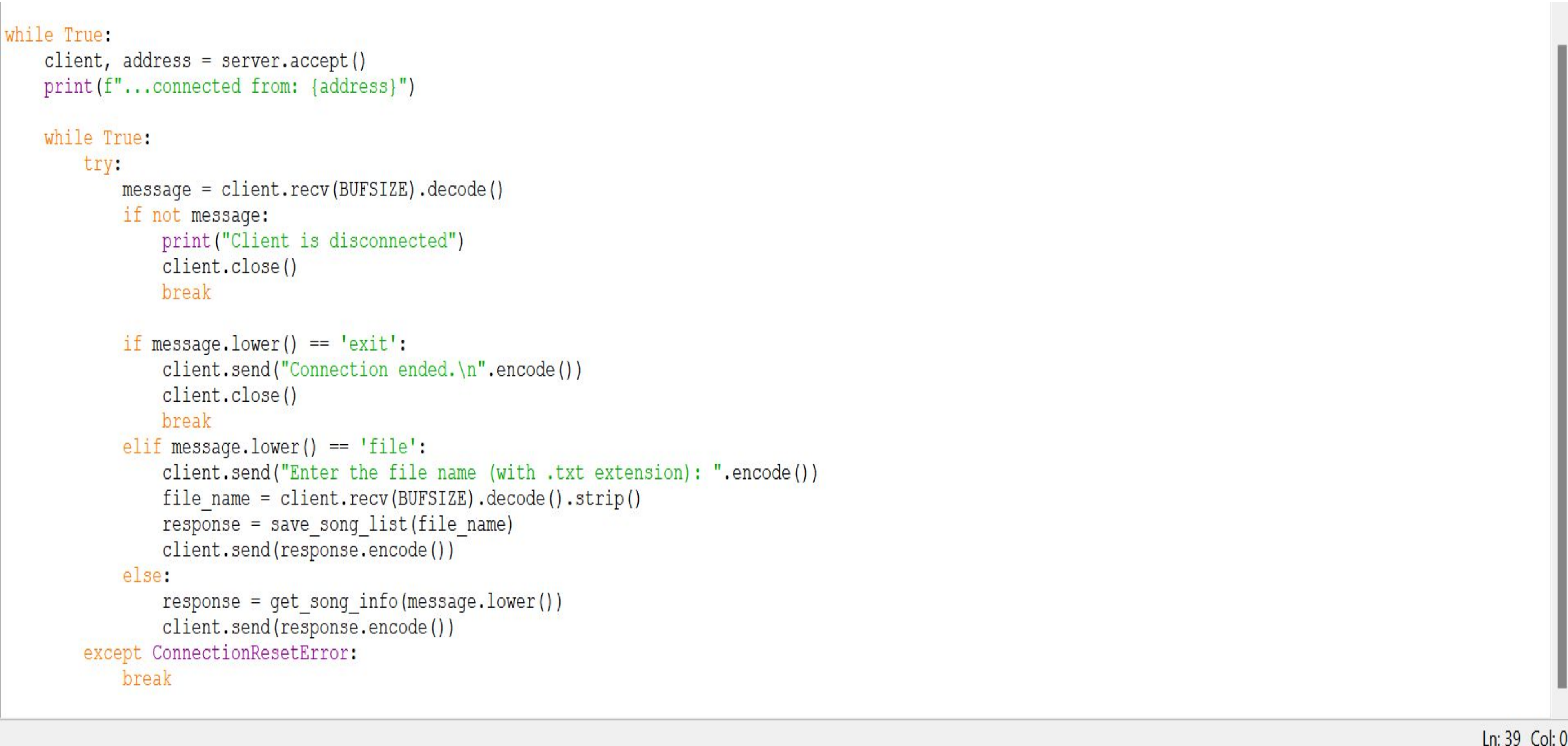
- client = socket.socket()
- client.connect(ADDRESS)
  (Line 7-8)

Server code:


Server.py - C:\Users\csicl\AppData\Local\Programs\Python\Python312\Server.py (3.12.3)
File
Edit
Format
Run
Options
Window
Help

