A Redux Inspite Podcast App With Dynamic Themes For Android

A Project Report

Submitted by

Team Leader	Reg Number
MANICKA PREAM.P	20201071401111

Team Member	Reg Number
Arunkumar.U	20201071401102
SUNDARRAJ.P	20201071401125

BACHELOR OF COMPUTER APPLICATION

AGE QUOD AGIS

Department of Computer Applications

The M.D.T Hindu College

Un-Aided Courses

Tirunelveli – 10

CONTENTS

INTRODUCTION

1.1 Overview A brief description about your project

In this project, we are developing a podcast app for Android that is built on the Redux architecture. The app aims to provide users with a seamless and intuitive podcast listening experience, while also allowing them to customize the app's appearance with dynamic themes.

Redux is a popular architectural pattern that is used to manage application state in JavaScript-based applications. It is particularly well-suited to mobile app development, as it helps to simplify the management of complex state changes and makes it easier to maintain app consistency.

The app we are developing will be designed to meet the needs of podcast listeners who are looking for a reliable and easy-to-use podcast app that also offers customization options.

Features:

The podcast app we are developing will include a range of features to help users discover, subscribe to, and listen to their favorite podcasts. Some of the key features of the app include:

Podcast discovery: The app will include a search function that allows users to search for podcasts by keyword, genre, or specific podcast name. Users will also be able to browse popular podcasts and view recommendations based on their listening history.

Podcast subscriptions: Once users have discovered a podcast they want to listen to, they will be able to subscribe to it within the app. This will allow them to receive automatic updates when new episodes are released.

Podcast playback: The app will allow users to listen to podcasts within the app, with features such as playback speed adjustment, the ability to skip forward or backward, and the ability to download episodes for offline listening.

Dynamic themes: One of the key features of the app will be the ability to customize the app's appearance with dynamic themes. These themes will be designed to change based on the time of day or user preferences, providing a personalized experience for each user.

User profiles: The app will allow users to create profiles, which will store their listening history and preferences. This will allow users to easily resume listening to a podcast from where they left off, and receive personalized recommendations based on their listening habits.

Social sharing: The app will allow users to share podcasts with friends and followers on social media platforms such as Facebook and Twitter.

1.2 Purpose

The purpose of the project is to develop a podcast app for Android that is built on the Redux architecture and provides users with a seamless and intuitive podcast listening experience. The app aims to meet the needs of podcast listeners who are looking for a reliable and easy-to-use podcast app that also offers customization options.

The app will include features such as podcast discovery, subscriptions, playback, dynamic themes, user profiles, and social sharing. These features will be designed to provide users with a personalized experience and allow them to easily discover, subscribe to, and listen to their favorite podcasts.

The app is built on the Redux architecture, which helps to manage the state of the application in a more predictable and scalable way. This architecture is particularly well-suited to mobile app development, as it simplifies the management of complex state changes and ensures app consistency.

Overall, the purpose of the project is to develop a high-quality podcast app for Android that provides users with a seamless and personalized listening experience.

2 Problem Definition & Design Thinking

Problem Definition:

The problem we are trying to solve with this project is that many podcast listeners struggle to find a reliable and user-friendly app for listening to their favorite podcasts. While there are many podcast apps available, they often lack customization options and can be difficult to navigate.

Additionally, some podcast apps may not be designed with a user-friendly interface, making it challenging for users to find and subscribe to the podcasts they want to listen to. As a result, podcast listeners may spend a lot of time searching for podcasts and navigating through different apps, which can be frustrating and time-consuming.

Design Thinking:

To solve this problem, we will use a design thinking approach to develop a podcast app that meets the needs of podcast listeners. Design thinking is a problem-solving approach that emphasizes understanding user needs and developing solutions that meet those needs in a creative and effective way.

We will begin by conducting research to gain a deeper understanding of the needs and preferences of podcast listeners. This research will include surveys, interviews, and user testing to gather insights on what features users are looking for in a podcast app and how they currently use existing apps.

