

Podcast Plus

A Redux-Inspired Podcast App with Dynamic Themes

1. INTRODUCTION

1.1.Objective of the Project

With the rise in podcast consumption globally, users seek a streamlined and engaging podcast experience. The "Podcast Plus" app is designed to cater to modern listeners, combining a clean, dynamic UI with powerful playback features. The project focuses on delivering intuitive controls, easy access to podcast collections, and a personalized theme experience.

1.2.Key Features:

- **User Authentication:** Secure login and registration for personalized use.
- **Browse and Search:** Easily discover podcasts by browsing categories or using the search function.
- **Playback Controls:** Play, pause, resume, and seek within podcasts.
- **Dynamic Themes:** Customizable themes for personalized user experiences.
- **Favorites:** Save and access frequently played podcasts.

1.3.Project Goals:

- Build an intuitive and aesthetically pleasing podcast app using **Jetpack Compose**.
- Integrate **ExoPlayer** for robust media playback.
- Incorporate centralized state management to ensure smooth app performance.
- Optimize the app for Android devices running version 8.0 (Oreo) or later.

2. LITERATURE SURVEY

2.1. Existing Problems

While there are several podcast applications available, most suffer from the following limitations:

- **Complex Interfaces:** Overloaded with features, making navigation cumbersome.
- **Limited Personalization:** Lack of flexibility in adapting themes and layouts to user preferences.
- **Inconsistent Performance:** Issues with playback reliability, especially during multitasking or background usage.

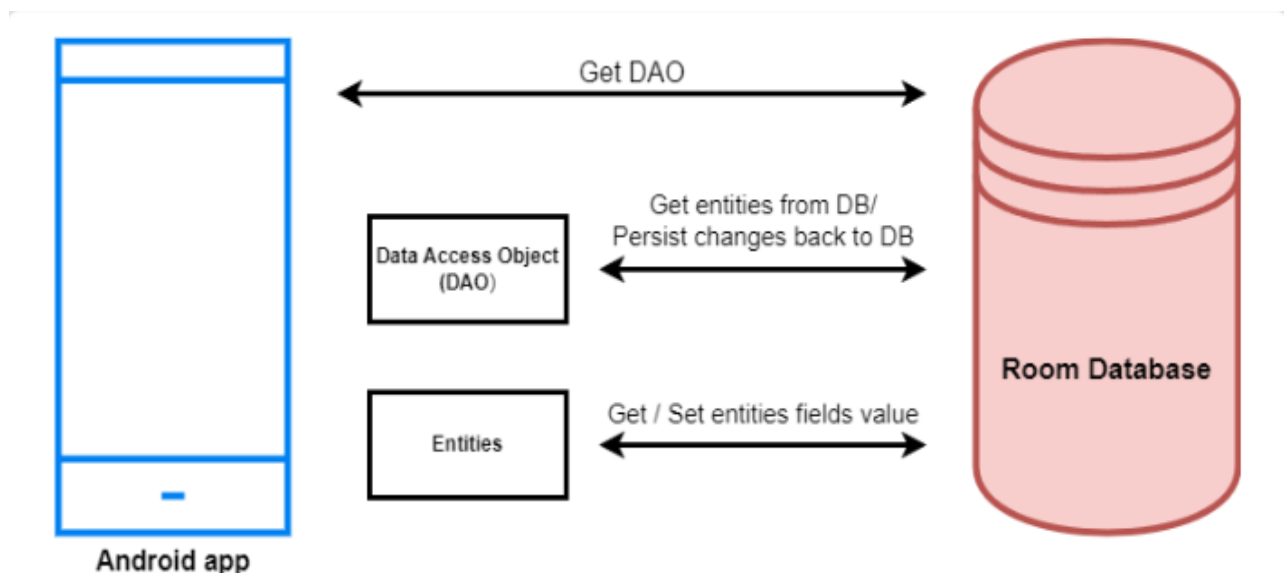
2.2. Proposed Solution

"Podcast Plus" addresses these challenges by introducing:

- **Simplified Navigation:** A clean, Compose-based UI to ensure ease of use.
- **Dynamic Themes:** Customizable themes that enhance user engagement and accessibility.
- **Reliable Playback:** Integrated ExoPlayer for a smooth and uninterrupted listening experience.
- **Optimized Architecture:** A Redux-inspired state management system to ensure scalability and maintainability.

