



# **MUSIC APPLICATION**

---

BY

NAMAN SHAH


DEEPSANCHANIYA

SAKSHI BASANTANI

# INTRODUCTION



WE ARE BUILDING A HYBRID MUSIC APPLICATION. WHERE USER CAN LISTEN MUSIC ADVERTISEMENT FREE AND CAN ENJOY THEIR MUSIC WITHOUT ANY DISTURBANCE. THIS APPLICATION HAS ALL BASIC FEATURES NEEDED IN A MUSIC APPLICATION



# INDEX

The slide features a minimalist design with a white background. Below the title 'INDEX', there are two horizontal green bars. In the bottom-left corner, there are three overlapping geometric shapes: a teal triangle pointing right, a yellow triangle pointing left, and a green triangle pointing right.

- COMPANY PROFILE
- PROJECT PROFILE
- SCREEN LAYOUTS
- CONCLUSION
- BIBLIOGRAPHY



# **COMPANY PROFILE**

# COMPANY PROFILE



Name : Arham Technosoft Pvt. Ltd

Contact : 98988 63554

Website : [arhamtechnosoft.com](http://arhamtechnosoft.com)

Email : [info@arhamtechnosoft.com](mailto:info@arhamtechnosoft.com)

Address : B-406 Synthesis The First, B/h Keshavbaug Party  
Plot, Vastrapur, Ahmedabad, Gujarat 380015




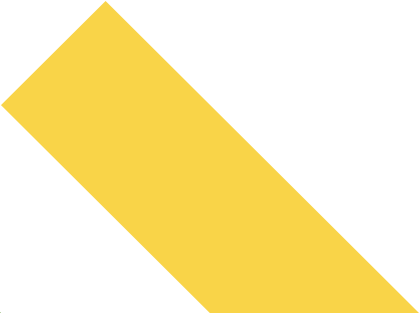

# **PROJECT PROFILE**

# PROJECT PROFILE



## SYSTEM REQUIREMENT

FRONTEND	IONIC FRAMEWORK,ANGULAR, JAVASCRIPT, HTML, CSS
BACK END	FIREBASE



# PROJECT PROFILE

The slide features two horizontal green bars, one on the left and one on the right, positioned below the title. In the bottom-left corner, there are three overlapping geometric shapes: a teal triangle pointing right, a yellow parallelogram pointing down-right, and a green triangle pointing right.

## EXISTING SYSTEM

Currently we face problems while listening to music because of advertisements after every song and also advertisement banners on the screen. The experience of enjoying music is disturbed because of the advertisement and to avoid it we need to pay a subscription.



# PROJECT PROFILE

The slide features two horizontal green bars positioned below the title and above the section header. In the bottom-left corner, there are three overlapping geometric shapes: a teal triangle pointing right, a yellow parallelogram pointing down-right, and a green triangle pointing down-right.

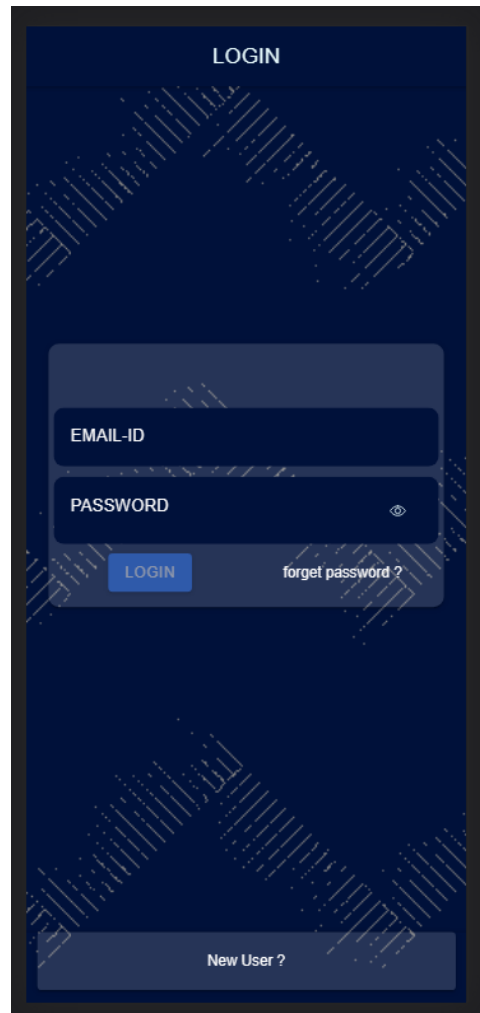
## **PROPOSED SYSTEM**

This project will be a Music Streaming Application. Using Ionic Framework for building Hybrid mobile applications . Firebase for making Application Programming Interface (API) and data storage.



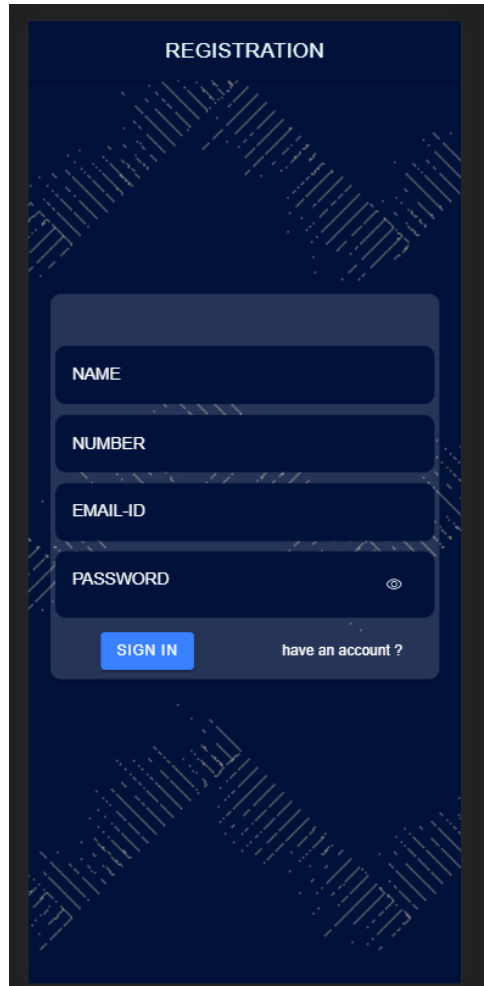
# **SCREEN LAYOUTS**

# LOGIN AND REGISTRATION SCREEN



The login page will be shown to the user when they install the application or they clear the app data from their phones.