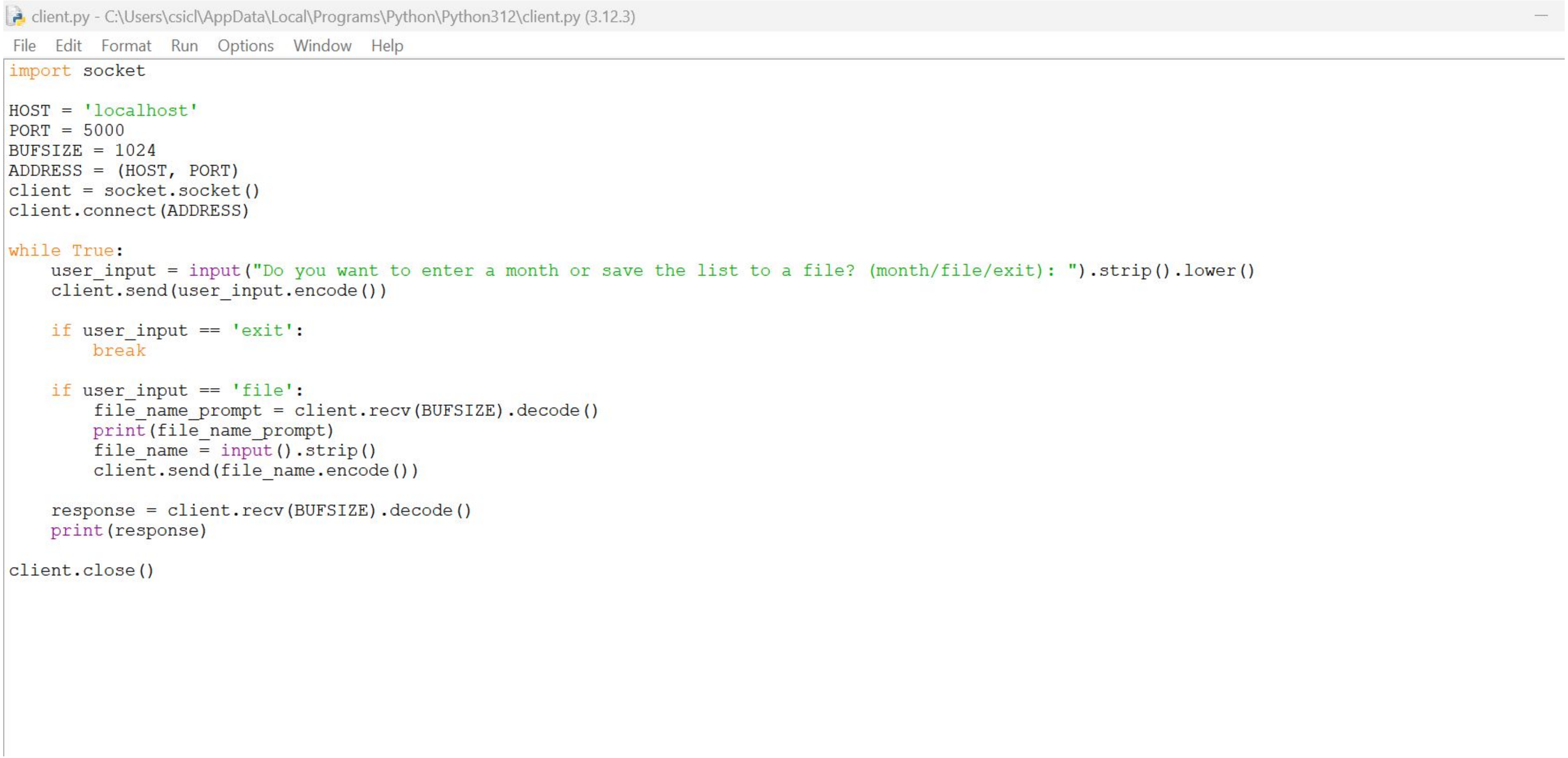
```python
import socket
from time import ctime

TopSong = {
    "january": {'artist': 'Taylor Swift', 'song': 'Anti-Hero', 'genre': 'Pop-Rock'},
    "february": {'artist': 'Miley Cyrus', 'song': 'Flowers', 'genre': 'Pop'},
    "march": {'artist': 'The Weekend featuring Arianna Grande', 'song': 'Die for You', 'genre': 'Pop'},
    "april": {'artist': 'SZA', 'song': 'Kill Bill', 'genre': 'R&B'},
    "may": {'artist': 'Morgan Wallen', 'song': 'Last Night', 'genre': 'Country'},
    "june": {'artist': 'Lil Uzi Vert', 'song': 'Just Wanna Rock', 'genre': 'Heavy Metal-Rap'},
    "july": {'artist': 'Jung Kook featuring Latto', 'song': 'Seven', 'genre': 'Rap'},
    "august": {'artist': 'Nikki Minaj & Ice Spice', 'song': 'Barbie World', 'genre': 'Pop-Rap'},
    "september": {'artist': 'Olivia Rodrigo', 'song': 'Vampire', 'genre': 'Rock'},
    "october": {'artist': 'Pink Pantheress & Ice Spice', 'song': 'Boys is a Liar Part 2', 'genre': 'Rap'},
    "november": {'artist': 'Mitski', 'song': 'My Love All Mine', 'genre': 'Alternative'},
    "december": {'artist': 'Mariah Carey', 'song': 'All I Want for Christmas is You', 'genre': 'Christmas'}
}

def get_song_info(month):
    song_info = TopSong.get(month)
    if song_info:
        return f"In {month.capitalize()}, the Top Song was:\nSong: {song_info['song']}\nArtist: {song_info['artist']}\nGenre: {song_info['genre']}\n"
    else:
        return f"No song information available for the month {month.capitalize()}\n"

def save_song_list(file_name):
    with open(file_name, 'w') as file:
        for month, info in TopSong.items():
            file.write(f"{month.capitalize()}: {info['song']} by {info['artist']} ({info['genre']})\n")
    return f"Saved songs to {file_name} successfully.\n"

HOST = 'localhost'
PORT = 5000
BUFSIZE = 1024
ADDRESS = (HOST, PORT)
server = socket.socket()
server.bind(ADDRESS)
server.listen()
```

