Tribhuvan University

National College of Computer Studies

Paknajol, Kathmandu

Report on:

Voice Assistant System

SUNA

Submitted by:

Aastha Neupane

Bishal Phuyal

Rashika Putuwar

Ravi Singh

Ritina Maharjan

Sheetal Gurung

Acknowledgement

The final outcome of this assessment required a lot of guidance and assistance. It is a genuine pleasure to convey out special thanks of gratitude to our module leader who gave us this opportunity. Working in a good environment and motivation enhance the quality of the work and we get it from our college through this project.

We have done research through various sources like old projects, internet, etc. We are thankful to our teachers to help in our projects.

Abstract

This report is related to *Voice Assistant System Suna* using Python. Python is a general-purposes programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured, object-oriented and functional programming.

Contents

1. Introduction

With the advent of music applications such as Spotify and Apple music, the world is spinning faster than ever before. People do not have patience to search for new songs and find it very convenient for a system to recommend songs to them. While there are options to go to an application and say out a song name or search similar songs.

While it comes to song recommendations, there are two ways to go about, either a voice-based filtering system or a content-based filtering.

This our online based system, which recognizes the speech and recommends the song according to the song name or humming or lyrics of the song. In this system, user can manually ask the song by typing too.

2. Objectives

- Responsive way to interact with AI Chatbot
- To let user enjoy voice search
- To make site adaptable to blind people
- To recommend music
- To provide one place for the most of voice AI tools

3. Features

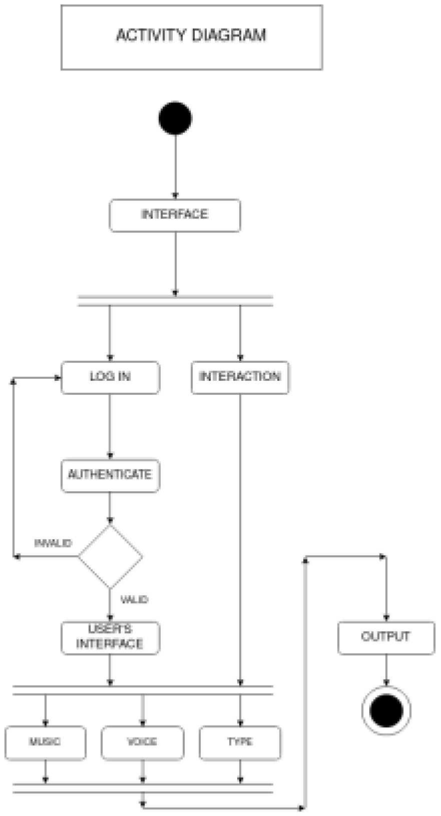
- Command through voice /text
- Search music through voice/test
- Music recommendation
- Single Page Application(SPA)
- User friendly color and background

4. Limitations

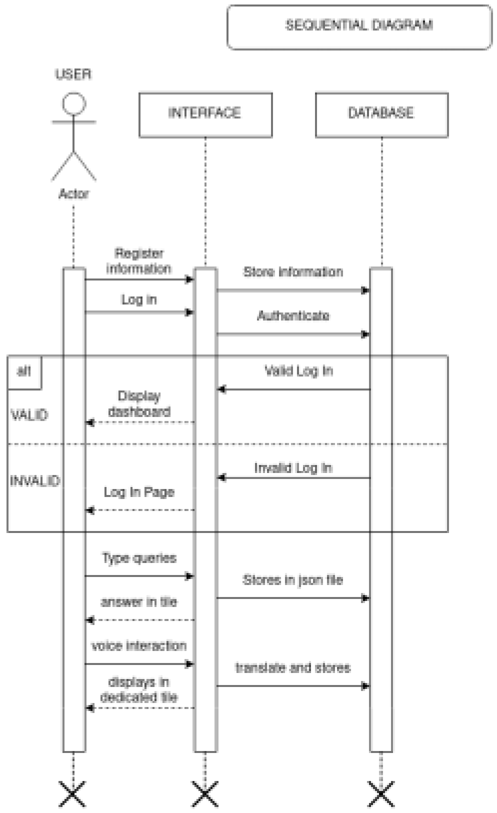
- There may be some occurrence of problems in website.
- Might make common grammatical errors
- Makes communication less personal
- Privacy concerns