3. SYSTEM DESIGN AND REQUIREMENTS

3.1. Architectural Design



3.1.1. Key components

- **Authentication Module:** Manages secure user registration and login.
- **Playback Module:** Uses ExoPlayer for audio playback.
- **Database Module:** SQLite or Room Database for storing user and podcast data.
- **UI Module:** Built with Jetpack Compose for modern, declarative UI development.

3.2. Workflow Description:

- **Authentication:** Users register or log in to access the app.
- **Podcast Selection:** The main page lists available podcasts, fetched via APIs or a local database.
- **Playback:** Users can play, pause, resume, or seek within a podcast.
- **Theme Customization:** Dynamic themes adapt to user preferences or system settings.

3.3. Hardware Requirements

Development Environment:

- **Processor:** Intel Core i5/i7 or equivalent.
- **RAM:** Minimum 8 GB (16 GB recommended).
- **Storage:** SSD (minimum 512 GB).
- **Display:** Full HD (1920x1080) resolution or higher.
- **Smartphone:** Android device (version 8.0 or newer) for testing.

End-User Device:

- **Operating System:** Android 8.0 or newer.
- **RAM:** Minimum 2 GB (4 GB recommended).
- **Storage:** 50 MB free for app installation and user data.
- **Connectivity:** Wi-Fi or mobile data for streaming podcasts.

3.3. Software Requirements

Development Environment:

- **Programming Language:** Kotlin.
- **IDE:** Android Studio Arctic Fox or newer.
- **Libraries:** Jetpack Compose, Retrofit, Room, ExoPlayer.
- **Version Control:** Git for collaboration and versioning.

End-User Environment:

- **Distribution Platform:** Google Play Store.
- **Permissions:** Internet, storage, and notifications.

4. PROJECT WORKFLOW

4.1. Functional Overview

☐ Authentication:

- New users register by providing personal details.
- Existing users log in with secure credentials.

☐ Main Page:

- Display podcasts categorized by genres.
- Allow searching for specific podcasts.

□ Playback Features:

- Users can play, pause, and resume podcasts.
- Seek functionality enables skipping forward or backward.

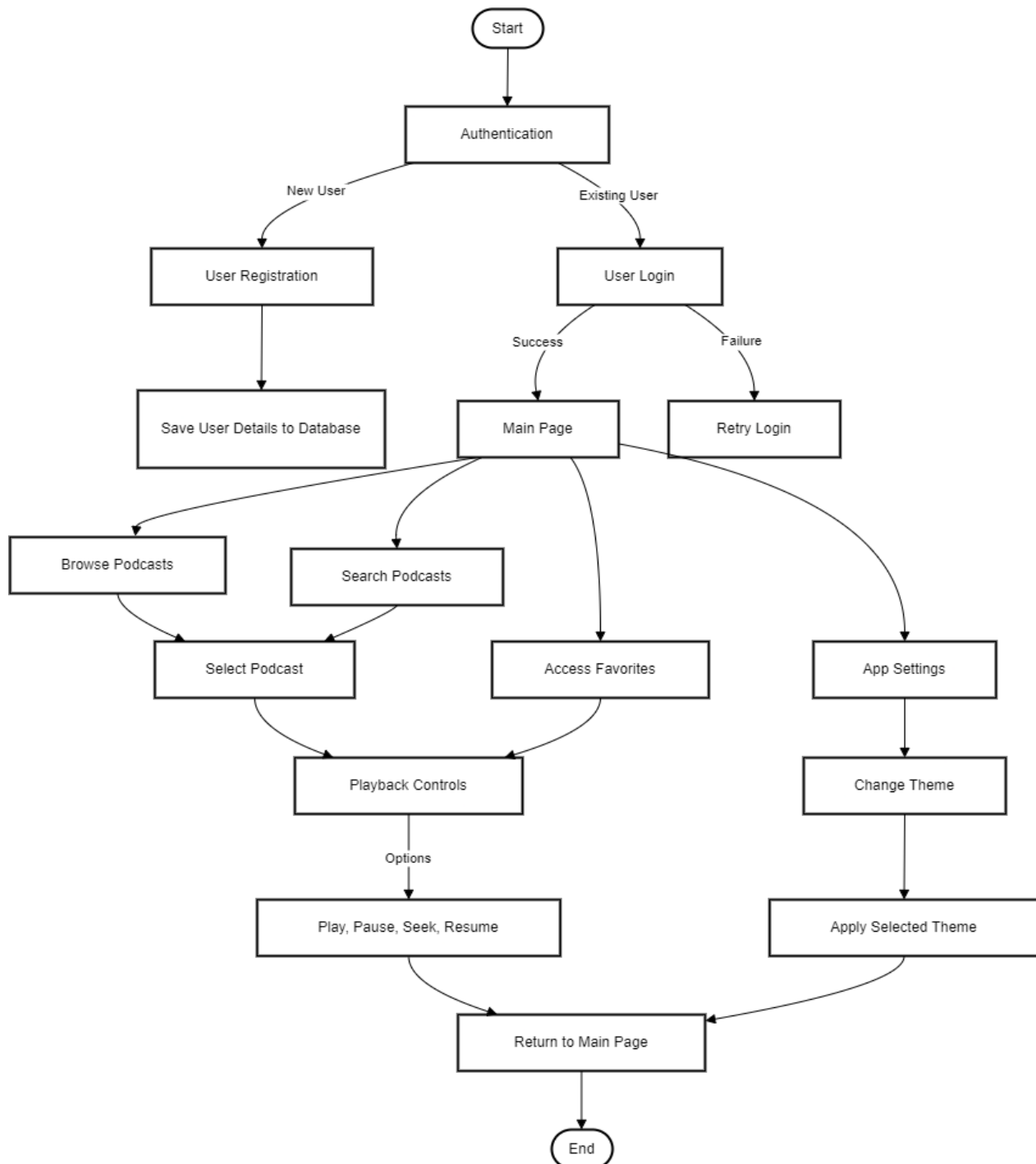
□ Dynamic Themes:

- Users can customize themes from settings.

□ Favorites:

- Users can save favorite podcasts for quick access.

5.FLOWCHART



6.Program

MainActivity.kt

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

```

7. RESULTS

The *Podcast Plus* app has successfully demonstrated:

- A modern, responsive UI using Jetpack Compose.
- Robust audio playback powered by ExoPlayer.
- Dynamic themes for an engaging user experience.

8. ADVANTAGES AND DISADVANTAGES

8.1. Advantages

- **User-Friendly:** The Compose-based UI simplifies navigation and enhances user experience.
- **Dynamic Design:** Themes provide aesthetic flexibility, ensuring accessibility.
- **Efficient Performance:** Optimized state management ensures minimal resource usage.

8.2. Disadvantages

- **Platform Dependency:** Currently limited to Android devices.
- **Internet Reliance:** Requires an active connection for streaming content.

9.Applications of Podcast Plus

1. Entertainment:

- Serves as a platform for users to discover and enjoy their favorite podcasts, ranging from comedy to drama.

2. Education:

- Provides access to educational and informative podcasts, such as language learning, history lessons, or tech insights.

3. Professional Development:

- Offers podcasts focused on career growth, industry trends, and skills development.

4. News and Current Events:

- Acts as a reliable source for staying updated on daily news, politics, and global trends.

5. Mental Health and Wellness:

- Features wellness and mindfulness podcasts to help users relax and unwind.

6. Marketing and Branding for Podcasters:

- Allows podcasters to reach a wider audience and promote their brand effectively.

7. **Community Building:**

- Encourages users to engage with niche communities by joining fan groups or discussion forums.

