Render songs

const $ = document.querySelector.bind(document)

const $$ = document.querySelectorAll.bind(document)

const app = {

    songs: [

        {

            name: 'Nevada',

            singer: 'Vicetone',

            path: 'https://aredir.nixcdn.com/NhacCuaTui924/Nevada-Vicetone-4494556.mp3?st=\_IjpS9u0LjapNgzm058wVw&e=1623143773',

            image: 'https://i.pinimg.com/originals/f8/6f/33/f86f3378e656883b33594f06d78d1634.jpg',

        },

        {

            name: 'Muộn rồi mà sao còn',

            singer: 'Sơn Tùng M-TP',

            path: 'https://aredir.nixcdn.com/Believe\_Audio19/MuonRoiMaSaoCon-SonTungMTP-7011803.mp3?st=w9AA-eyRI7yD\_VYGfvVWeQ&e=1623141624',

            image: 'https://pbs.twimg.com/media/Ez5jRyVVgAQN6Kh.jpg',

        },

        {

            name: 'Dusk till dawn',

            singer: 'Zayn',

            path:'https://c1-ex-swe.nixcdn.com/Sony\_Audio37/DuskTillDawn-ZaynSia-5164057.mp3?st=aHn9hp\_NIBradUX6IoFaiQ&e=1626691866&t=1626605467250',

            image:'https://dep.com.vn/wp-content/uploads/2020/03/zayn-malik-hop-tac-martyre-1.jpg'

        },

        {

            name: 'Beautiful Mistake',

            singer: 'Maroon 5',

            path:'https://c1-ex-swe.nixcdn.com/Unv\_Audio195/BeautifulMistakes-Maroon5MeganTheeStallion-6953886.mp3?st=8CBhxbaU-5Ld\_4GzRzzRCA&e=1626693144&t=1626606751441',

            image:'https://we25.vn/media/images/main-workout-build-a-body-like-adam-levine.jpg'

        },

        {

            name: 'La la la',

            singer: 'Naughty boy',

            path:'https://c1-ex-swe.nixcdn.com/Unv\_Audio12/LaLaLa-NaughtyBoy-2555758.mp3?st=uG7chTC8F4fCew8J392nSw&e=1626693144&t=1626606798428',

            image:'https://i.ytimg.com/vi/2WmBa1CviYE/maxresdefault.jpg'

        },

        {

            name: 'Love me like you do',

            singer: 'Ellie',

            path:'https://c1-ex-swe.nixcdn.com/Unv\_Audio73/LoveMeLikeYouDo-EllieGoulding-4233682.mp3?st=qRE707xSAzKlEMkwDjZPrQ&e=1626693394&t=1626606991625',

            image:'https://upload.wikimedia.org/wikipedia/vi/1/17/Ellie\_Goulding\_-\_Love\_Me\_Like\_You\_Do.png'

        }

    ],

    render: function() {

        const htmls = this.songs.map(song => {

            return `

                <div class="song">

                    <div class="thumb"

                        style="background-image:url('${song.image}')">

                    </div>

                    <div class="body">

                        <h3 class="title">${song.name}</h3>

                        <p class="author">${song.singer}</p>

                    </div>

                    <div class="option">

                        <i class="fas fa-ellipsis-h"></i>

                    </div>

                </div>

            `

        })

        $('.playlist').innerHTML =  htmls.join('');

    },

    start: function() {

        this.handleEvents();

        this.render();

    }

}

app.start();

Scroll top

 // handleEvents là hàm lưu mấy cái xử lí các sự kiện

    handleEvents: function() {

        const cd = $('.cd')

        // console.log([cd])

        const cdWidth = cd.offsetWidth

        document.onscroll = function() {

            const scrollTop = document.documentElement.scrollTop || window.scrollY

            const newCdWidth = cdWidth - scrollTop

            // console.log(newCdWidth)

            cd.style.width = newCdWidth > 0 ? newCdWidth + "px" : 0

            cd.style.opacity = newCdWidth / cdWidth;

            //kich thuoc mới chia kích thước cũ => để làm opacity

        }

    },

3. Play/ Pause / Seek

Graphical user interface, text, application

Description automatically generated

DOM: khi tương tắc với phương thức trong element phải get cái element trước

const playBtn = $('.btn-toggle-play')

const player = $('.player')

const progress = $('#progress')

const app = {

    currenIndex: 0,

    isPlaying: false,

    //định nghĩa thuộc tính

    defineProperties: function() {

        // Object.defineProperty() là 1 getter, không cần

        // ngoặc nhọn như hàm

        Object.defineProperty(this, 'currentSong', {

            get: function() {

                return this.songs[this.currenIndex];

            }

        })