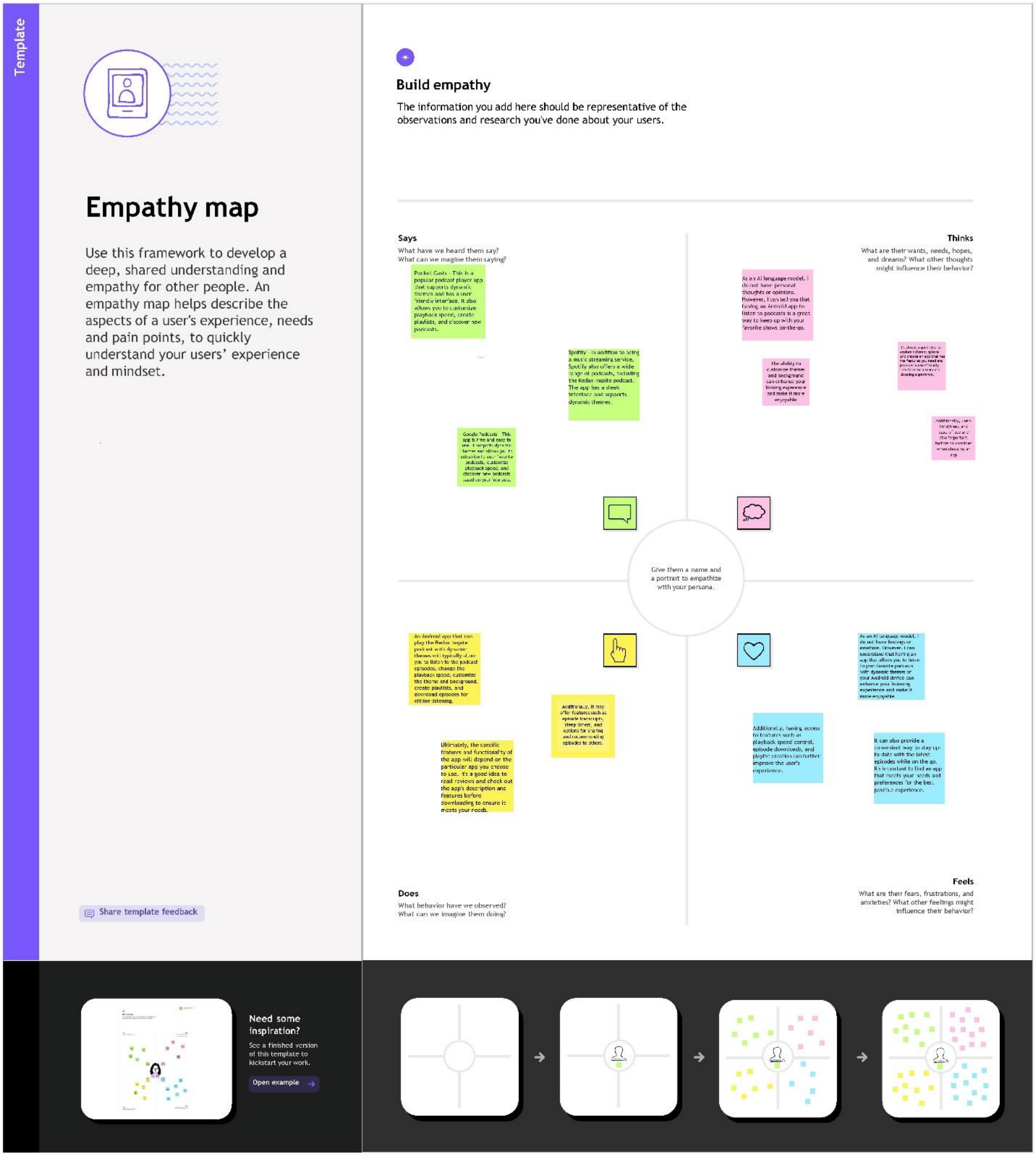
Using the insights gathered from our research, we will develop a set of design principles and use them to guide our development process. These principles will include usability, simplicity, customization, and personalization.

We will then create a series of wireframes and prototypes to test the app's functionality and user experience. These prototypes will be tested with actual users, and their feedback will be used to iterate and refine the design.

Once the design is finalized, we will use the Redux architecture to develop the app, which will allow us to manage the app's state in a more predictable and scalable way. We will also incorporate dynamic themes to allow users to customize the app's appearance based on their preferences and time of day.

Overall, our design thinking approach will ensure that the app we develop is user-centric and meets the needs and preferences of podcast listeners. By focusing on usability, simplicity, customization, and personalization, we believe we can create a podcast app that provides users with a seamless and personalized listening experience.

2.1 Empathy Map Paste the empathy map screenshot



Empathy map

Use this framework to develop a deep, shared understanding and empathy for other people. An empathy map helps describe the aspects of a user's experience, needs and pain points, to quickly understand your users' experience and mindset.

Build empathy

The information you add here should be representative of the observations and research you've done about your users.

Says

What have we heard them say?
What can we magine them saying?

Pocket Casts - This is a popular podcast player app that supports dynamic themes and has a user-friendly interface. It also allows you to customize playback speed, create playlists, and discover new podcasts.

Spotify - In addition to being a music streaming service, Spotify also offers a wide range of podcasts, including the Redux inspite podcast. The app has a sleek interface and supports dynamic themes.

Google Podcasts - This app is free and easy to use. It supports dynamic themes and allows you to subscribe to your favorite podcasts, customize playback speed, and discover new podcasts based on your interests.