8. **Accessibility:**

- Offers a platform for people with disabilities to access audio-based content easily.

10. **Future Scope of Podcast Plus**

1. **Cross-Platform Expansion:**

- **iOS Support:** Expanding the app to iOS devices to capture a broader audience.
- **Web Version:** Building a responsive web application for desktop and browser users.

2. **Offline Mode:**

- Introducing offline downloads for podcasts to enable playback without internet connectivity.

3. **Personalized Recommendations:**

- Utilizing **machine learning** algorithms to suggest podcasts based on user interests, history, and trends.

4. **Enhanced Playback Features:**

- **Speed control:** Allow users to listen at various playback speeds.
- **Chapter markers:** For easy navigation within episodes.
- **Sleep timer:** Automatically stop playback after a set time.

5. **Monetization Options:**

- Adding a subscription model for ad-free listening and exclusive content.
- Support for in-app purchases like premium podcasts or bundles.

6. **Social Sharing and Collaboration:**

- Features to share favorite podcasts or create playlists with friends.
- Integration with social media platforms for better community engagement.

7. **Integration with Smart Devices:**

- Support for smart speakers (e.g., Amazon Echo, Google Nest) and wearables.
- Voice command support for hands-free navigation.

8. **Podcast Creator Tools:**

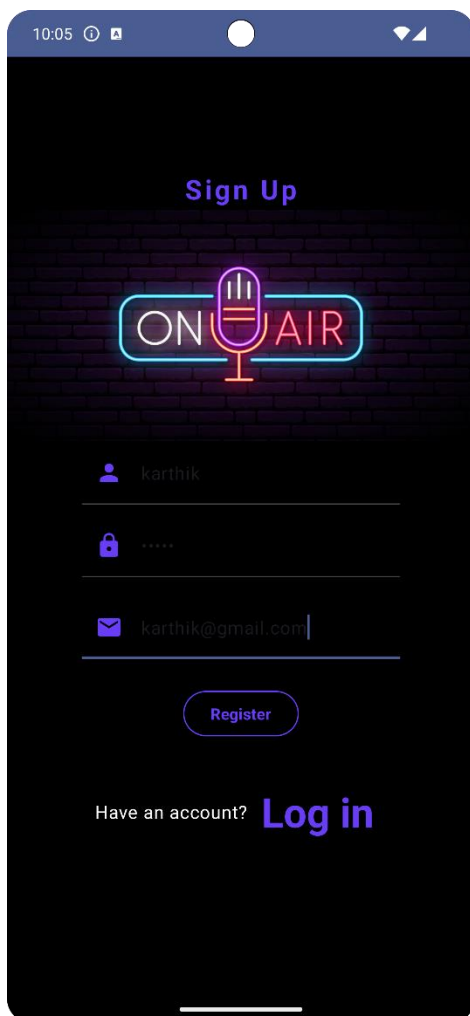
- Provide features for podcasters to upload and manage their content directly within the app.

9. Improved Analytics:

- Display in-depth stats like most-played episodes, listener trends, and time spent on the app.
- Offer podcast creators insights into listener engagement.

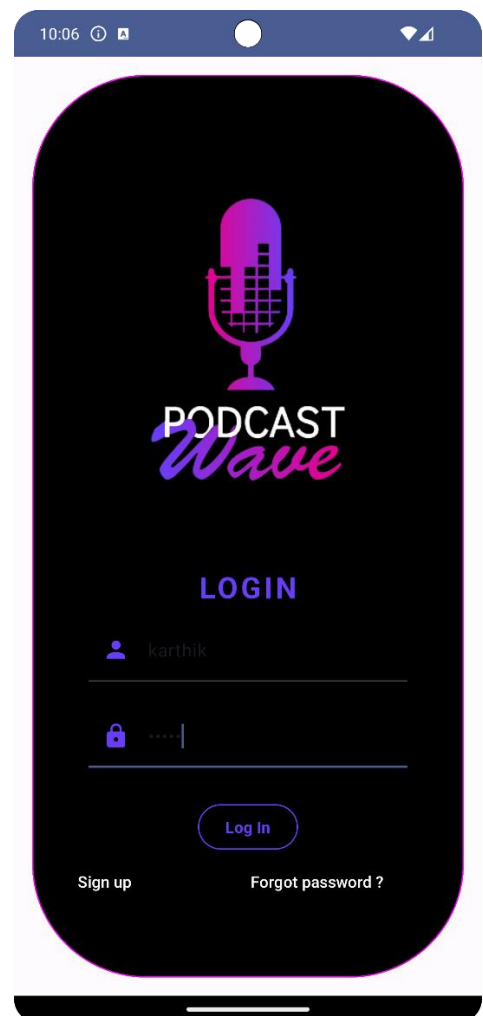