5. Software Development

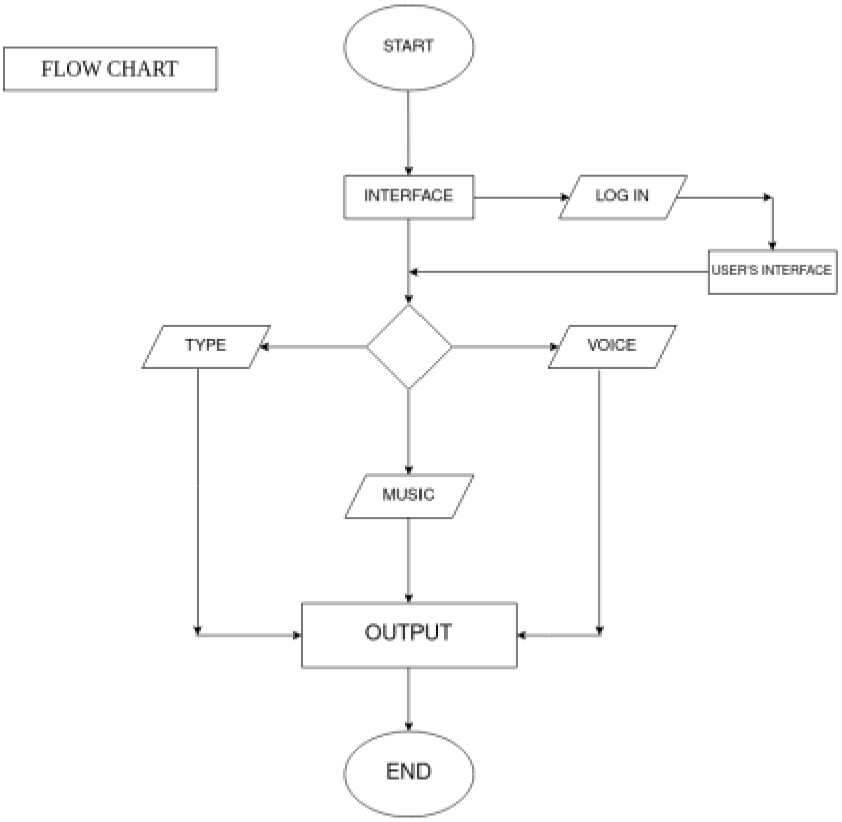
5.1 Activity Diagram



5.2 Sequential Diagram

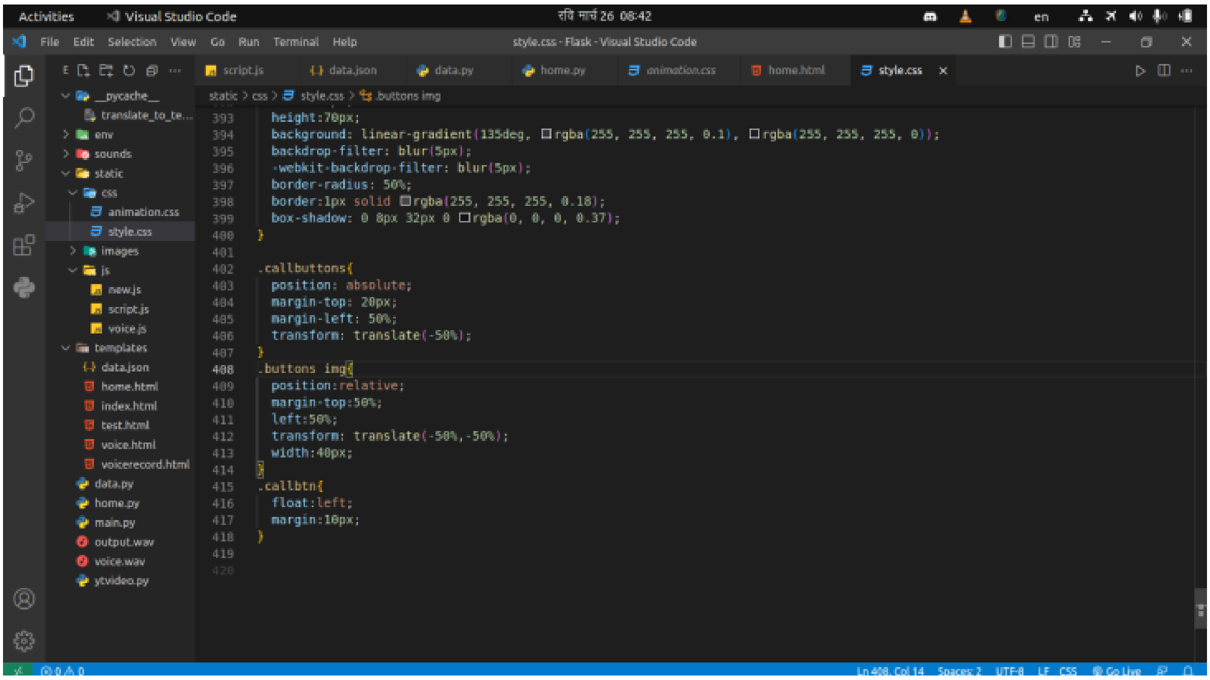


5.3 Flow Chart

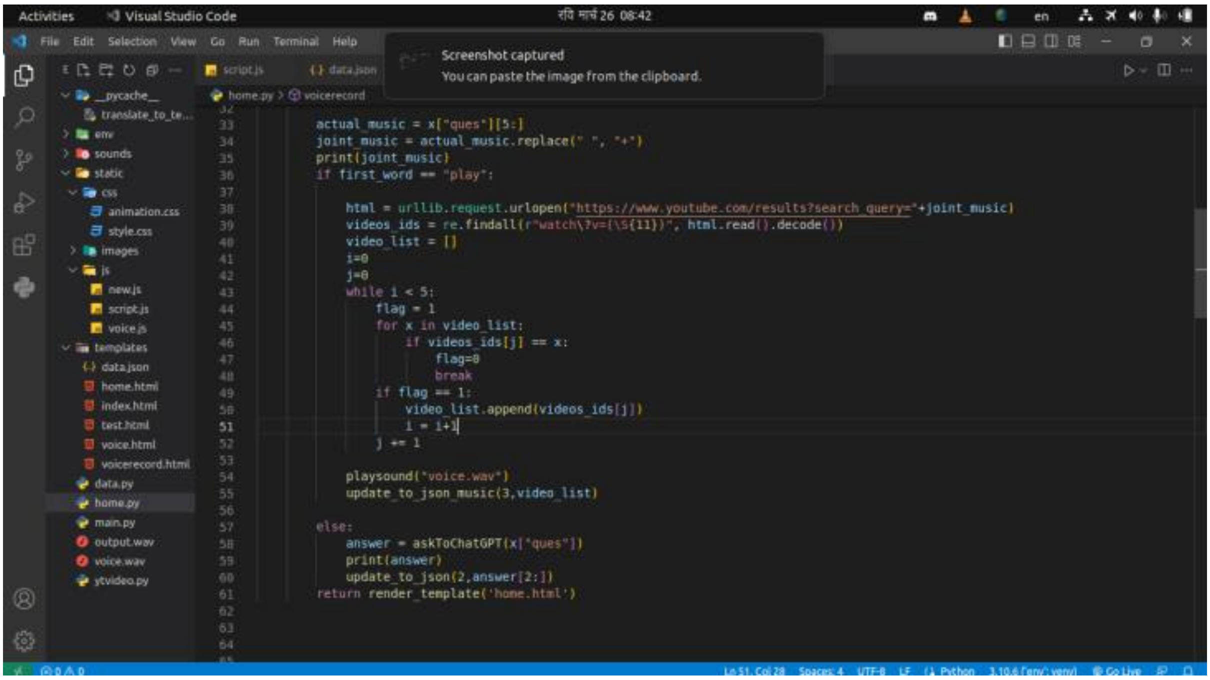


6. Codes and Screenshots

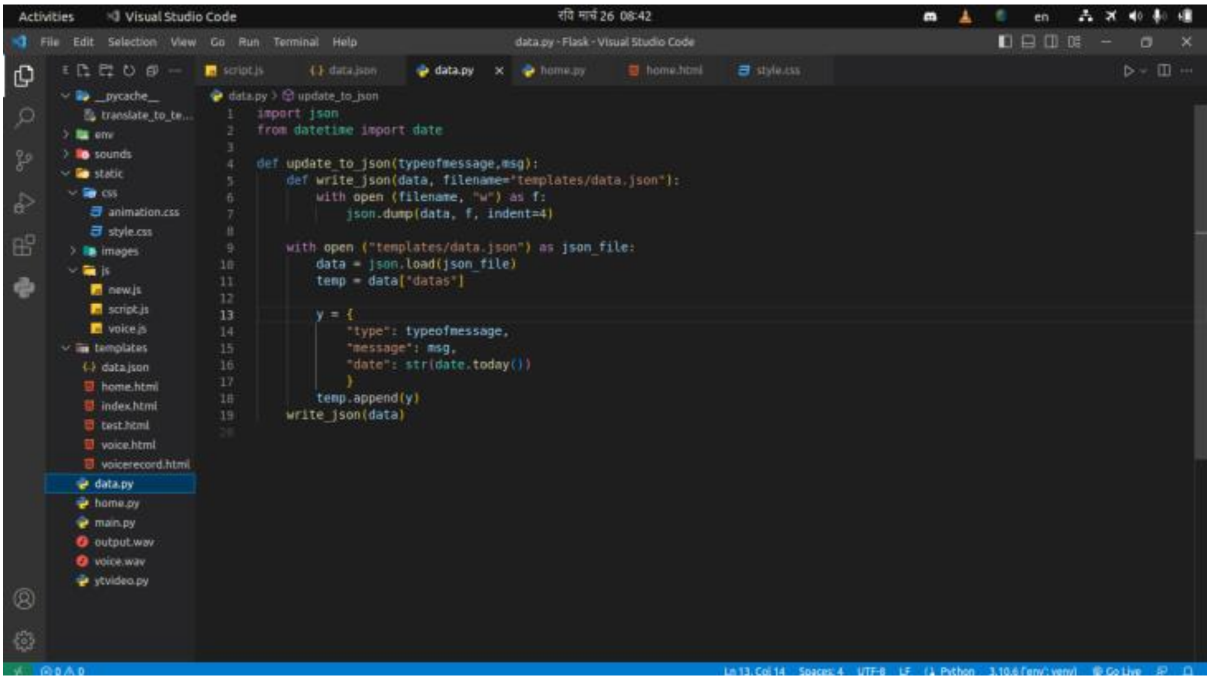
6.1 Codes



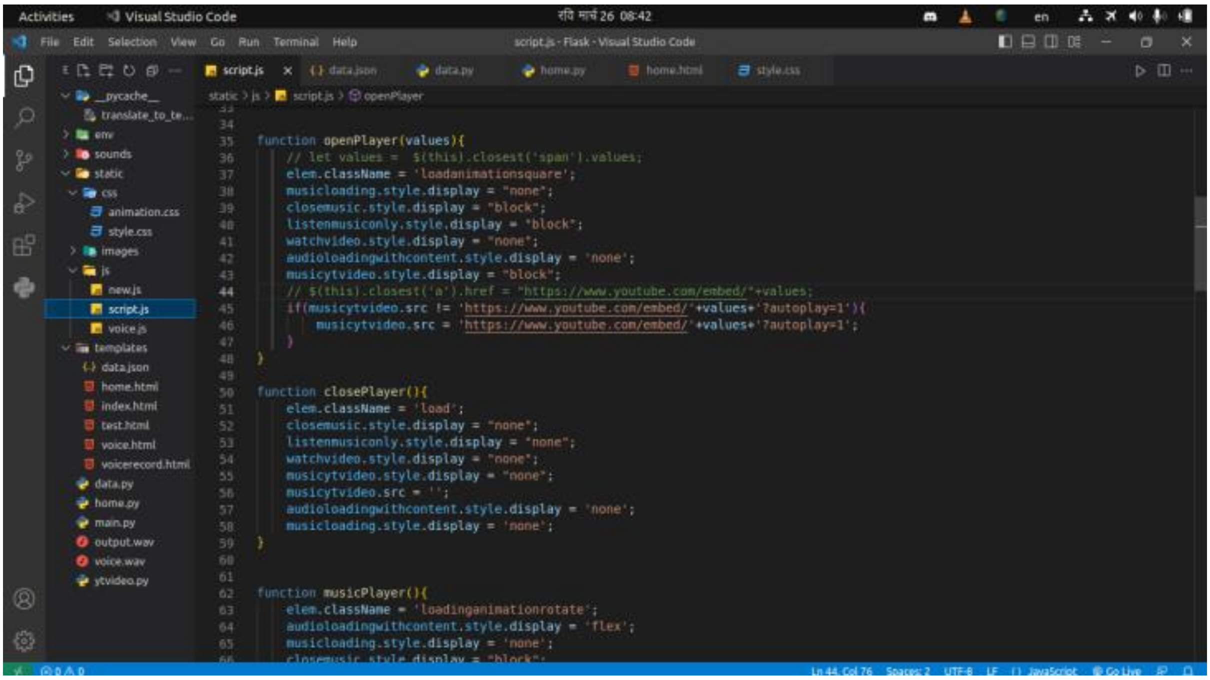
```css
  height:70px;
  background: linear-gradient(135deg, rgba(255, 255, 255, 0.1), rgba(255, 255, 255, 0));
  backdrop-filter: blur(5px);
  -webkit-backdrop-filter: blur(5px);
  border-radius: 50%;
  border:1px solid rgba(255, 255, 255, 0.18);
  box-shadow: 0 8px 32px 0 rgba(0, 0, 0, 0.37);
}

.callbuttons{
  position: absolute;
  margin-top: 20px;
  margin-left: 50%;
  transform: translate(-50%);
}
.buttons img{
  position:relative;
  margin-top:50%;
  left:50%;
  transform: translate(-50%,-50%);
  width:40px;
}
.callbtn{
  float:left;
  margin:10px;
}
```



```python
        actual_music = x["ques"][5:]
        joint_music = actual_music.replace(" ", "+")
        print(joint_music)
        if first_word == "play":

            html = urllib.request.urlopen("https://www.youtube.com/results?search_query="+joint_music)
            videos_ids = re.findall(r"watch\?v=(\S{11})", html.read().decode())
            video_list = []
            i=0
            j=0
            while i < 5:
                flag = 1
                for x in video_list:
                    if videos_ids[j] == x:
                        flag=0
                        break
                if flag == 1:
                    video_list.append(videos_ids[j])
                    i = i+1
                j += 1

            playsound("voice.wav")
            update_to_json_music(3,video_list)

        else:
            answer = askToChatGPT(x["ques"])
            print(answer)
            update_to_json(2,answer[2:])
        return render_template('home.html')
```