Thinks

What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

As an AI language model, I do not have personal thoughts or opinions. However, I can tell you that having an Android app to listen to podcasts is a great way to keep up with your favorite shows on-the-go.

The ability to customize themes and background can enhance your listning experience and make it more enjoyable

It's always a good idea to explore different options and choose an app that has the features you need and provides a user-friendly interface for a seamless listening experience.

Additionally, user-friendliness and ease of use are also important factors to consider when choosing an app.

Give them a name and a portrait to empathize with your persona.

An Android app that can play the Redux inspite podcast with dynamic themes will typically allow you to listen to the podcast episodes, change the playback speed, customize the theme and background, create playlists, and download episodes for offline listening.

Additionally, it may offer features such as episode transcripts, sleep timers, and options for sharing and recommending episodes to others.

Ultimately, the specific features and functionality of the app will depend on the particular app you choose to use. It's a good idea to read reviews and check out the app's description and features before downloading to ensure it meets your needs.

Does

What behavior have we observed?
What can we imagine them doing?

As an AI language model, I do not have feelings or emotions. However, I can understand that having an app that allows you to listen to your favorite podcasts with dynamic themes on your Android device can enhance your listening experience and make it more enjoyable.

Additionally, having access to features such as playback speed control, episode downloads, and playlist creation can further improve the user's experience.

It can also provide a convenient way to stay up-to-date with the latest episodes while on the go. It's important to find an app that meets your needs and preferences for the best possible experience.

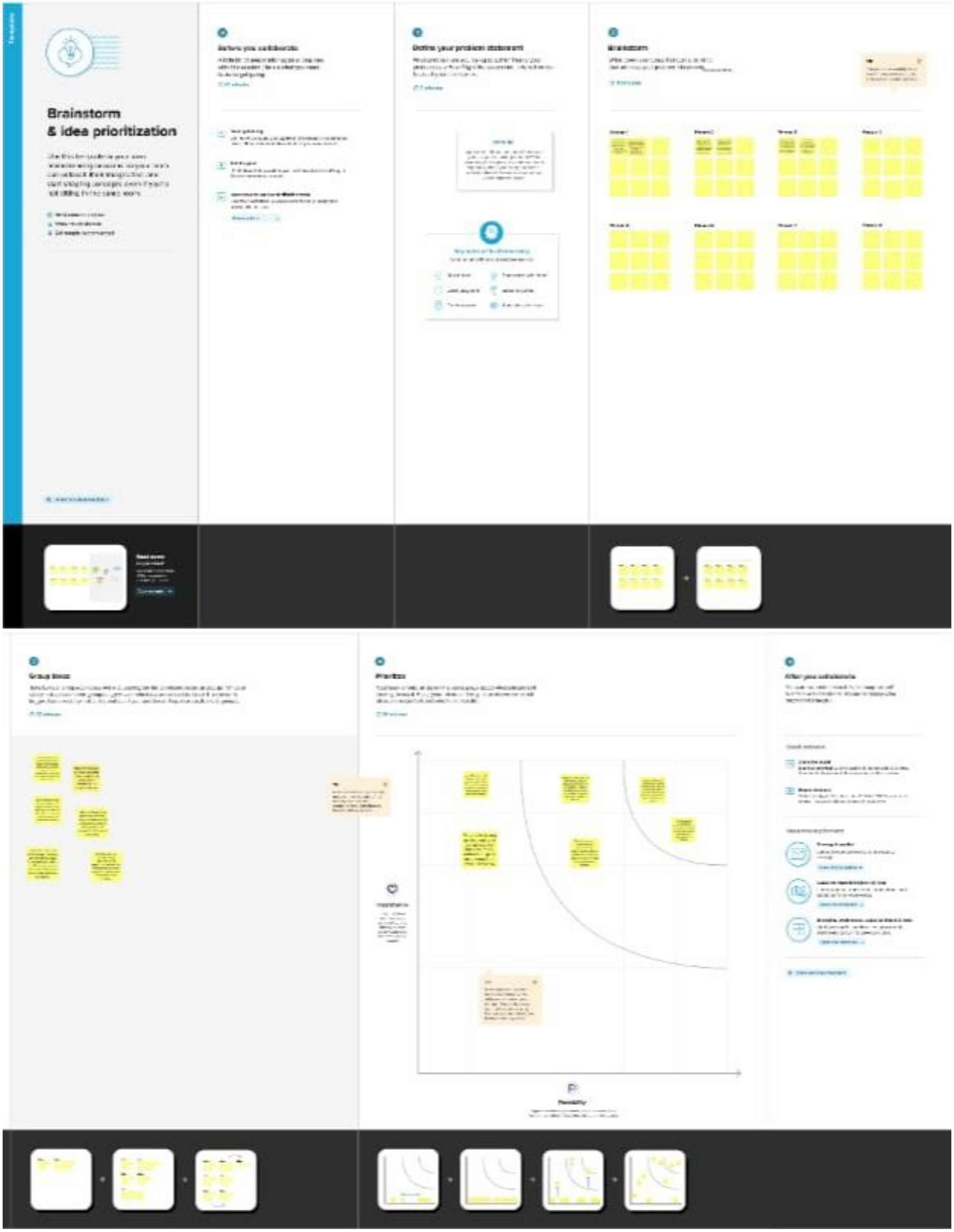
Feels

What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

Need some inspiration?

