MUSIC APP USING RAD MODEL

INTRODUCTION:

The Rapid Application Development (RAD) model is an adaptive software development approach that emphasizes quick development and user feedback through iterative prototyping, rather than extensive pre-planning.

For the development of the LET'S VIBE!!! music streaming app, the RAD model is highly suitable because it allows fast creation and frequent updates based on user experience testing and real-time feedback.

Phase 1: Requirements Planning

OBJECTIVES:

- Create a smooth and visually appealing mobile app that allows users to browse, search, and stream their favorite songs easily.
- Use the RAD model to quickly build working prototypes and gather feedback to refine user interface (UI) designs and functionality.

Key Features Identified:

User-friendly Home Screen:

- Displays personalized sections like "Liked Songs" and "Trending This Week."

- Quick access to top songs, artists, and new releases.

Advanced Search Bar:

- Allows users to instantly search for songs, albums, and artists.

Liked Songs Collection:

- Users can mark their favorite songs and view them easily from the Home Screen.

Use Cases:

Use case1: Search for a Song --- User searches for a song by typing into the search bar.

Use case 2: Play a Song from Liked Songs --- User taps on a song from the Liked Songs section to start playing it

Use case 3: Browse Trending Songs --- User scrolls through the "Trending This Week" section and explores new songs.

Phase 2: User Design

1. Install and Launch Axure RP:

- Download and install **Axure RP** from the official Axure website.
- Launch the application and set up your workspace.

2. Create a New Project:

- Go to **File** → **New** to create a new project.
- Name the project: **"LET'S VIBE!!! Music App UI"**.

3. Create Wireframes:

Use the **widget library** to drag and drop elements onto the canvas and create wireframes for each screen:

- **Home Page**
  - Search Bar, Liked Songs section, Trending Songs section, Mini Player.
- **Liked Songs Page**
  - Grid/List view of favorite songs.
- **Trending Songs Page**
  - Explore songs trending this week.
- **Now Playing Screen**
  - Album art, playback controls, progress bar, lyrics section.
- **Lyrics Screen**
  - Display full lyrics with auto-scroll feature.
- **Mini Player Component**
  - Persistent player at the bottom of every screen.
- **Search Results Page**
  - Display search results dynamically.
- **Settings/Profile Page**
  - User settings, preferences, and app info.

4. Add Interactions:

- Select interactive elements like **Play Button**, **Next**, **Previous**, **Heart Icon**, etc.
- In **Properties → Interactions**, define actions like:

    - **OnClick → Open Now Playing screen**
    - **OnClick → Like/Unlike song**
    - **OnClick → Navigate to Lyrics section**

5. Create Masters:

- Create reusable components (Masters):
    - **Header** (App logo, notification, profile icon).
    - **Footer/Mini Player** (song title, album art, play controls).
    - **Trending Song Card** (image + title + artist name).

6. Add Annotations:

- Use **Notes Panel** to add details for every element:
    - Purpose of each button.
    - Navigation paths.
    - Expected behavior on interaction (play, pause, next, shuffle, etc.).

Phase 3: Construction

1. Develop Interactive Prototypes:

- Convert your static wireframes into fully interactive prototypes.
- Add:
    - Page transitions (smooth slide for screen navigation).
    - Dynamic Panels (for pop-ups like lyrics display or options menu).

    - Carousel/Dynamic list panels (for Liked Songs and Trending Songs).

2. Test and Iterate:

- Use **Preview** to view and test all interactions.
- Conduct **user testing sessions** with real users/stakeholders.
- Collect feedback on usability, visual flow, and navigation.
- Iterate based on feedback — improve layouts, interactions, or screen transitions.

Phase 4: Cutover

1. Finalize and Export:

- Polish and finalize your prototype:
    - Ensure all navigations are working.
    - All songs/cards/mini-player elements interact smoothly.
- Export as:
    - **HTML files** for local testing.
    - Or **share via Axure Cloud** for easy stakeholder access.