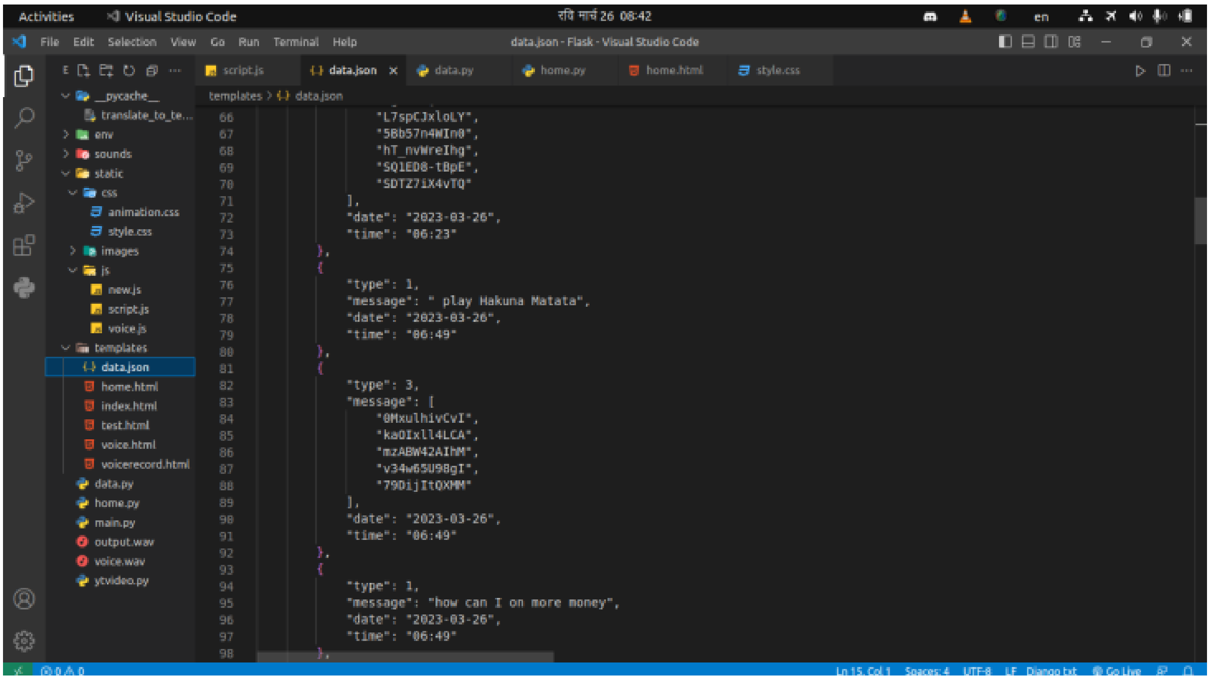


```python
import json
from datetime import date

def update_to_json(typeofmessage,msg):
    def write_json(data, filename="templates/data.json"):
        with open (filename, "w") as f:
            json.dump(data, f, indent=4)

    with open ("templates/data.json") as json_file:
        data = json.load(json_file)
        temp = data["datas"]

        y = {
            "type": typeofmessage,
            "message": msg,
            "date": str(date.today())
            }
        temp.append(y)
    write_json(data)
```