# LOGIN AND REGISTRATION SCREEN



REGISTRATION

NAME

NUMBER

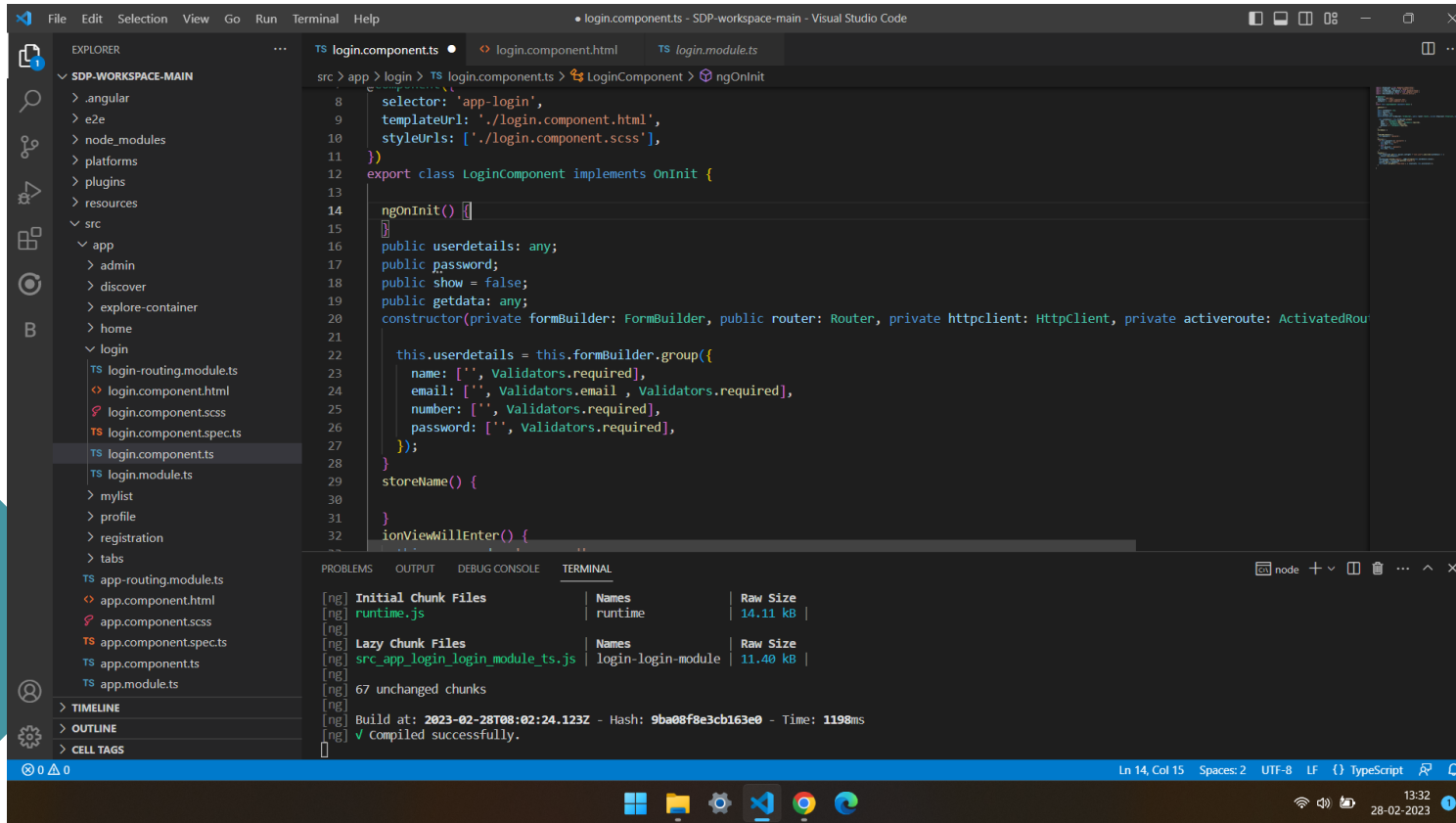
EMAIL-ID

PASSWORD

[SIGN IN](#) [have an account ?](#)

The Registration page will only be shown when user clicks on the "new User ?" button on Login page .

# LOGIN AND REGISTRATION SCREEN



The screenshot shows the Visual Studio Code editor with the file explorer on the left displaying the project structure. The main editor window shows the `login.component.ts` file. The code defines a `LoginComponent` that implements `OnInit`. It includes a `selector`, `templateUrl`, and `styleUrls`. The `ngOnInit` method initializes the `userdetails` form with validation rules for name, email, number, and password. The `storeName` method is also defined. The terminal at the bottom shows the build output, indicating that the application was compiled successfully.

```
src > app > login > TS login.component.ts > LoginComponent > ngOnInit
8 selector: 'app-login',
9 templateUrl: './login.component.html',
10 styleUrls: ['./login.component.scss'],
11 })
12 export class LoginComponent implements OnInit {
13
14   ngOnInit() {
15   }
16   public userdetails: any;
17   public password;
18   public show = false;
19   public getdata: any;
20   constructor(private FormBuilder: FormBuilder, public router: Router, private httpClient: HttpClient, private activatedRoute: ActivatedRoute) {}
21
22   this.userdetails = this.FormBuilder.group({
23     name: ['', Validators.required],
24     email: ['', Validators.email, Validators.required],
25     number: ['', Validators.required],
26     password: ['', Validators.required],
27   });
28   storeName() {
29   }
30   ionViewWillEnter() {
31   }
32 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Initial Chunk Files	Names	Raw Size
[ng] runtime.js	runtime	14.11 kB

Lazy Chunk Files	Names	Raw Size
[ng] src_app_login_login_module_ts.js	login-login-module	11.40 kB

67 unchanged chunks

Build at: 2023-02-28T08:02:24.123Z - Hash: 9ba08f8e3cb163e0 - Time: 1198ms

✓ Compiled successfully.

This code shows that the submit button of form will be disabled Until the form is filled with valid information .