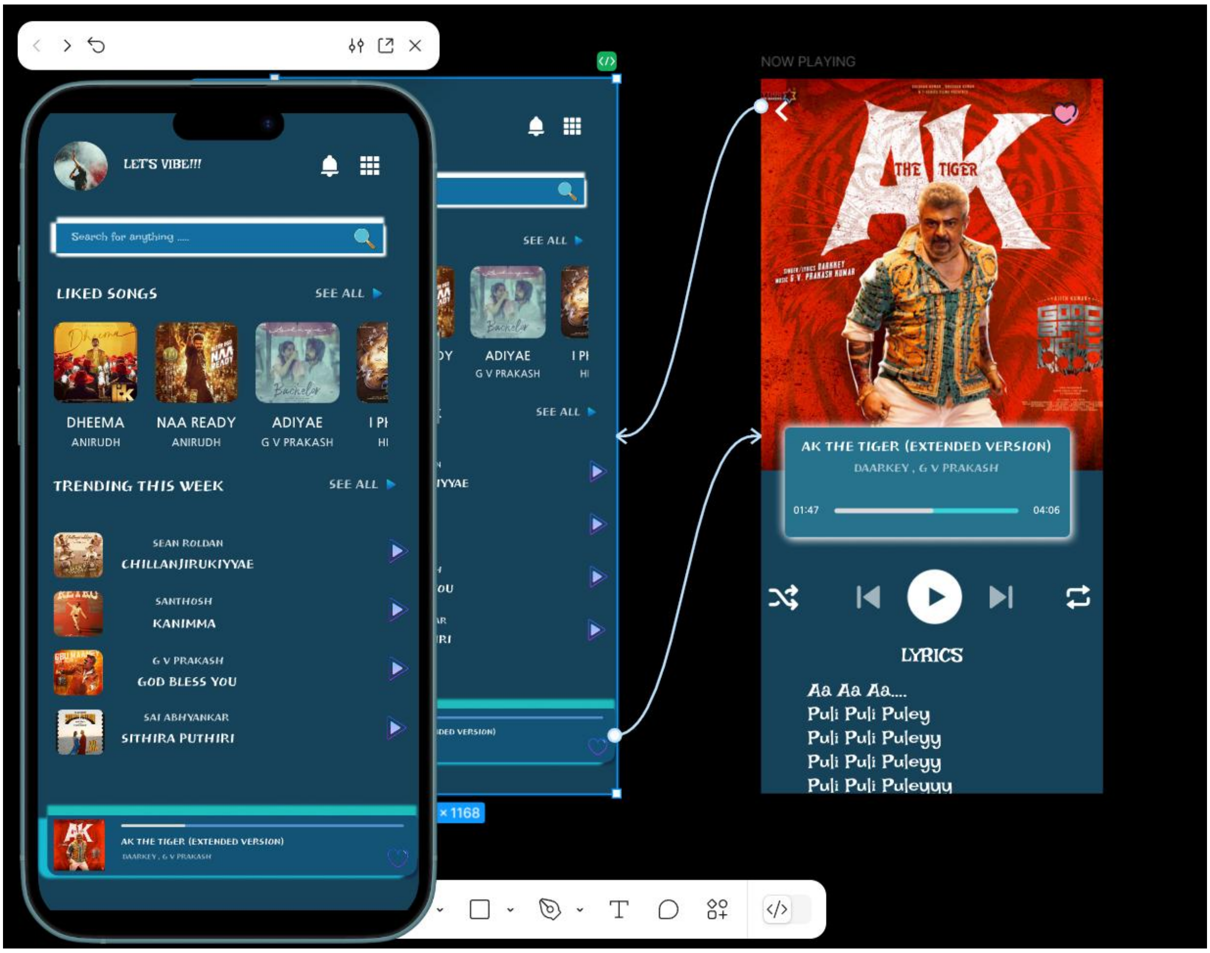
2. User Training and Support:

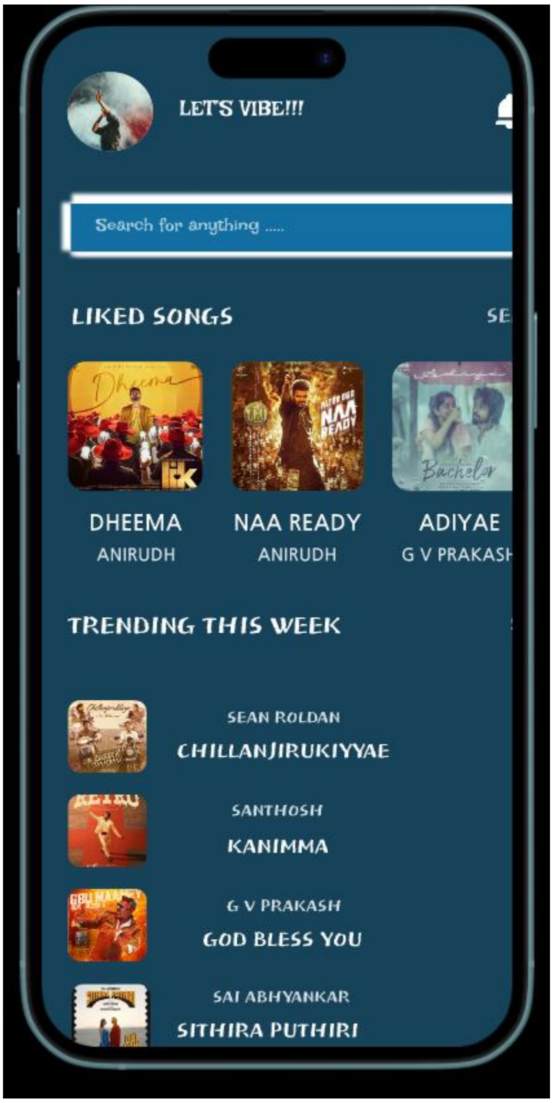
- Conduct **walkthrough sessions** demonstrating how the music app flows from Search → Play → Lyrics → Like Songs.
- Prepare **simple documentation** for key features:
    - How to use search, mini-player, like songs, view lyrics, etc.
- Offer **basic troubleshooting support** for navigating the app.

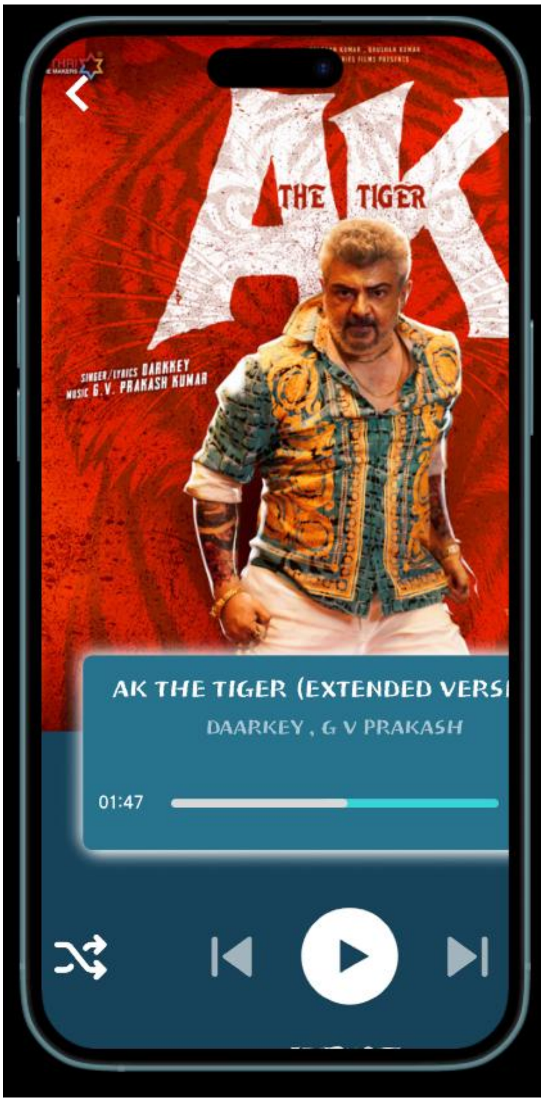
CONCLUSION:

During the **User Design phase**, detailed wireframes were created using Axure RP to map out the app's core screens — including Home, Now Playing, Lyrics, and Trending sections — with a focus on intuitive navigation, aesthetic appeal, and seamless user interactions. Interactive elements such as playback controls, like buttons, and dynamic mini-players were thoughtfully designed to provide a smooth and engaging user experience.

MUSIC APP DESIGNS AND PROTOTYPES:




LET'S VIBE!!!
Search for anything .....
LIKED SONGS
SE
DHEEMA
ANIRUDH
NAA READY
ANIRUDH
ADIYAE
G V PRAKASH
TRENDING THIS WEEK
SEAN ROLDAN
CHILLANJIRUKIYYAE
SANTHOSH
KANIMMA
G V PRAKASH
GOD BLESS YOU
SAI ABHYANKAR
SITHIRA PUTHIRI


AK THE TIGER
SINGER/LYRICS DARKKEY
MUSIC G.V. PRAKASH KUMAR
AK THE TIGER (EXTENDED VERSI
DAARKEY , G V PRAKASH
01:47