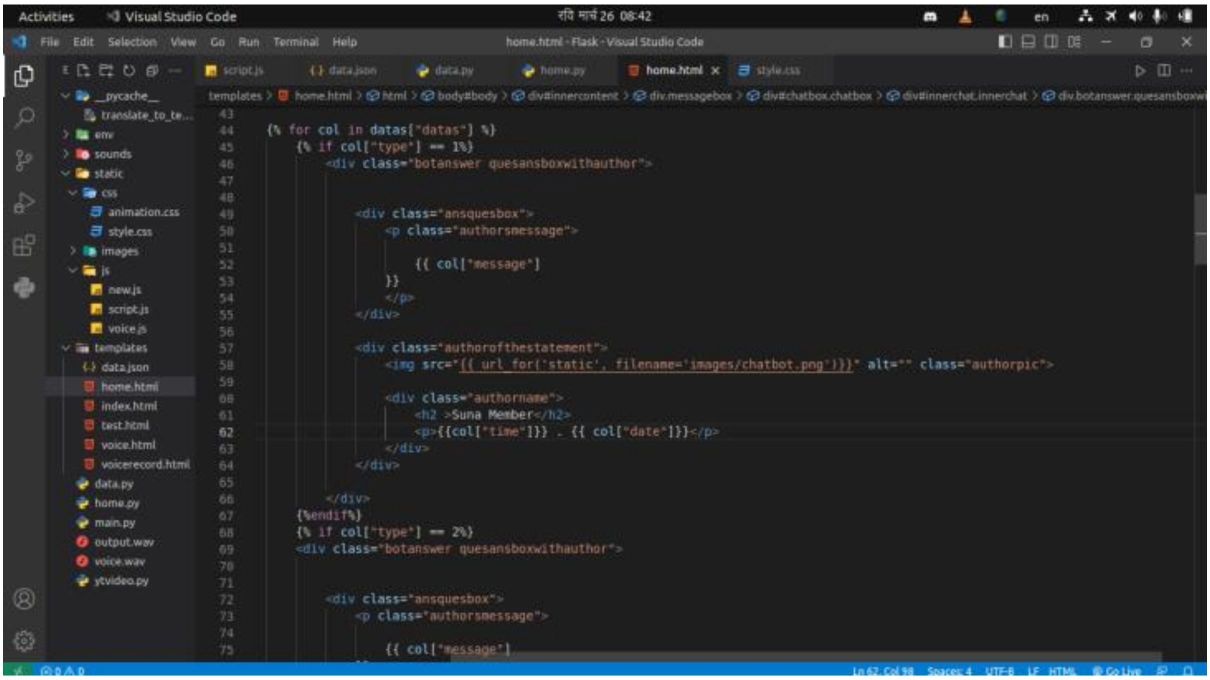


script.js - Flask - Visual Studio Code
static > js > script.js > openPlayer

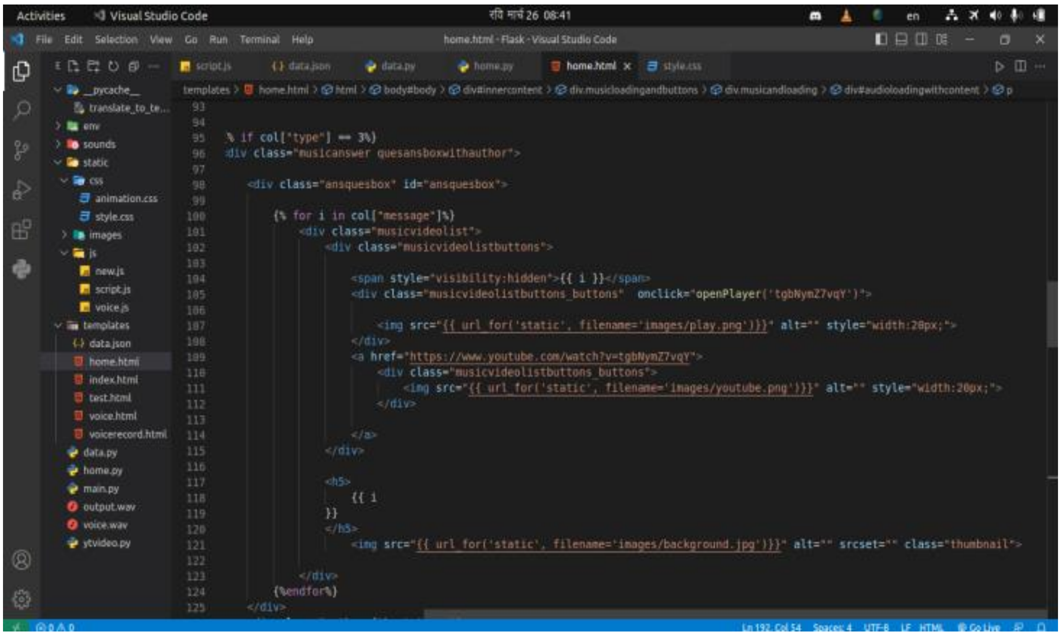
```javascript

function openPlayer(values){
    // let values =  $(this).closest('span').values;
    elem.className = 'loadanimationsquare';
    musicloading.style.display = "none";
    closemusic.style.display = "block";
    listenmusiconly.style.display = "block";
    watchvideo.style.display = "none";
    audioloadingwithcontent.style.display = 'none';
    musicytvideo.style.display = "block";
    // $(this).closest('a').href = "https://www.youtube.com/embed/"+values;
    if(musicytvideo.src != 'https://www.youtube.com/embed/'+values+'?autoplay=1'){
        musicytvideo.src = 'https://www.youtube.com/embed/'+values+'?autoplay=1';
    }
}

function closePlayer(){
    elem.className = 'load';
    closemusic.style.display = "none";
    listenmusiconly.style.display = "none";
    watchvideo.style.display = "none";
    musicytvideo.style.display = "none";
    musicytvideo.src = '';
    audioloadingwithcontent.style.display = 'none';
    musicloading.style.display = 'none';
}


function musicPlayer(){
    elem.className = 'loadinganimationrotate';
    audioloadingwithcontent.style.display = 'flex';
    musicloading.style.display = 'none';
    closemusic.style.display = "block";
```



```
                "L7spCJxloLY",
                "5Bb57n4WIn0",
                "hT_nvWreIhg",
                "SQ1ED8-tBpE",
                "SDTZ7iX4vTQ"
            ],
            "date": "2023-03-26",
            "time": "06:23"
        },
        {
            "type": 1,
            "message": " play Hakuna Matata",
            "date": "2023-03-26",
            "time": "06:49"
        },
        {
            "type": 3,
            "message": [
                "0MxulhivCvI",
                "ka0Ixll4LCA",
                "mzABW42AIhM",
                "v34w65U98gI",
                "79DijItQXMM"
            ],
            "date": "2023-03-26",
            "time": "06:49"
        },
        {
            "type": 1,
            "message": "how can I on more money",
            "date": "2023-03-26",
            "time": "06:49"
        },
```