```
<ion-col size="6" class="ion-text-center">
  <ion-button (click)="onSubmit()" [disabled]="userdetails.valid" block color="primary" type="submit">
    <ion-text class="ion-text-center">
      Login
    </ion-text>
  </ion-button>
</ion-col>
```

# HOME SCREEN



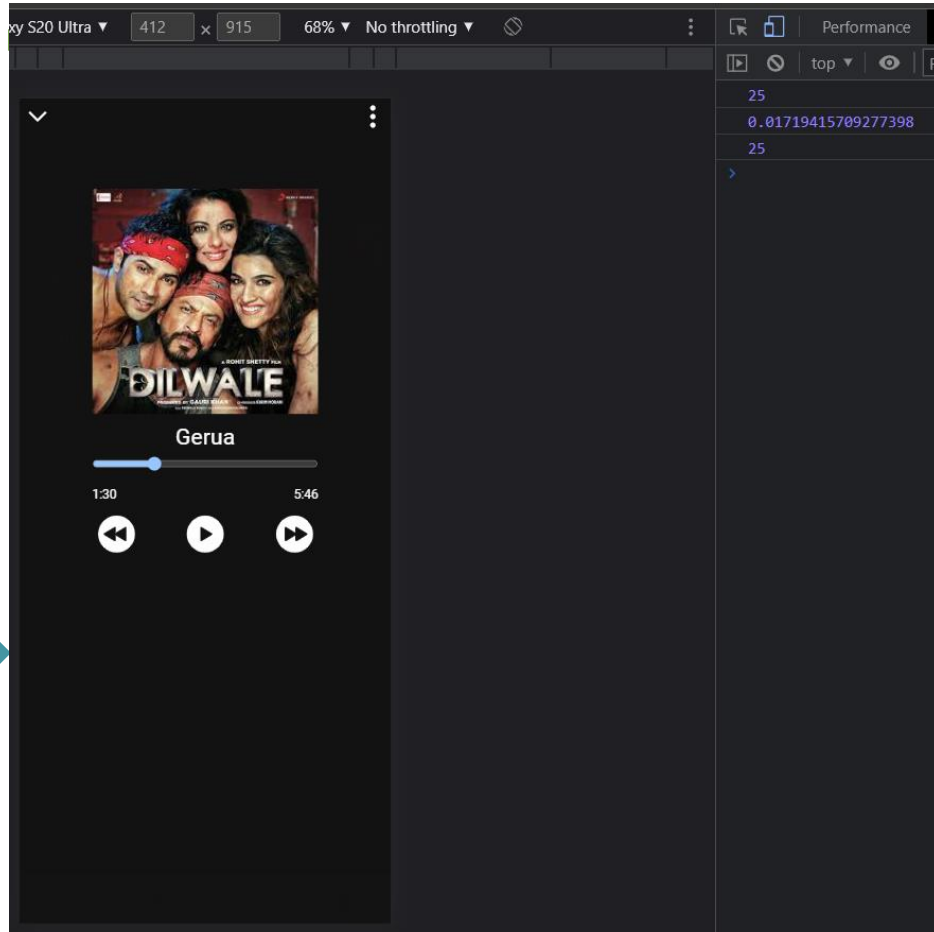
These are list of songs available.  
The audio controller is made  
using howler.js library

The image shows a Visual Studio Code editor window with the following components:

- Explorer (Left):** Displays the project structure. The 'src/app/login' directory is expanded, showing files like 'login-routing.module.ts', 'login.component.html', 'login.component.scss', 'login.component.spec.ts', and 'login.component.ts'.
- Main Editor:** Displays the 'login.component.ts' file. The code defines a 'LoginComponent' class that implements 'OnInit'. It includes a 'selector' of 'app-login', a 'templateUrl' of './login.component.html', and 'styleUrls' of ['./login.component.scss']. The class has properties for 'userdetails', 'password', 'show', and 'getdata', and a constructor that takes 'formBuilder', 'router', 'httpClient', and 'activatedRoute' as arguments. It also includes a 'storeName()' method and an 'ionViewWillEnter()' method.
- Terminal (Bottom):** Shows the output of a build command. It lists 'Initial Chunk Files' (runtime.js, 14.11 kB) and 'Lazy Chunk Files' (src\_app\_login\_login\_module\_ts.js, 11.40 kB). It also shows '67 unchanged chunks' and a successful build message: 'Build at: 2023-02-28T08:02:24.123Z - Hash: 9ba08f8e3cb163e0 - Time: 1198ms'.

## Logic of how the list of songs is called

# HOME SCREEN



Logic of how the list of songs is called

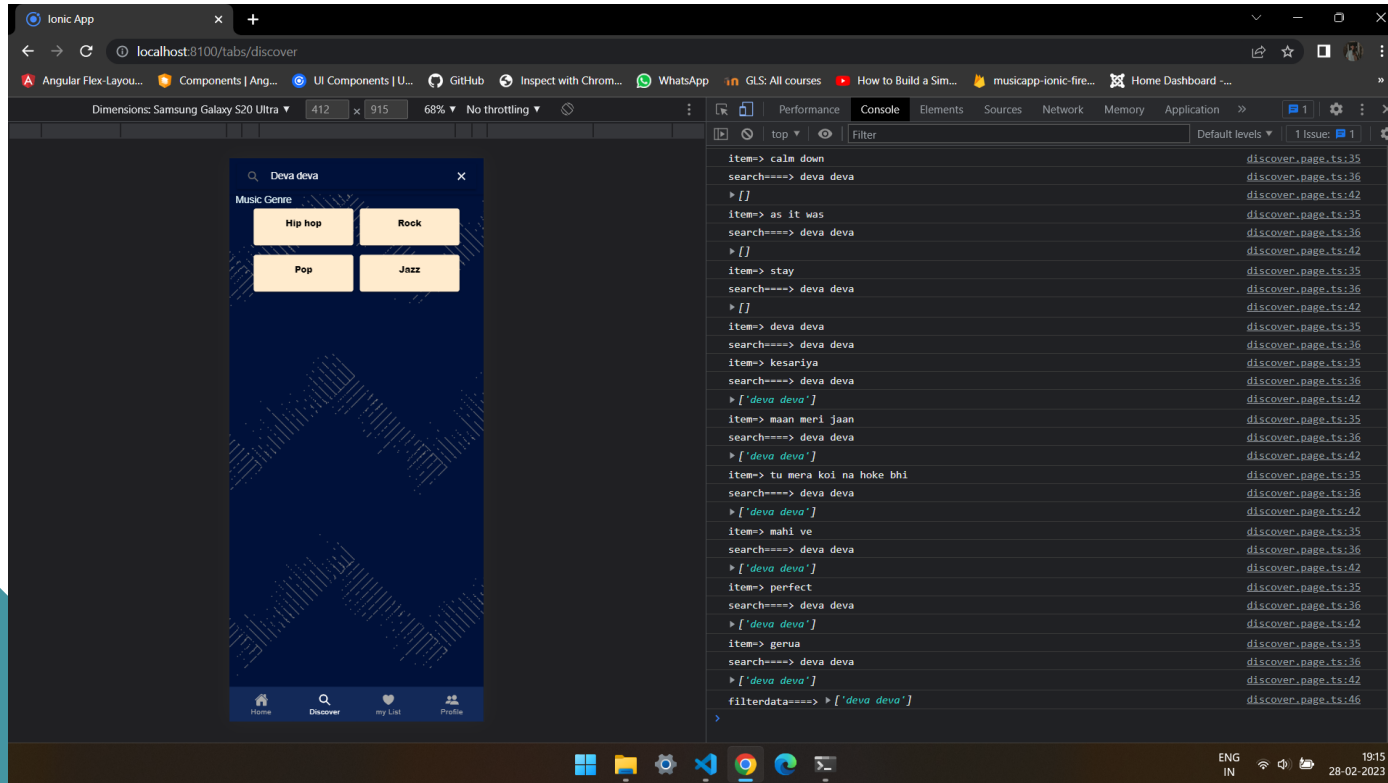


# HOME SCREEN