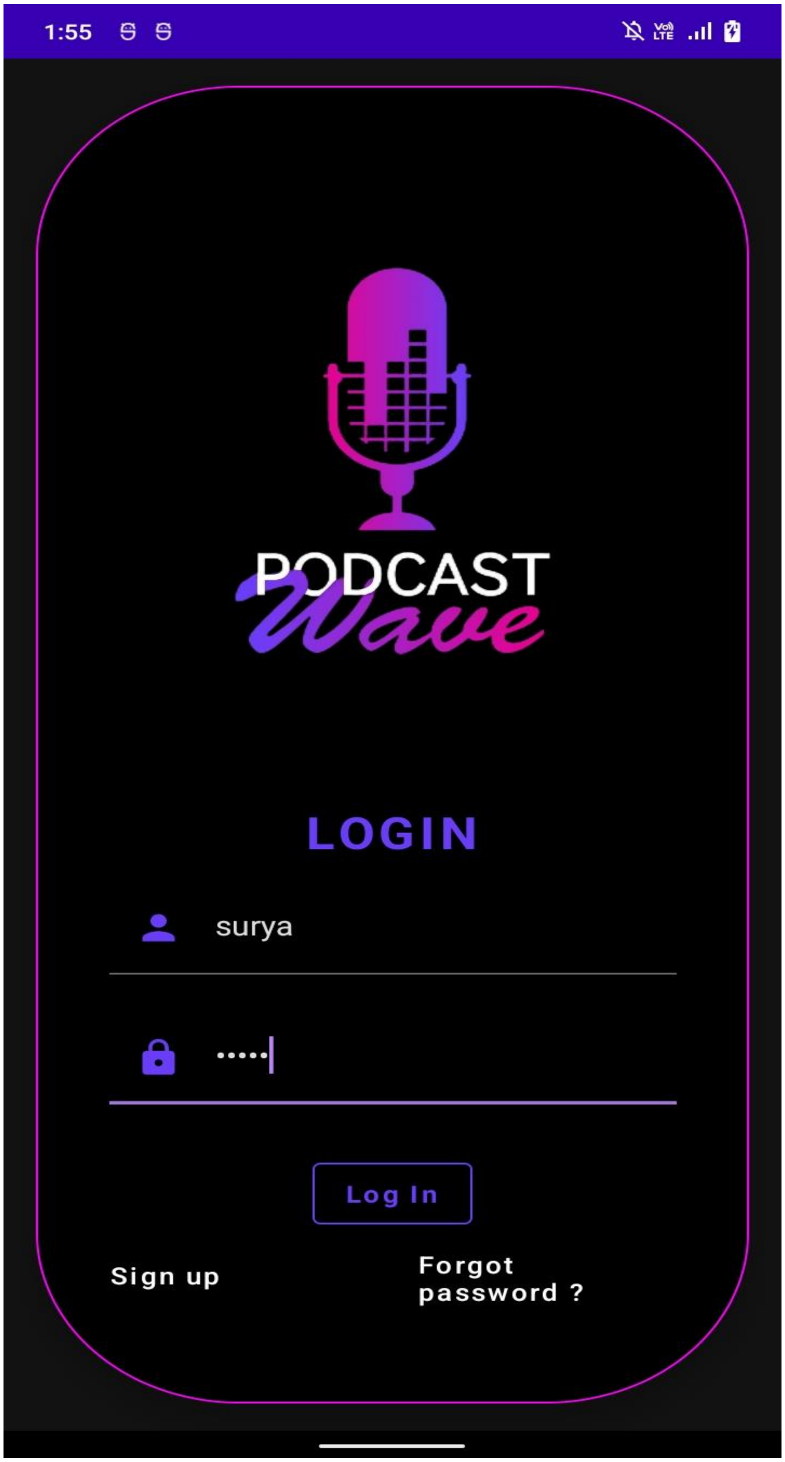
See a finished version of this template to kickstart your work.

