

React Native Intern Assignment: Music Player

Build a Music Player

Create a music streaming app using the JioSaavn API. The evaluation will focus on architecture, state management, and UI polish.

Design Reference:

<https://www.figma.com/design/jm3TbqEdkR15QNVDE4rSIX/Lokal---Music-Player-App-UI-Kit-Sample>

API

Base URL:

<https://saavn.sumit.co/>

No API key required.

Documentation: <https://saavn.sumit.co/docs>

Endpoints

Search API

GET /api/search
GET /api/search/songs
GET /api/search/albums
GET /api/search/artists
GET /api/search/playlists

Sample Endpoint: `curl -X GET "https://saavn.sumit.co/api/search/songs?query=arijit"`

Sample Response

```
{  
  "status": "SUCCESS",  
  "data": {
```

```
"results": [
{
  "id": "4NRpZd1v",
  "name": "Hawayein",
  "type": "",
  "album": {
    "id": "11287835",
    "name": "Jab Harry Met Sejal",
    "url": "https://www.jiosaavn.com/album/jab-harry-met-sejal/fcxD-6fyEgM_"
  },
  "year": "2017",
  "releaseDate": null,
  "duration": "289",
  "label": "Sony Music Entertainment India Pvt. Ltd.",
  "primaryArtists": "Pritam, Arijit Singh",
  "primaryArtistsId": "456323, 459320",
  "featuredArtists": "",
  "featuredArtistsId": "",
  "explicitContent": 0,
  "playCount": "180221404",
  "language": "hindi",
  "hasLyrics": "false",
  "url": "https://www.jiosaavn.com/song/hawayein/RCY5QS5UBkU",
  "copyright": "(P) 2017 Sony Music Entertainment India Pvt. Ltd.",
  "image": [
    {
      "quality": "50x50",
      "link": "https://c.saavncdn.com/584/Jab-Harry-Met-Sejal-Hindi-2017-20170803161007-50x50.jpg"
    },
    {
      "quality": "150x150",
      "link": "https://c.saavncdn.com/584/Jab-Harry-Met-Sejal-Hindi-2017-20170803161007-150x150.jpg"
    },
    {
      "quality": "500x500",
      "link": "https://c.saavncdn.com/584/Jab-Harry-Met-Sejal-Hindi-2017-20170803161007-500x500.jpg"
    }
  ],
  "downloadUrl": [
    {
      "quality": "12kbps",
      "link": "https://aac.saavncdn.com/584/1c4f10826f2336b0cb7db275f1051f8c_12.mp4"
    },
    {
      "quality": "48kbps",
      "link": "https://aac.saavncdn.com/584/1c4f10826f2336b0cb7db275f1051f8c_48.mp4"
    },
    {
      "quality": "96kbps",
      "link": "https://aac.saavncdn.com/584/1c4f10826f2336b0cb7db275f1051f8c_96.mp4"
    },
    {
      "quality": "160kbps",
      "link": "https://aac.saavncdn.com/584/1c4f10826f2336b0cb7db275f1051f8c_160.mp4"
    },
    {
      "quality": "320kbps",
      "link": "https://aac.saavncdn.com/584/1c4f10826f2336b0cb7db275f1051f8c_320.mp4"
    }
  ]
}
```

```
],
  "total": 24551,
  "start": 1
}
}
```

Songs API

GET /api/songs
GET /api/songs/{id}
GET /api/songs/{id}/suggestions

Sample Endpoint: curl -X GET "<https://saavn.sumit.co/api/songs/aRZbUYD7>"

Sample Response

```
{
  "success": true,
  "data": [
    {
      "id": "aRZbUYD7",
      "name": "Tum Hi Ho",
      "duration": 262,
      "language": "hindi",
      "album": {
        "id": "1139549",
        "name": "Aashiqui 2"
      },
      "artists": {
        "primary": [
          {
            "id": "459320",
            "name": "Arijit Singh"
          }
        ]
      },
      "image": [
        {
          "quality": "150x150",
          "url": "https://c.saavncdn.com/430/Aashiqui-2-Hindi-2013-150x150.jpg"
        },
        {
          "quality": "500x500",
          "url": "https://c.saavncdn.com/430/Aashiqui-2-Hindi-2013-500x500.jpg"
        }
      ],
      "downloadUrl": [
        {
          "quality": "96kbps",
          "url": "https://aac.saavncdn.com/430/5c5ea5cc00e3bff45616013226f376fe_96.mp4"
        },
        {
          "quality": "128kbps",
          "url": "https://aac.saavncdn.com/430/5c5ea5cc00e3bff45616013226f376fe_128.mp4"
        }
      ]
    }
  ]
}
```

```

    {
      "quality": "320kbps",
      "url": "https://aac.saavncdn.com/430/5c5ea5cc00e3bff45616013226f376fe_320.mp4"
    }
  ]
}
]
}

```

Artists API

GET /api/artists
 GET /api/artists/{id}
 GET /api/artists/{id}/songs
 GET /api/artists/{id}/albums

Sample Endpoint: curl -X GET "<https://saavn.sumit.co/api/artists/459320>"

Requirements

Build the following screens with proper state synchronization between them:

Screen	Core Functionality
Home	Song list, search, pagination
Player	Full controls, seek bar, background playback
Mini Player	Persistent bar synced with full player
Queue	Add, reorder, remove songs, persisted locally

Key Challenges:

- Background playback must continue when the app is minimized or the screen is off
- Mini Player and Full Player must remain perfectly in sync across navigation

Bonus Requirements:

- Shuffle and repeat modes
- Ability to download songs for offline listening

Feel free to add any extra features, and mention them

Tech Stack

- React Native(Expo) with TypeScript
 - React native Navigation v6+ (do not use Expo Router)
 - State management: Zustand or Redux Toolkit
 - Storage: MMKV or AsyncStorage
-

Submit

1. GitHub repository
2. APK
3. README (setup, architecture, trade-offs)
4. Demo video (2–3 minutes)

Do not use mock data. Do not over-engineer.