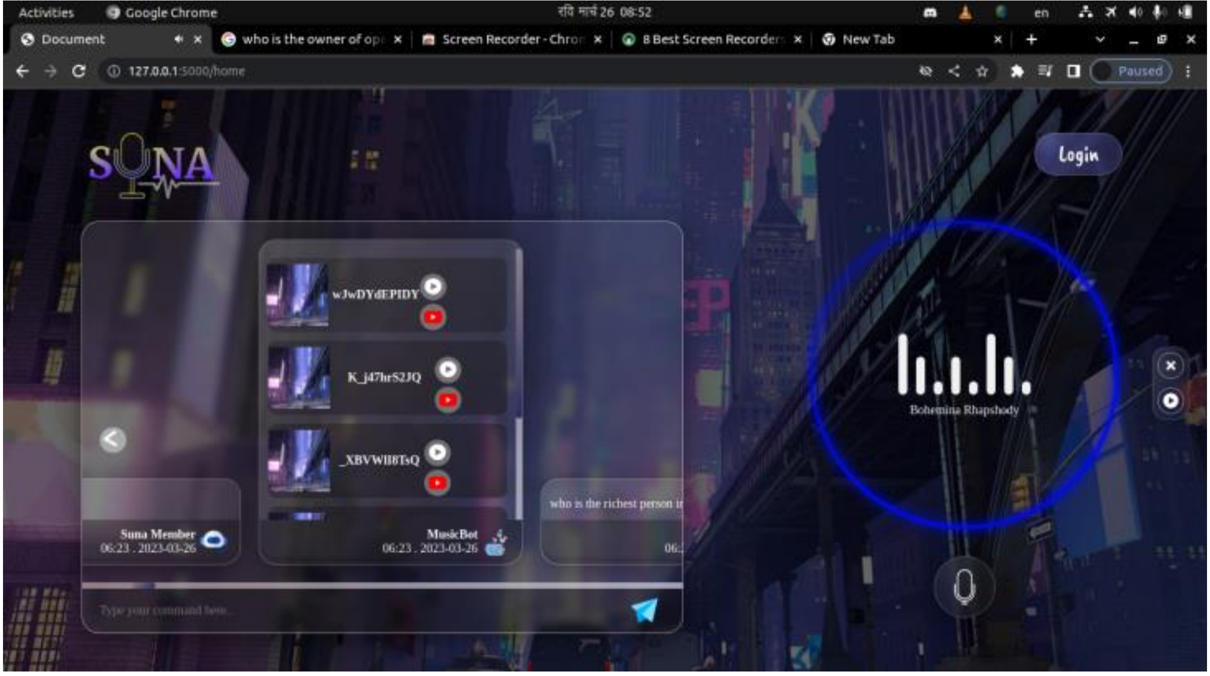


```

{% for col in datas["datas"] %}
    {% if col["type"] == 1%}
        <div class="botanswer quesansboxwithauthor">


            <div class="ansquesbox">
                <p class="authorsmessage">

                    {{ col["message"]
                }}
                </p>
            </div>

            <div class="authorofthestatement">
                <img src="{{ url_for('static', filename='images/chatbot.png')}}" alt="" class="authorpic">

                <div class="authorname">
                    <h2 >Suna Member</h2>
                    <p>{{col["time"]}} . {{ col["date"]}}</p>
                </div>
            </div>

        </div>
    {%endif%}
    {% if col["type"] == 2%}
    <div class="botanswer quesansboxwithauthor">


        <div class="ansquesbox">
            <p class="authorsmessage">

                {{ col["message"]
```



```
% if col["type"] == 3%)
div class="musicanswer quesansboxwithauthor">

    <div class="ansquesbox" id="ansquesbox">

        {% for i in col["message"]%}
            <div class="musicvideolist">
                <div class="musicvideolistbuttons">

                    <span style="visibility:hidden">{{ i }}</span>
                    <div class="musicvideolistbuttons_buttons"  onclick="openPlayer('tgbNymZ7vqY')">

                        <img src="{{ url_for('static', filename='images/play.png')}}" alt="" style="width:20px;">
                    </div>
                    <a href="https://www.youtube.com/watch?v=tgbNymZ7vqY">
                        <div class="musicvideolistbuttons_buttons">
                            <img src="{{ url_for('static', filename='images/youtube.png')}}" alt="" style="width:20px;">
                        </div>

                    </a>
                </div>

                <h5>
                    {{ i
                }}
                </h5>
                    <img src="{{ url_for('static', filename='images/background.jpg')}}" alt="" srcset="" class="thumbnail">

            </div>
        {%endfor%}
    </div>
```