```
91 playAudio(div: any) {
92   this.index = div;
93   if (this.audio) {
94     this.audio.pause();
95     this.audio.currentTime = 0;
96   }
97   this.audio = new Audio();
98
99   this.audio.src = this.musicdetails[div].link;
100   this.audio.load();
101   this.image = this.musicdetails[this.index].image;
102   this.title = this.musicdetails[this.index].text
103 }
104
105 pauseAudio() {
106   this.audio.pause();
107   this.works = true;
108 }
109
110 playButton() {
111   this.works = false;
112   this.audio.play();
113   // console.log('load', this.audio.load);
114   // console.log('time', this.time);
115   // console.log('audio', this.audio);
116   console.log('currentTime', this.audio.currentTime);
117   console.log('length', this.audio.duration);
118   this.convert();
119 }
120
121 next(index: any) {
122   // console.log('Index:', index)
123   if (index == this.musicdetails.length - 1) {
124     this.index = 0;
125     this.works = false;
126     this.audio.src = this.musicdetails[this.index].link;
127     this.audio.load();
128     this.audio.play();
129     this.title = this.musicdetails[this.index].text;
130     this.image = this.musicdetails[this.index].image;
131   } else {
132     this.index = this.index + 1;
133     this.audio.src = this.musicdetails[this.index].link;
134     this.audio.load();
135     this.audio.play();
136     this.works = false;
137     this.image = this.musicdetails[this.index].image;
138     this.title = this.musicdetails[this.index].text;
139   } // console.log('works:', this.index)
140 }
141
142 prev(index: any) {
143   // console.log('Index:', index)
144   if (index == 0) {
145     this.works = false;
146     this.index = this.musicdetails.length - 1;
147     this.audio.src = this.musicdetails[this.index].link;
148     this.title = this.musicdetails[this.index].text;
149     this.image = this.musicdetails[this.index].image;
150     this.audio.load();
151     this.audio.play();
152   } else {
153     this.index = this.index - 1;