10.SCREENSHOTS

Sign up Page



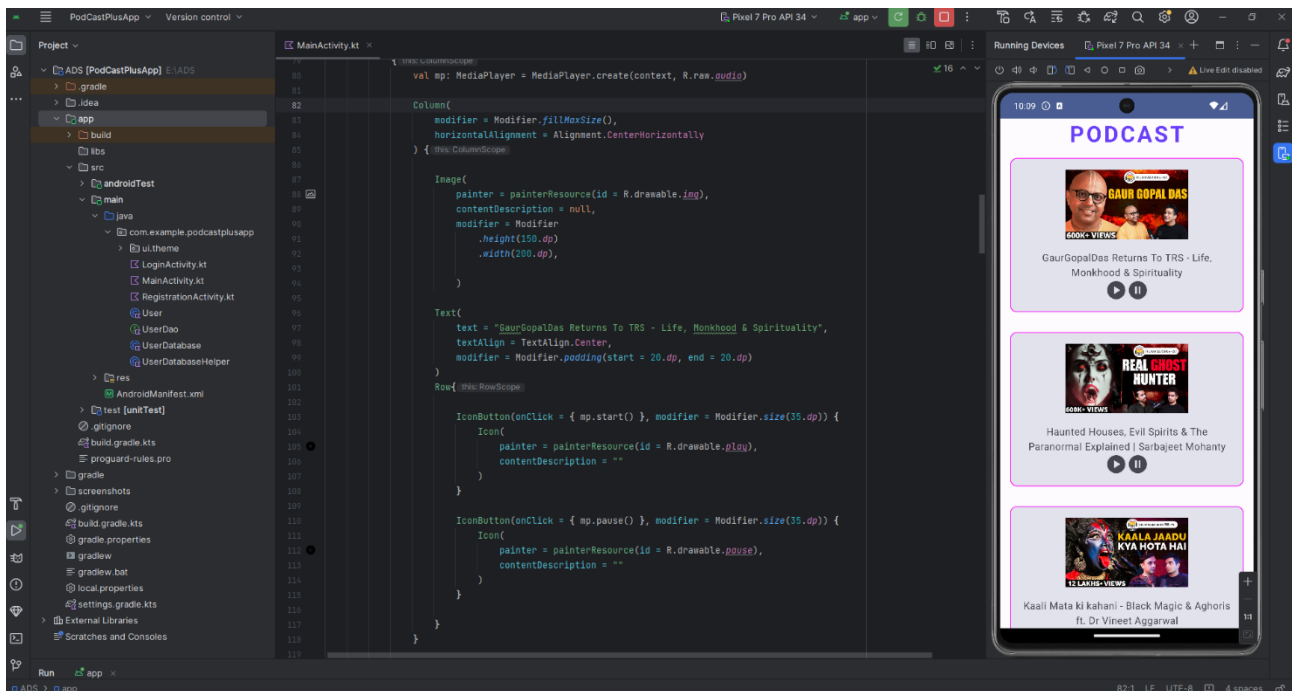
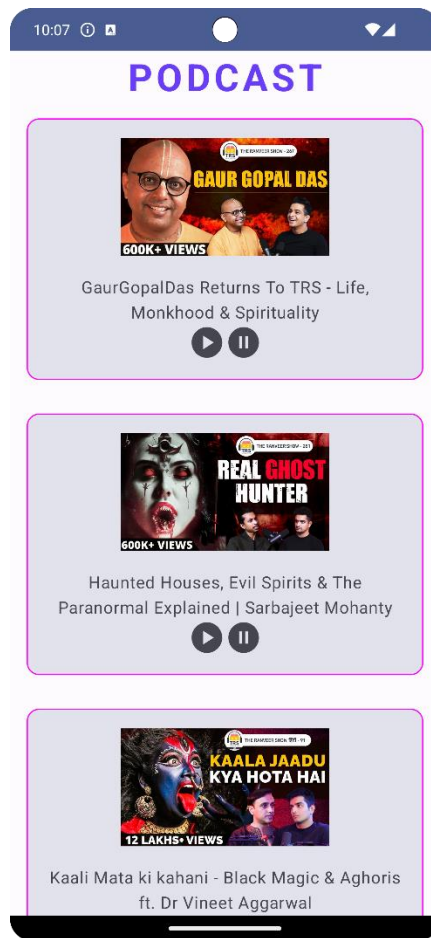
A screenshot of a mobile app's sign-up page. The background is dark with a brick wall texture. At the top, the text "Sign Up" is displayed in a light blue font. Below it is a logo featuring a microphone and the words "ON AIR" in a stylized, glowing font. The form consists of three input fields: a username field with "karthik", a password field with "*****", and an email field with "karthik@gmail.com". Each field has a corresponding icon (person, lock, and envelope) to its left. Below the email field is a blue "Register" button. At the bottom, there is a link that says "Have an account? Log in" in a light blue font.

Login Page



A screenshot of a mobile app's login page. The background is dark with a brick wall texture. At the top, the text "PODCAST Wave" is displayed in a light blue font, with "PODCAST" in a sans-serif font and "Wave" in a script font. Below it is a logo featuring a microphone. The form consists of two input fields: a username field with "karthik" and a password field with "*****". Each field has a corresponding icon (person and lock) to its left. Below the password field is a blue "Log In" button. At the bottom, there are two links: "Sign up" and "Forgot password ?" in a light blue font.

HomePage



12. CONCLUSION

The *Podcast Plus* app is a robust and user-friendly solution for podcast enthusiasts. By leveraging the latest Android technologies such as Jetpack Compose and ExoPlayer, the app ensures an engaging and reliable user experience. Future enhancements may include:

- Offline podcast downloads.
- Cross-platform support for iOS.
- Personalized recommendations based on user preferences and history.