    },

    // handleEvents là hàm lưu mấy cái xử lí các sự kiện

    handleEvents: function() {

        // lưu this ở bên ngoài handleEvents vào biến \_this

        const \_this = this

        const cd = $('.cd')

        // console.log([cd])

        // Xử lí phóng to / thu nhỏ CD

        const cdWidth = cd.offsetWidth

        document.onscroll = function() {

            const scrollTop = document.documentElement.scrollTop || window.scrollY

            const newCdWidth = cdWidth - scrollTop

            // console.log(newCdWidth)

            cd.style.width = newCdWidth > 0 ? newCdWidth + "px" : 0

            cd.style.opacity = newCdWidth / cdWidth;

            //kich thuoc mới chia kích thước cũ => để làm opacity

        }

        // Xử lí khi click play

        playBtn.onclick = function() {

            if(\_this.isPlaying) {

                // \_this.isPlaying = false

                audio.pause()

                // player.classList.remove('playing')

            }else {

                audio.play()

            }

        }

        //Khi song được play

        audio.onplay = function() {

            \_this.isPlaying = true

            player.classList.add('playing')

        }

        //Khi song bị pause

        audio.onpause = function() {

            \_this.isPlaying = false

            player.classList.remove('playing')

        }

        // Khi tiến độ bài hát thay đổi

        audio.ontimeupdate = function() {

            //currentTime là thời gian hiện tại

            // chia cho duration(thời gian toàn bài) là ra số phần trăm ở hiện tại

            if(audio.duration) {

                // Math.floor làm tròn dưới

                const progressPercent = Math.floor(audio.currentTime / audio.duration \* 100)

                //progress.value gán bởi progressPercent

                progress.value = progressPercent

                console.log(progress.value)

            }

            // console.log(audio.currentTime / audio.duration \* 100)

        }

        // Xử lý khi tua song

        progress.onchange = function(e) {

            //thơi gian tua

            const seekTime = audio.duration /100 \* e.target.value

            // gán thời gian tua vào audio

            audio.currentTime = seekTime

            // console.log(audio.duration /100 \* e.target.value )

        }

    },

    loadCurrentSong: function() {

        // bây giờ mình cần thay đổi cái header, cái cd-thumb,

        // và truyền src của audio vào

        // trước tiên get element ra

        const heading = $('header h2')

        const cdThumb = $('.cd-thumb')

        const audio = $('#audio')

        // console.log kiểm tra xem có lấy ra đc chưa

        heading.textContent = this.currentSong.name

        cdThumb.style.backgroundImage = `url('${this.currentSong.image})`

        audio.src = this.currentSong.path

        // console.log(heading, cdThumb, audio)

    },

    start: function() {

        //Định nghĩa các thuộc tính cho object

        this.defineProperties();

        // Lắng nghe/ Xử lý các sự kiện (DOM events)

        this.handleEvents();

        //tải thông tin bài hát đầu tiên vào UI(user interface) khi

        // chạy ứng dụng

        this.loadCurrentSong();

        // render playlist

        this.render();

    }

}

app.start();

đây là icon-pause

A picture containing text, person

Description automatically generated

4. CD Rotate

// handleEvents là hàm lưu mấy cái xử lí các sự kiện

    handleEvents: function() {

        // lưu this ở bên ngoài handleEvents vào biến \_this

        const \_this = this

        const cd = $('.cd')

        // console.log([cd])

        // Xử lí CD quay và dừng

        // có 2 đối số truyền vào

        const cdThumbAnimate = cdThumb.animate([

            {transform: 'rotate(360deg)'}

        ], {

            duration: 10000, // 10 seconds

            iterations: Infinity // lặp lại bn lần

        })

        cdThumbAnimate.pause()

        // console.log(cdThumbAnimate)

        // xong phần xoay CD

        //Khi song được play

        audio.onplay = function() {

            \_this.isPlaying = true

            player.classList.add('playing')

            cdThumbAnimate.play()

        }

        //Khi song bị pause

        audio.onpause = function() {

            \_this.isPlaying = false

            player.classList.remove('playing')

            cdThumbAnimate.pause()

        }

        // Khi next song

        nextBtn.onclick = function() {

            \_this.nextSong()

            audio.play()

        }

        // Khi prev song

        prevBtn.onclick = function() {

            \_this.prevSong()

            audio.play()

        }

    },

5. Next song / prevous song

nextSong: function() {

        //tăng lên bài hát kế tiếp

        this.currentIndex++

        // nếu bằng bài cuối thì quay lại bài đầu tiên

        console.log(this.currentIndex, this.songs.length)

        if(this.currentIndex >= this.songs.length) {

            this.currentIndex = 0

        }

        // gọi lại hàm loadCurrentSong, next sẽ tải thông tin mới

        this.loadCurrentSong()