154     this.audio.src = this.musicdetails[this.index].link;
155     this.title = this.musicdetails[this.index].text;
156     this.works = false;
157     this.audio.load();
158     this.audio.play();
159     this.image = this.musicdetails[this.index].image;
160   } // console.log('works:', this.index)
161 }
162
163 dismiss() {
164   this.modalController.dismiss({
165     'dismissed': true
166   });
167 }
168
169 convert() {
170   if (this.audio.currentTime == 'NaN') {
171     console.log('works works');
172   }
173   // console.log(this.audio.duration)
174   let that = this;
175   this.intervall = window.setInterval(function () {
176     // console.log(that.time)
177     // console.log(that.audio.currentTime / that.audio.duration ? that.audio.duration : 1.0);
178     if (that.audio.duration) that.time = that.audio.currentTime / that.audio.duration;
179     // console.log(that.audio.currentTime, that.audio.duration);
180     that.demos = that.time * 100;
181   }, 1000);
182 }
183
184 drag(event) {
185   console.log(this.demos);
186   console.log(this.time);
187   let change = this.demos;
188   console.log(change);
189   if (change != 0) {
190     this.audio.currentTime = (change * this.audio.duration) / 100;
191   }
192   // this.time=event
193 }
```

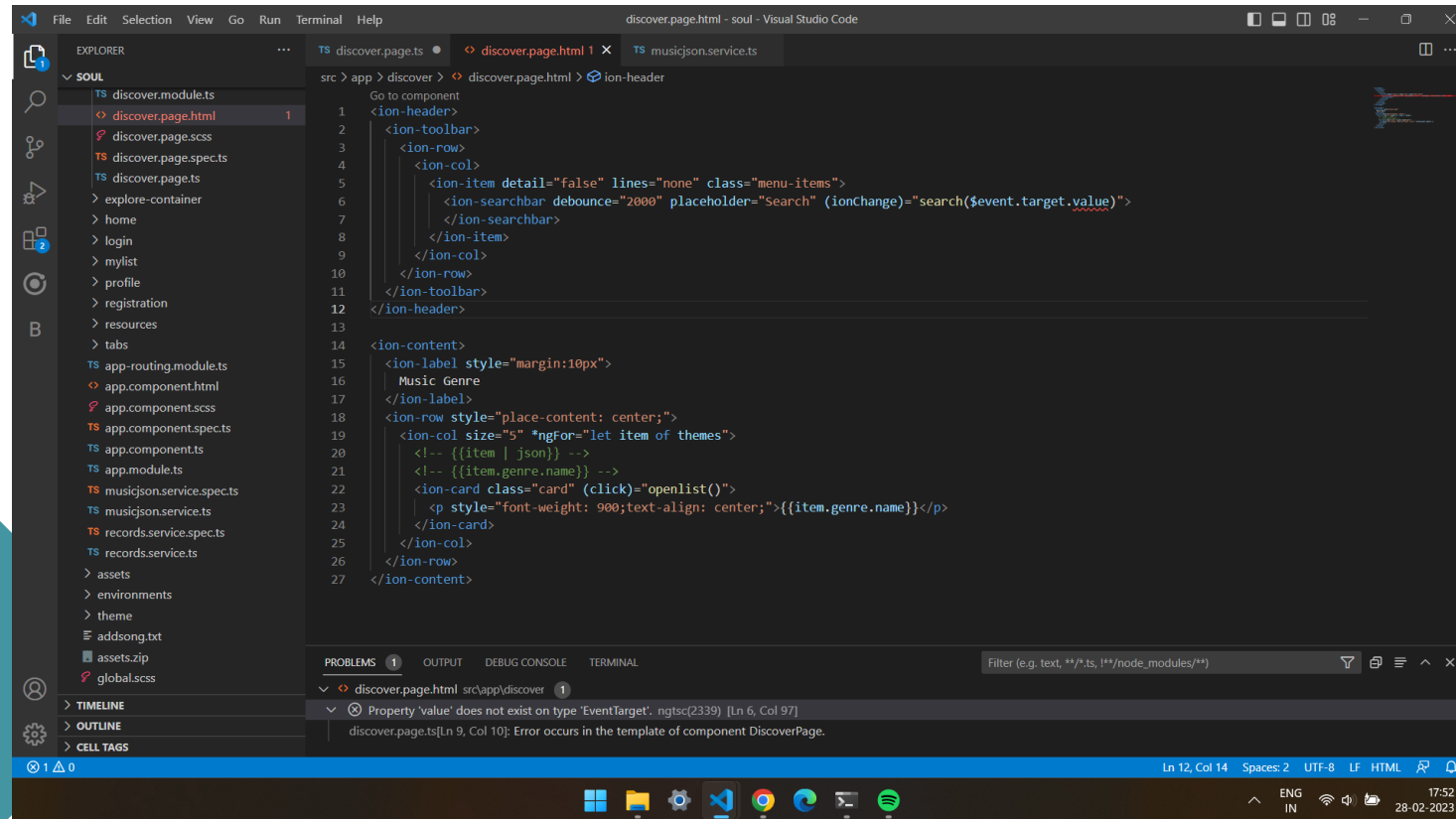
Logic for Music controller

# DISCOVER SCREEN



Search bar

# DISCOVER SCREEN



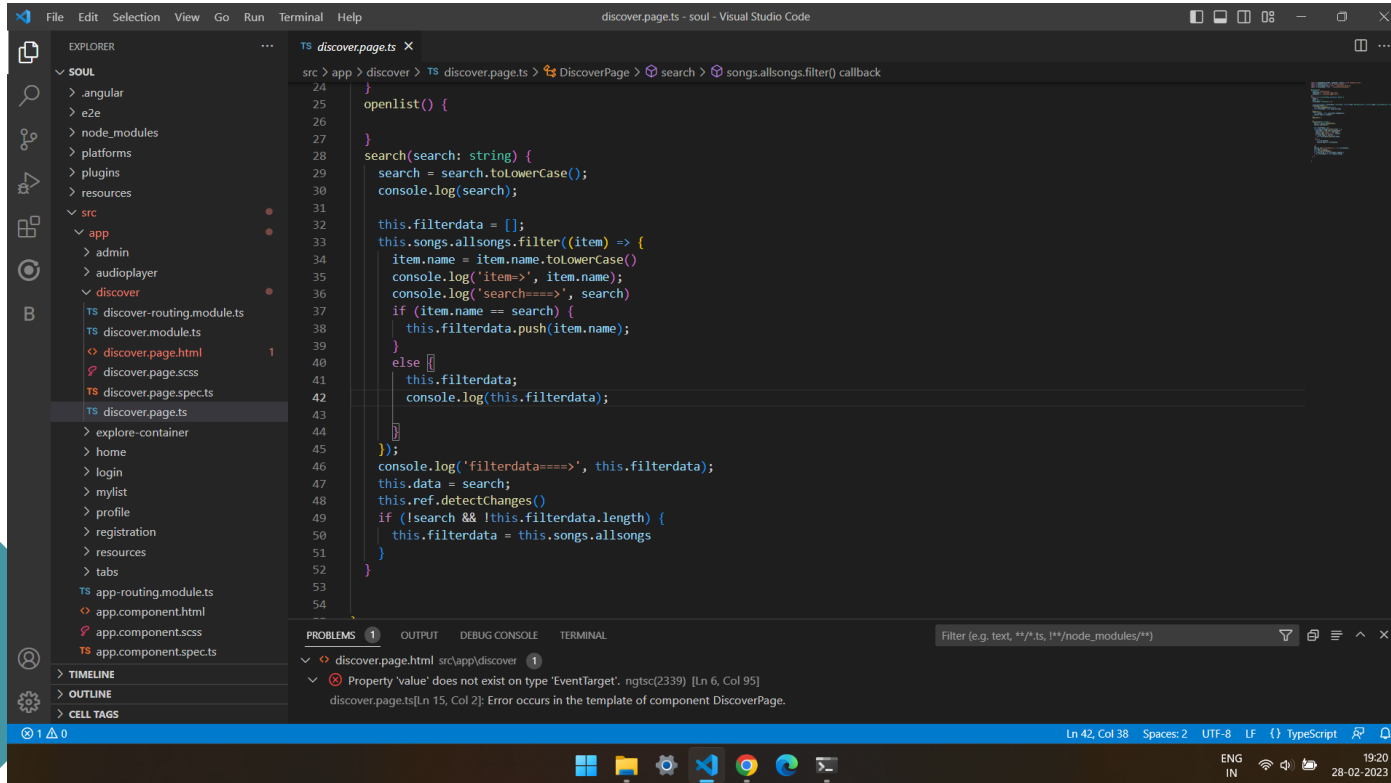
The screenshot shows the Visual Studio Code editor with the following components:

- EXPLORER:** A file tree on the left showing the project structure. The file `discover.page.html` is selected.
- EDITOR:** The main workspace showing the HTML template for the Discover page. The code includes an `<ion-header>` with a search bar and an `<ion-content>` with a list of music genres.
- PROBLEMS:** A panel at the bottom showing a TypeScript error: `Property 'value' does not exist on type 'EventTarget'. ngts(2339) [Ln 6, Col 97]`.