Continued:



```
while True:
    client, address = server.accept()
    print(f"...connected from: {address}")

    while True:
        try:
            message = client.recv(BUFSIZE).decode()
            if not message:
                print("Client is disconnected")
                client.close()
                break

            if message.lower() == 'exit':
                client.send("Connection ended.\n".encode())
                client.close()
                break
            elif message.lower() == 'file':
                client.send("Enter the file name (with .txt extension): ".encode())
                file_name = client.recv(BUFSIZE).decode().strip()
                response = save_song_list(file_name)
                client.send(response.encode())
            else:
                response = get_song_info(message.lower())
                client.send(response.encode())
        except ConnectionResetError:
            break
```

Client Code:


client.py - C:\Users\csicl\AppData\Local\Programs\Python\Python312\client.py (3.12.3)
File Edit Format Run Options Window Help

```python
import socket

HOST = 'localhost'
PORT = 5000
BUFSIZE = 1024
ADDRESS = (HOST, PORT)
client = socket.socket()
client.connect(ADDRESS)

while True:
    user_input = input("Do you want to enter a month or save the list to a file? (month/file/exit): ").strip().lower()
    client.send(user_input.encode())

    if user_input == 'exit':
        break

    if user_input == 'file':
        file_name_prompt = client.recv(BUFSIZE).decode()
        print(file_name_prompt)
        file_name = input().strip()
        client.send(file_name.encode())

    response = client.recv(BUFSIZE).decode()
    print(response)

client.close()
```

Conclusion

Overall our Popular Song Database has displayed the python implementations that allow users to accesses any needed information and data. By providing the user readability, efficiency and simplicity. Our code has displayed socket programming, loops, functions, files, strings, lists and dictionaries.