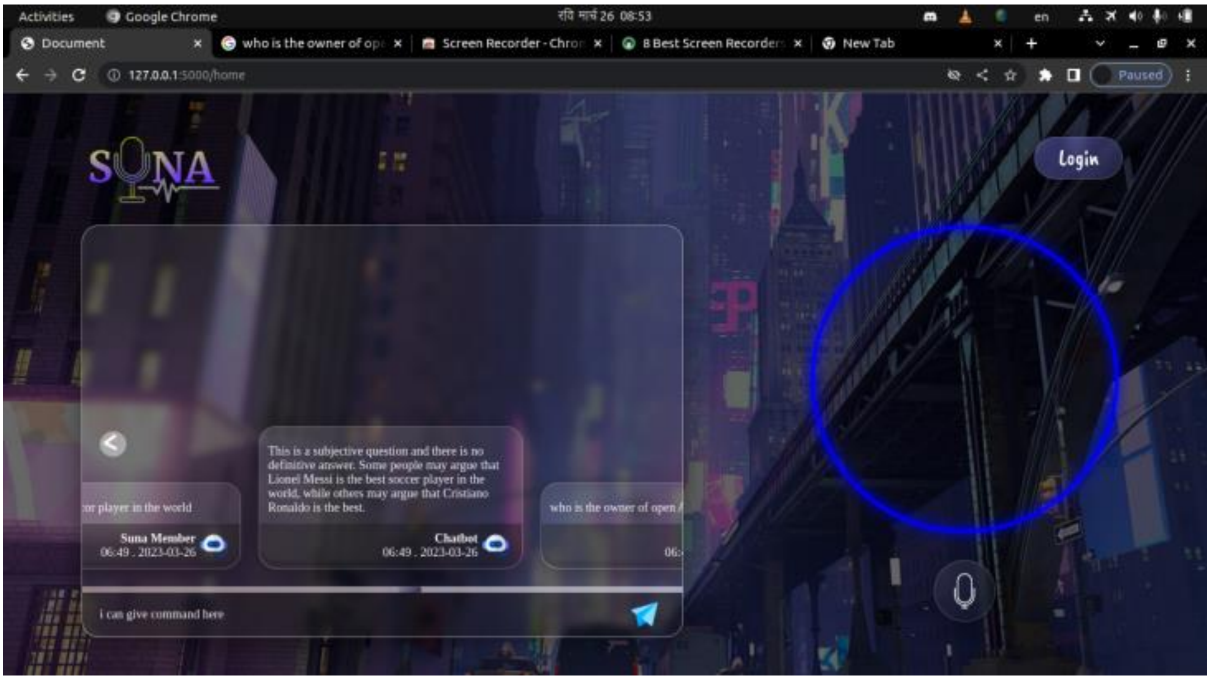
6.2 Outputs



Suna Member 06:23 . 2023-03-26

MusicBot 06:23 . 2023-03-26

who is the richest person i



:er player in the world

Suna Member 06:49 . 2023-03-26

This is a subjective question and there is no definitive answer. Some people may argue that Lionel Messi is the best soccer player in the world, while others may argue that Cristiano Ronaldo is the best.

Chatbot 06:49 . 2023-03-26

who is the owner of open

i can give command here

7. Conclusion

There is a discussion of UML diagram in assignment. It makes the program more user friendly. It will provide a quick access to song as it has voice assistant and text search availability. As a result of this system, user will able to search songs easily and efficiently.