```
1 <ion-header>
2   <ion-toolbar>
3     <ion-row>
4       <ion-col>
5         <ion-item detail="false" lines="none" class="menu-items">
6           <ion-searchbar debounce="2000" placeholder="Search" (ionChange)="search($event.target.value)">
7             </ion-searchbar>
8         </ion-item>
9       </ion-col>
10    </ion-row>
11  </ion-toolbar>
12 </ion-header>
13
14 <ion-content>
15   <ion-label style="margin:10px">
16     Music Genre
17   </ion-label>
18   <ion-row style="place-content: center;">
19     <ion-col size="5" *ngFor="let item of themes">
20       <!-- {{item | json}} -->
21       <!-- {{item.genre.name}} -->
22       <ion-card class="card" (click)="openlist()">
23         <p style="font-weight: 900;text-align: center;">{{item.genre.name}}</p>
24       </ion-card>
25     </ion-col>
26   </ion-row>
27 </ion-content>
```

Search bar

# DISCOVER SCREEN



```
src > app > discover > TS discover.page.ts > DiscoverPage > search > songs.allsongs.filter() callback
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

openlist() {
}

search(search: string) {
  search = search.toLowerCase();
  console.log(search);

  this.filterdata = [];
  this.songs.allsongs.filter((item) => {
    item.name = item.name.toLowerCase();
    console.log('item->', item.name);
    console.log('search====>', search);
    if (item.name == search) {
      this.filterdata.push(item.name);
    }
    else {
      this.filterdata;
      console.log(this.filterdata);
    }
  });
  console.log('filterdata====>', this.filterdata);
  this.data = search;
  this.ref.detectChanges()
  if (!search && !this.filterdata.length) {
    this.filterdata = this.songs.allsongs
  }
}
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

discover.page.html src/app/discover 1

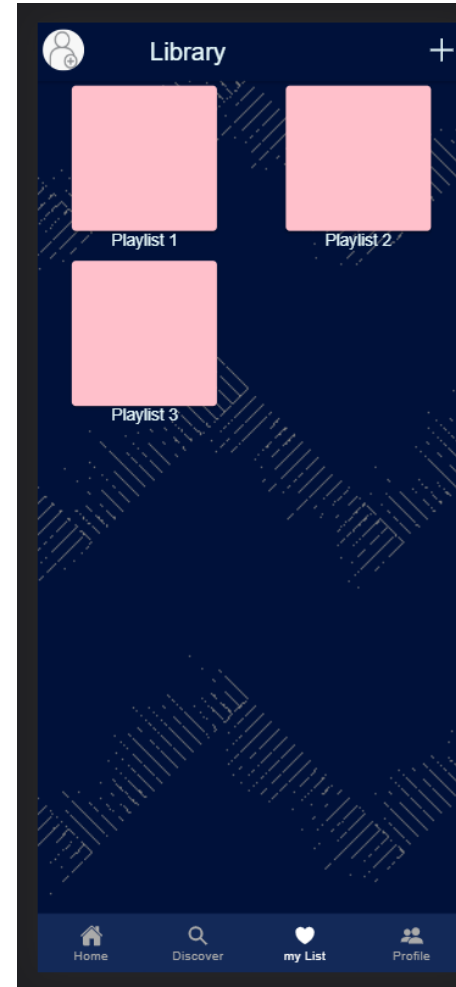
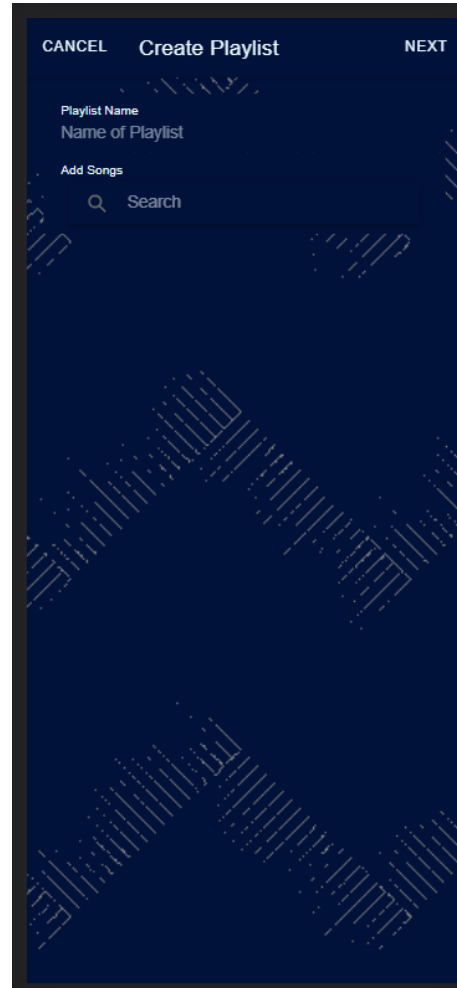
Property 'value' does not exist on type 'EventTarget'. ngts(2339) [Ln 6, Col 95]

discover.page.ts[Ln 15, Col 2]: Error occurs in the template of component DiscoverPage.

Ln 42, Col 38 Spaces: 2 UTF-8 LF TypeScript

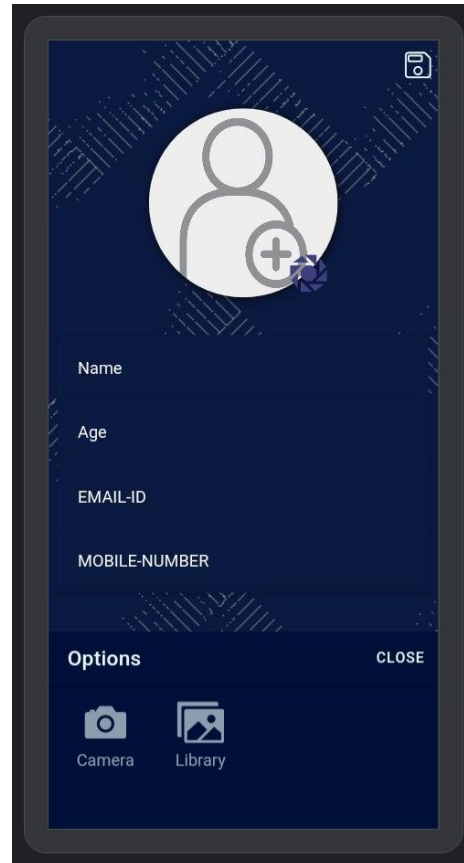
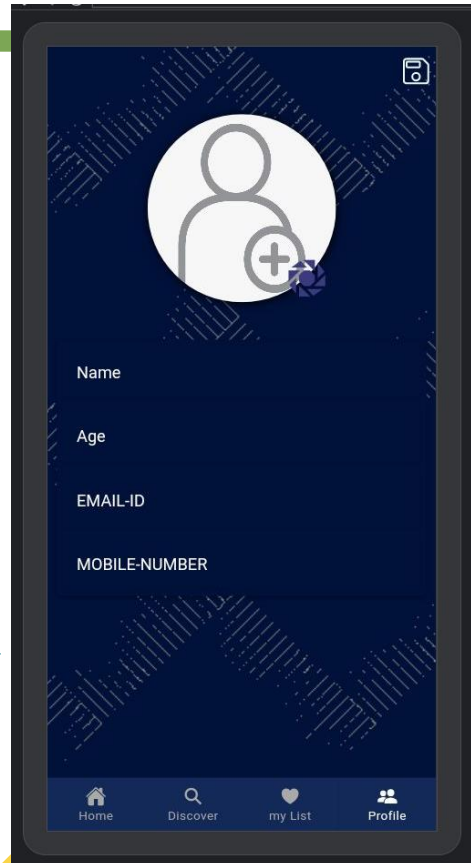
Search bar logic