2.2 Ideation & Brainstorming Map Paste the Ideation & brainstorming map screenshot



3 RESULT Final findings (Output) of the project along with screenshots.

Login page:

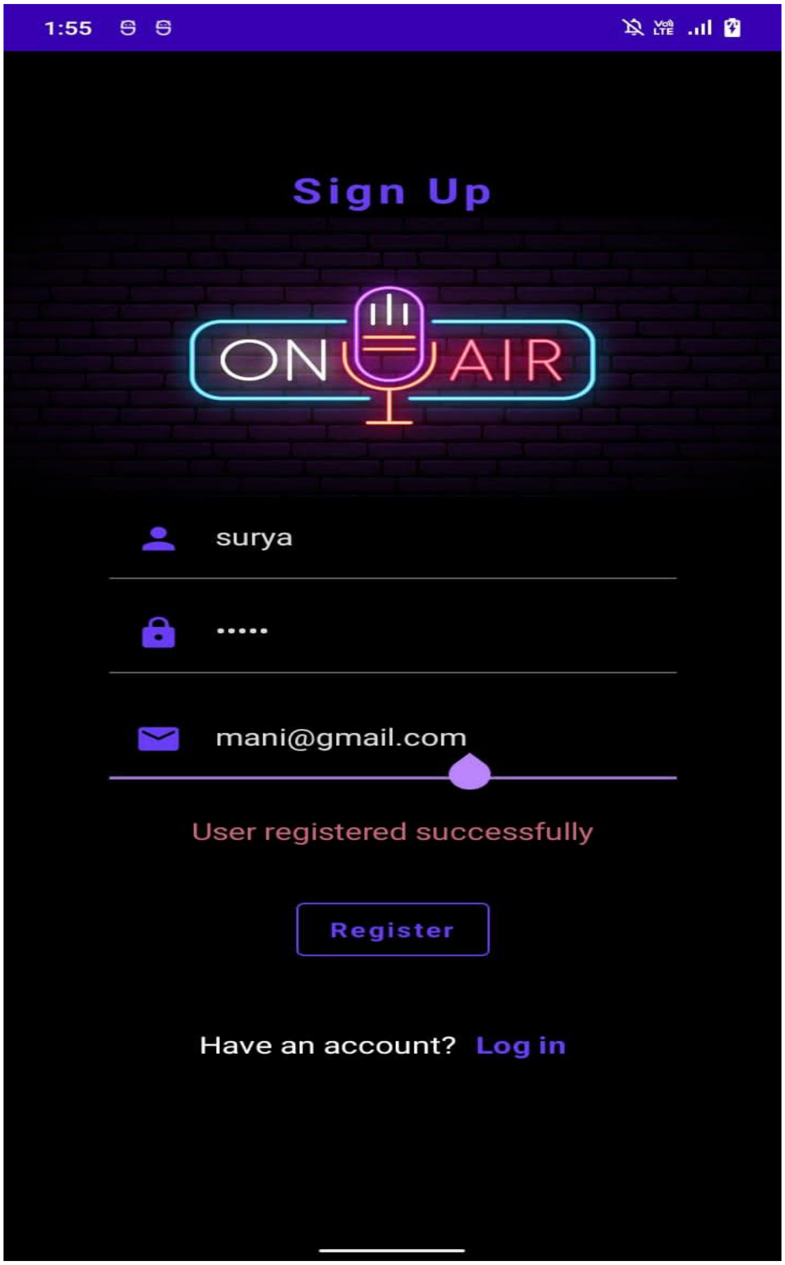


LOGIN

surya

•••••

Signup page:



1:55

Sign Up

surya

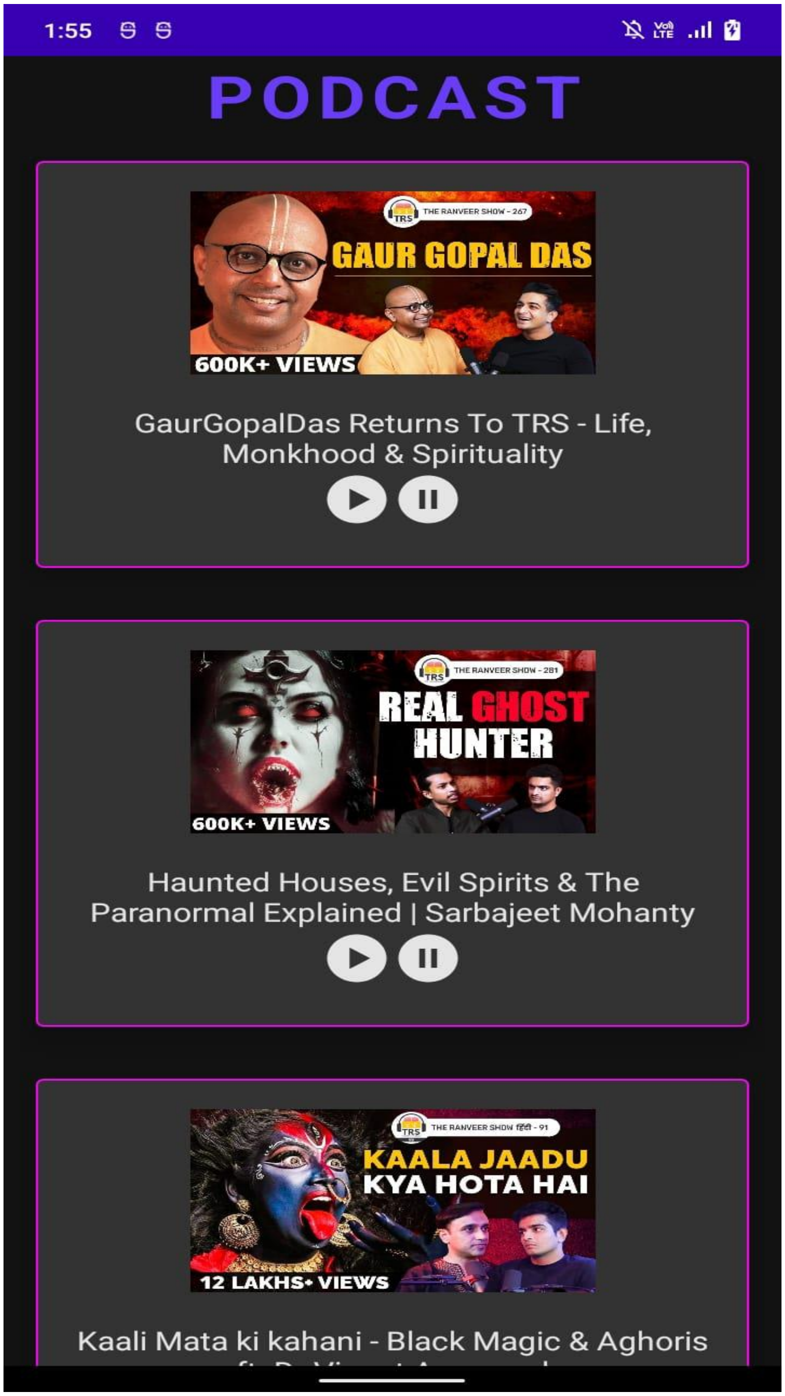
•••••

mani@gmail.com

User registered successfully

Have an account? Log in

Main page :



1:55

PODCAST

GaurGopalDas Returns To TRS - Life, Monkhood & Spirituality

▶ ⏸

Haunted Houses, Evil Spirits & The Paranormal Explained | Sarbajeet Mohanty

▶ ⏸

Kaali Mata ki kahani - Black Magic & Aghoris

4 ADVANTAGES & DISADVANTAGES

Advantages:

- Personalized listening experience: By incorporating customization and personalization features, the app can provide users with a personalized listening experience that meets their specific needs and preferences.
- User-friendly interface: The design thinking approach and user testing will ensure that the app is easy to navigate and understand, making it accessible to a wide range of users.
- Redux architecture: Using Redux architecture will enable the app to manage complex state changes in a more predictable and scalable way, resulting in a more stable and reliable app.
- Dynamic themes: Dynamic themes will allow users to customize the app's appearance based on their preferences and time of day, making the app more visually appealing and engaging.
- Social sharing: By incorporating social sharing features, the app can enable users to share their favorite podcasts with their friends and family, increasing engagement and user acquisition.

Disadvantages:

- Development time: Developing a high-quality podcast app with advanced features like dynamic themes and social sharing can take a significant amount of time and resources.
- Platform-specific: The app will only be available on Android devices, limiting its potential user base.
- Competition: The podcast app market is highly competitive, and there are many established players in the market. It may be challenging to gain traction and attract a significant user base.
- Cost: Developing a high-quality podcast app can be expensive, and monetizing the app through ads or subscription fees may not be feasible if the user base is small.
- User adoption: Even with a well-designed and user-friendly app, there is no guarantee that users will adopt the app and make it their primary podcast listening platform.

5 APPLICATIONS :

Personal podcast listening: The app can be used by individuals to listen to their favorite podcasts in a personalized and user-friendly way.

Educational podcast listening: The app can be used by students and educators to listen to educational podcasts and supplement their learning.

Business podcast listening: The app can be used by professionals to listen to business-related podcasts and stay up-to-date with industry news and trends.

Entertainment podcast listening: The app can be used by anyone who enjoys listening to podcasts for entertainment purposes, such as comedy, sports, or pop culture.

Social sharing: The social sharing features of the app can enable users to share their favorite podcasts with their friends and family, increasing engagement and user acquisition.