    },

    prevSong: function() {

        //tăng lên bài hát kế tiếp

        this.currentIndex--

        // nếu bằng bài cuối thì quay lại bài đầu tiên

        console.log(this.currentIndex, this.songs.length)

        if(this.currentIndex < 0) {

            this.currentIndex = this.songs.length - 1

        }

        // gọi lại hàm loadCurrentSong, next sẽ tải thông tin mới

        this.loadCurrentSong()

    },

6. Random

playRandomSong: function() {

        let newIndex

        do{

            // this ở đây chính là app

            newIndex = Math.floor(Math.random() \* this.songs.length)

            // nếu bằng bài hiện tại thì lặp tiếp

        }while (newIndex === this.currentIndex)

        // console.log(newIndex)

        app.currentIndex = newIndex

        this.loadCurrentSong()

    },

7. Next / repeat when ended song

Next when ended song

audio.onended = function() {

            // console.log(123)

            nextBtn.click()

        }

// Xử lí lặp lại 1 song

        repeatBtn.onclick = function(e) {

            \_this.isRepeat = !\_this.isRepeat

            repeatBtn.classList.toggle('active', \_this.isRepeat)

        }

        // xử lí next song khi audio end

        audio.onended = function() {

            // console.log(123)

            if(\_this.isRepeat) {

                audio.play()

            }else{

                nextBtn.click()

            }

        }

8. Active song

render: function() {

        const htmls = this.songs.map((song, index) => {

            return `

                <div class="song ${index === this.currentIndex ? 'active' : ''}">

                    <div class="thumb"

                        style="background-image:url('${song.image}')">

                    </div>

                    <div class="body">

                        <h3 class="title">${song.name}</h3>

                        <p class="author">${song.singer}</p>

                    </div>

                    <div class="option">

                        <i class="fas fa-ellipsis-h"></i>

                    </div>

                </div>

            `

        })

        $('.playlist').innerHTML =  htmls.join('');

    },

nextBtn.onclick = function() {

            if(\_this.isRandom) {

                \_this.playRandomSong()

            }else {

                \_this.nextSong()

            }

            audio.play()

            \_this.render()

        }

        // Khi prev song

        prevBtn.onclick = function() {

            if(\_this.isRandom) {

                \_this.playRandomSong()

            }else {

                \_this.prevSong()

            }

            audio.play()

            \_this.render()

        }

Active thì gọi lại\_this.render()

     \*  9. Scroll active song into view

// Khi next song

        nextBtn.onclick = function() {

            if(\_this.isRandom) {

                \_this.playRandomSong()

            }else {

                \_this.nextSong()

            }

            audio.play()

            \_this.render()

            \_this.scrollToActiveSong()

        }

        // Khi prev song

        prevBtn.onclick = function() {

            if(\_this.isRandom) {

                \_this.playRandomSong()

            }else {

                \_this.prevSong()

            }

            audio.play()

            \_this.render()

            \_this.scrollToActiveSong()

        }

scrollToActiveSong: function() {

        setTimeout(() => {

            $('.song.active').scrollIntoView({

                behavior: 'smooth',

                block: 'center',

                inlines: 'end'

            })

        }, 200)

    },

10. play song when click

//Lắng nghe hành vi click vào playlist

        // e là event nhận được ở tham số

        // e.target là đích mà click vào

        // nó sẽ trả về cái mà mình click vào

        playlist.onclick = function(e) {

            // console.log(e.target)

            // closest : khi click vào thz con ko tìm được

            // thì tìm tới thz cha

            const songNode = e.target.closest('.song:not(.active)')

            if(songNode || e.target.closest('.option'))

            {

                   // xử lí khi click vào song

                   if(songNode) {

                        // console.log(songNode.getAttribute('data-index')) cách 1

                        //set index, loadCurrentSong sau đó play

                        \_this.currentIndex = Number(songNode.dataset.index)

                        \_this.loadCurrentSong()

                        \_this.render()

                        audio.play()

                   }

            }

        }

render: function() {

        const htmls = this.songs.map((song, index) => {

            return `

                <div class="song ${index === this.currentIndex ? 'active' : ''}" data-index=${index}>

                    <div class="thumb"

                        style="background-image:url('${song.image}')">

                    </div>

                    <div class="body">

                        <h3 class="title">${song.name}</h3>

                        <p class="author">${song.singer}</p>

                    </div>

                    <div class="option">

                        <i class="fas fa-ellipsis-h"></i>

                    </div>

                </div>

            `

        })

        playlist.innerHTML =  htmls.join('');

    },