# MY-LIST SCREEN



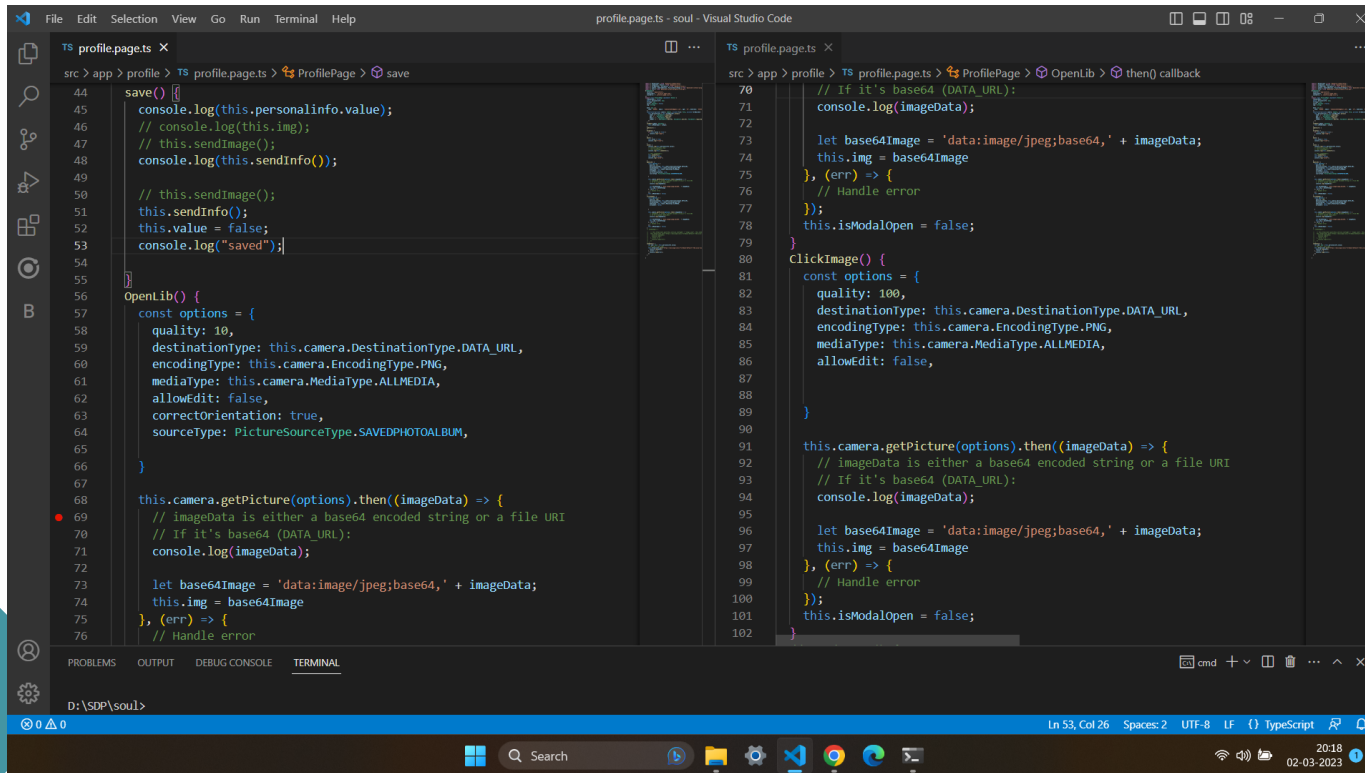
This page is used by user to create their playlist