Brand promotion: Brands can use the app as a platform to promote their podcasts and reach a wider audience.

Monetization: If the app gains a significant user base, it can be monetized through subscription fees or advertising, generating revenue for the app developers.

6 CONCLUSION

In conclusion, we are developing a podcast app for Android that is built on the Redux architecture. The app will provide users with a seamless and intuitive podcast listening experience, while also allowing them to customize the app's appearance with dynamic themes. The app will be designed to meet the needs of podcast listeners who are looking for a reliable and easy-to-use podcast app that also offers customization options. With features such as podcast discovery, subscriptions, playback, dynamic themes, user profiles, and social sharing, we believe this app will be a valuable tool for podcast enthusiasts.

7 FUTURE SCOPE

The future scope of the podcast app developed with Redux architecture and dynamic themes is significant, with several possibilities for further development and improvement. Here are some potential future scope areas:

Platform expansion: Currently, the app is only available on Android devices. The future scope of the app includes expanding the platform to iOS and other platforms, increasing the potential user base and market share.

Improved AI-based recommendations: The app can be improved by integrating AI-based recommendation engines that suggest podcasts based on user preferences and listening habits, making the app more personalized and user-friendly.

Integration with voice assistants: The app can be integrated with popular voice assistants such as Siri, Google Assistant, or Alexa, enabling users to listen to podcasts hands-free and improve the app's accessibility.

User-generated content: The future scope of the app includes incorporating user-generated content, such as user-generated podcasts or user-generated playlists, increasing engagement and user-generated content creation.

Monetization: As the app gains a significant user base, it can be monetized through sponsorships, merchandise sales, or other revenue-generating models.

Localization: The future scope of the app includes localizing the app's content to cater to different regions and cultures, increasing the app's global appeal and market share.

Enhanced social features: The social sharing features of the app can be enhanced by incorporating social listening, enabling users to listen to podcasts simultaneously and share their thoughts and opinions in real-time.

Overall, the podcast app developed with Redux architecture and dynamic themes has a significant future scope, and the app's success will depend on its ability to adapt to changing user needs and preferences and provide a personalized and user-friendly listening experience.

Source code:

Main activity:

```
package com.example.podcastplayer

import android.content.Context
import android.media.MediaPlayer
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.BorderStroke
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.em
import androidx.compose.ui.unit.sp
import com.example.podcastplayer.ui.theme.PodcastPlayerTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
            setContent {
                PodcastPlayerTheme {
                    // A surface container using the 'background' color from the theme
                    Surface(
                        modifier = Modifier.fillMaxSize(),
                        color = MaterialTheme.colors.background

                    ) {
                        playAudio(this)
                    }
                }
            }
        }
    }




@Composable
fun playAudio(context: Context) {

    Column(modifier = Modifier.fillMaxSize()) {

        Column(horizontalAlignment = Alignment.CenterHorizontally, verticalArrangement =
Arrangement.Center) {
            Text(text = "PODCAST",
                modifier = Modifier.fillMaxWidth(),
                textAlign = TextAlign.Center,
                color = Color(0xFF6a3ef9),
                fontWeight = FontWeight.Bold,
                fontSize = 36.sp,
                style = MaterialTheme.typography.h1,
                letterSpacing = 0.1.em

            )
        }

        Column(modifier = Modifier
            .fillMaxSize()
            .verticalScroll(rememberScrollState())) {


            Card(
                elevation = 12.dp,
                border = BorderStroke(1.dp, Color.Magenta),
                modifier = Modifier
                    .padding(16.dp)
                    .fillMaxWidth()
                    .height(250.dp)
            )
            {
```

```
    val mp: MediaPlayer = MediaPlayer.create(context, R.raw.audio)

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {

        Image(
            painter = painterResource(id = R.drawable.img),
            contentDescription = null,
            modifier = Modifier
                .height(150.dp)
                .width(200.dp),

            )

        Text(
            text = "GaurGopalDas Returns To TRS - Life, Monkhood & Spirituality",
            textAlign = TextAlign.Center,
            modifier = Modifier.padding(start = 20.dp, end = 20.dp)
        )
        Row() {

            IconButton(onClick = { mp.start() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.play),
                    contentDescription = ""
                )
            }

            IconButton(onClick = { mp.pause() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.pause),
                    contentDescription = ""
                )
            }

        }
    }

}
```

```
Card(
    elevation = 12.dp,
    border = BorderStroke(1.dp, Color.Magenta),
    modifier = Modifier
        .padding(16.dp)
        .fillMaxWidth()
        .height(250.dp)
)
{
    val mp: MediaPlayer = MediaPlayer.create(context, R.raw.audio_1)

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally

    ) {

        Image(
            painter = painterResource(id = R.drawable.img_1),
            contentDescription = null,
            modifier = Modifier
                .height(150.dp)
                .width(200.dp)
        )

        Text(
            text = "Haunted Houses, Evil Spirits & The Paranormal Explained | Sarbajeet Mohanty",
            textAlign = TextAlign.Center,
            modifier = Modifier.padding(start = 20.dp, end = 20.dp)
        )

        Row() {

            IconButton(onClick = { mp.start() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.play),
                    contentDescription = ""
                )
            }

            IconButton(onClick = { mp.pause() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.pause),
                    contentDescription = ""
                )
            }

        }
    }

}



Card(
    elevation = 12.dp,
    border = BorderStroke(1.dp, Color.Magenta),
    modifier = Modifier
        .padding(16.dp)
        .fillMaxWidth()
        .height(250.dp)
)
{
    val mp: MediaPlayer = MediaPlayer.create(context, R.raw.audio_2)

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally

    ) {

        Image(
            painter = painterResource(id = R.drawable.img_2),
            contentDescription = null,
            modifier = Modifier
                .height(150.dp)
                .width(200.dp)
        )

        Text(
            text = "Kaali Mata ki kahani - Black Magic & Aghoris ft. Dr Vineet Aggarwal",
            textAlign = TextAlign.Center,
            modifier = Modifier.padding(start = 20.dp, end = 20.dp)
        )

        Row() {

            IconButton(onClick = { mp.start() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.play),
                    contentDescription = ""
                )
            }

            IconButton(onClick = { mp.pause() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.pause),
                    contentDescription = ""
                )
            }

        }
    }

}


Card(
    elevation = 12.dp,
    border = BorderStroke(1.dp, Color.Magenta),
    modifier = Modifier
        .padding(16.dp)
        .fillMaxWidth()
        .height(250.dp)
)
{
    val mp: MediaPlayer = MediaPlayer.create(context, R.raw.audio_3)

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {

        Image(
            painter = painterResource(id = R.drawable.img_3),
            contentDescription = null,
            modifier = Modifier
                .height(150.dp)
                .width(200.dp),

            )

        Text(
            text = "Tantra Explained Simply | Rajarshi Nandy - Mata, Bhairav & Kamakhya Devi",
            textAlign = TextAlign.Center,
            modifier = Modifier.padding(start = 20.dp, end = 20.dp)
        )
        Row() {

            IconButton(onClick = { mp.start() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.play),
                    contentDescription = ""
                )
            }

            IconButton(onClick = { mp.pause() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.pause),
                    contentDescription = ""
                )
            }

        }
    }

}


Card(
    elevation = 12.dp,
    border = BorderStroke(1.dp, Color.Magenta),
    modifier = Modifier
        .padding(16.dp)
        .fillMaxWidth()
        .height(250.dp)
)
```

```
{
    val mp: MediaPlayer = MediaPlayer.create(context, R.raw.audio_4)

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {

        Image(
            painter = painterResource(id = R.drawable.img_4),
            contentDescription = null,
            modifier = Modifier
                .height(150.dp)
                .width(200.dp),

            )

        Text(
            text = "Complete Story Of Shri Krishna - Explained In 20 Minutes",
            textAlign = TextAlign.Center,
            modifier = Modifier.padding(start = 20.dp, end = 20.dp)
        )
        Row() {

            IconButton(onClick = { mp.start() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.play),
                    contentDescription = ""
                )
            }

            IconButton(onClick = { mp.pause() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.pause),
                    contentDescription = ""
                )
            }

        }
    }

}
```

```
Card(
    elevation = 12.dp,
    border = BorderStroke(1.dp, Color.Magenta),
    modifier = Modifier
        .padding(16.dp)
        .fillMaxWidth()
        .height(250.dp)
)
{
    val mp: MediaPlayer = MediaPlayer.create(context, R.raw.audio_5)

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally
    ) {

        Image(
            painter = painterResource(id = R.drawable.img_5),
            contentDescription = null,
            modifier = Modifier
                .height(150.dp)
                .width(200.dp),

            )

        Text(
            text = "Mahabharat Ki Poori Kahaani - Arjun, Shri Krishna & Yuddh - Ami Ganatra ",
            textAlign = TextAlign.Center,
            modifier = Modifier.padding(start = 20.dp, end = 20.dp)
        )
        Row() {

            IconButton(onClick = { mp.start() }, modifier = Modifier.size(35.dp)) {
                Icon(
                    painter = painterResource(id = R.drawable.play),
                    contentDescription = ""
                )
            }

            IconButton(onClick = { mp.pause() }, modifier = Modifier.size(35.dp)) {
```

```
                            Icon(
                                painter = painterResource(id = R.drawable.pause),
                                contentDescription = ""
                            )
                        }

                    }
                }

            }


        }
    }
}
```