Podcast Plus: A Redux-Inspired Podcast App with Dynamic Themes for Android

Listen in style. Podcasting made personal

Kaviya.K
Reg no:710422104022

OBJECTIVE

The goal of Podcast Plus is to provide Android users with a seamless and engaging podcast streaming experience by leveraging a Redux-inspired architecture for efficient state management. This ensures predictability, maintainability, and real-time updates across the app, enhancing overall performance. The app features dynamic themes, including light, dark, and customizable options, allowing users to personalize their interface. With an intuitive design, personalized recommendations, and offline support, Podcast Plus prioritizes user convenience and flexibility. Its modular and scalable structure ensures the app remains adaptable to future enhancements, delivering a reliable and visually appealing podcast experience.

MainActivity.kt

```kotlin
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
```

```kotlin
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)

                    }
                }
            }
        }
    }
```

```kotlin
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)
                }
            }
        }
    }
}
```

```kotlin
@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()
```

```kotlin
    .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 = ""
```

```kotlin
                )
            }

        }
    }

}


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
```

```
        )

                }

            }
        }

        }

        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
```

```kotlin
) {

    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 = ""
            )
```

```kotlin
        }

                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)
```

```kotlin
)
{
    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)
        )
```

```kotlin
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),

            )
```

```kotlin
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 = ""
        )
    }

}
}
```

```kotlin
}

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,
```
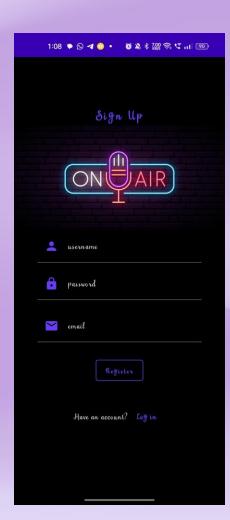
```kotlin
modifier = Modifier
        .height(150.dp)
        .width(200.dp),

    )

    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),
```

```kotlin
                    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(),
```

```kotlin
        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 = ""
                    )
                }
```

```kotlin
            }

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

            }
        }

        }

      }
    }
}
```

OUTPUT:

PODCAST

GaurGopalDas Returns To TRS - Life, Monkhood & Spirituality

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

Tantra Explained Simply | Rajarshi Nandy - Mata, Bhairav & Kamakhya Devi

Complete Story Of Shri Krishna - Explained In 20 Minutes

Mahabharat Ki Poori Kahaani - Arjun, Shri Krishna & Yuddh - Ami Ganatra

Conclusion

The completion of Podcast Plus: A Redux-Inspired Podcast App with Dynamic Themes for Android represents a significant achievement in enhancing the podcast listening experience. By incorporating Redux-inspired state management and dynamic theming, the app delivers seamless performance while allowing users to personalize their interface, creating a unique and engaging platform. This project showcases the synergy between modern design principles and robust development methodologies, offering a solution that is both functional and visually appealing. With its innovative approach, Podcast Plus sets a new standard for podcasting applications, blending technology and creativity to redefine how users interact with their favorite content.