# PROFILE SCREEN



Add profile picture to your profile

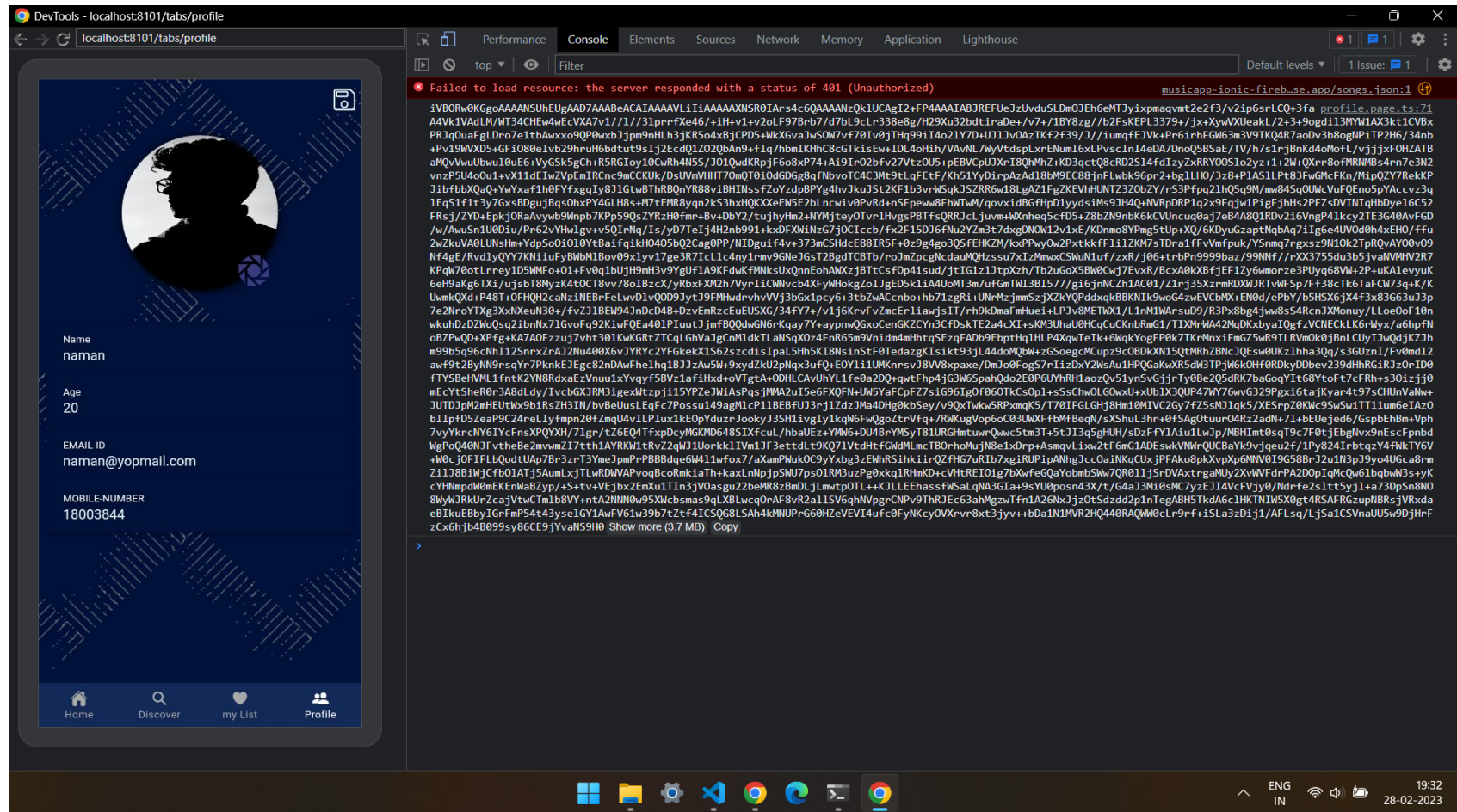
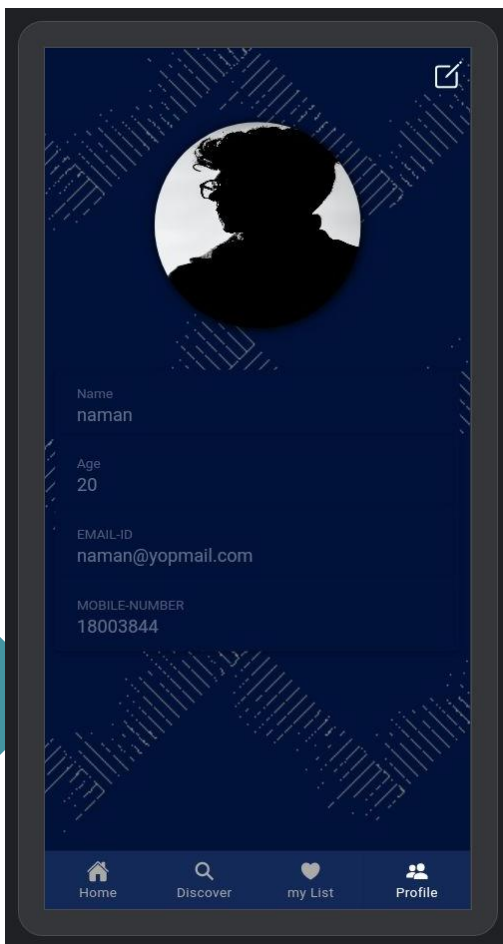
# PROFILE SCREEN



```
src > app > profile > TS profile.page.ts > ProfilePage > save
44 save() {
45   console.log(this.personalinfo.value);
46   // console.log(this.img);
47   // this.sendImage();
48   console.log(this.sendInfo());
49
50   // this.sendImage();
51   this.sendInfo();
52   this.value = false;
53   console.log("saved");
54 }
55
56 OpenLib() {
57   const options = {
58     quality: 10,
59     destinationType: this.camera.DestinationType.DATA_URL,
60     encodingType: this.camera.EncodingType.PNG,
61     mediaType: this.camera.MediaType.ALLMEDIA,
62     allowEdit: false,
63     correctOrientation: true,
64     sourceType: PictureSourceType.SAVEDPHOTOALBUM,
65   }
66
67   this.camera.getPicture(options).then((imageData) => {
68     // imageData is either a base64 encoded string or a file URI
69     // If it's base64 (DATA_URL):
70     console.log(imageData);
71
72     let base64Image = 'data:image/jpeg;base64,' + imageData;
73     this.img = base64Image
74   }, (err) => {
75     // Handle error
76   })
77 }
78
79
80 ClickImage() {
81   const options = {
82     quality: 100,
83     destinationType: this.camera.DestinationType.DATA_URL,
84     encodingType: this.camera.EncodingType.PNG,
85     mediaType: this.camera.MediaType.ALLMEDIA,
86     allowEdit: false,
87   }
88
89   this.camera.getPicture(options).then((imageData) => {
90     // imageData is either a base64 encoded string or a file URI
91     // If it's base64 (DATA_URL):
92     console.log(imageData);
93
94     let base64Image = 'data:image/jpeg;base64,' + imageData;
95     this.img = base64Image
96   }, (err) => {
97     // Handle error
98   })
99   this.isModalOpen = false;
100 }
101
102
```

Add profile picture to your profile

# PROFILE SCREEN





# CONCLUSION

---

Our main aim is to provide user experience with a proper user-interface to interact with our music Application . Thus users can enjoy their own choice of music anytime and without any advertisement. Our internal guide Garima mam helped us a lot in improving the quality of our project. As well as our external guide Mr. Aakash Vora helped us a lot in giving suggestions for better functioning of our project.

# Bibliography

---

- <https://ionicframework.com/docs/components>
- <https://ionicframework.com/docs/cli/commands/cordova-plugin>
- <https://material.angular.io/components/categories>
- <https://tburleson-layouts-demos.firebaseio.com/#/docs>
- <https://firebase.google.com/docs?authuser=1&hl=en>
- <https://www.npmjs.com/package/howler>



**Thank you**

---