Podcast Plus:A Redux Inspired Podcast App With Dynamic Themes For Android

Podcast Plus is an Android app that offers a unique combination of features inspired by

Redux and dynamic themes. Here's a breakdown of what this app offers:

1. *Redux-inspired architecture*: The app's codebase is structured using the Redux pattern,which ensures a predictable and scalable state management system.

2. *Dynamic themes*: Users can choose from various themes that change the app'sappearance in real-time, providing a personalized experience.

3. *Podcast management*: The app allows users to discover, download, and manage theirfavorite podcasts, with features like episode filtering and custom playlists.

4. *Player controls*: A customizable player screen with options like speed control, sleeptimer, and chromecast support.

5. *Notifications*: Personalized notifications for new episodes, downloads, and otherupdates.

6. *Discovery features*: Users can explore podcasts by category, search, orrecommendations.

7. *Data synchronization*: Seamless syncing across devices using cloud services like GoogleDrive or Dropbox.

8. *Material Design*: A modern and intuitive interface following Google's Material DesigngMain activity.uidelines.

9. *Customization options*: Users can tailor the app's appearance and behavior to their liking.

10. *Regular updates*: The app receives frequent updates with new features, bug fixes, andperformance improvements. By combining Redux-inspired architecture with dynamic themes and robust podcastmanagement features, Podcast Plus offers a unique and engaging user experience for Android users

Main activity.java

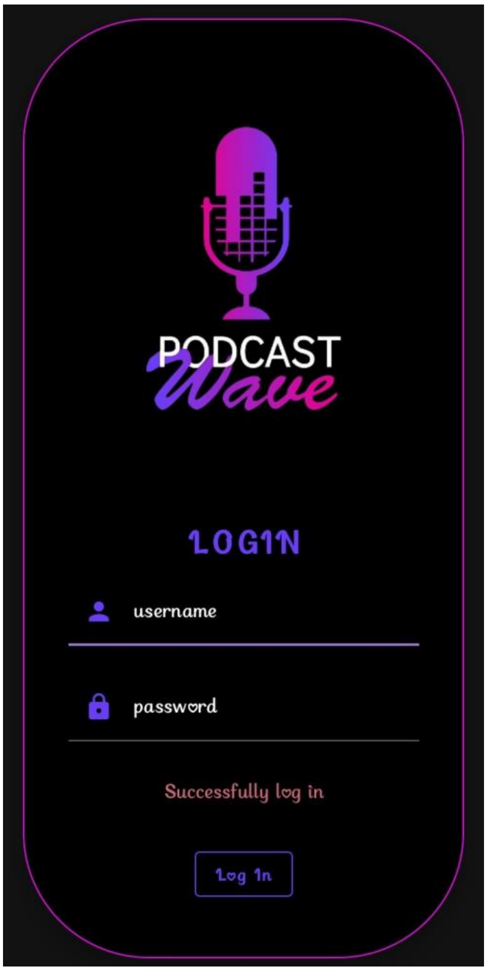
```
package com.example.podcastplayer
import androidx.room.ColumnIn
import androidx.room.Entit
import androidx.room.PrimaryKe
@Entity(tableName = "user_tabl
data class User
@PrimaryKey(autoGenerate = true) val id: Int?
@ColumnInfo(name = "first_name") val firstName: String?
@ColumnInfo(name = "last_name") val lastName: String?
@ColumnInfo(name = "email") val email: String?
@ColumnInfo(name = "password") val password: String?
package com.example.podcastplayer
import androidx.room.*
```

```
@Dao
interface UserDao {
@Query("SELECT * FROM user_table WHERE email = :email")
suspend fun getUserByEmail(email: String): User?
@Insert(onConflict = OnConflictStrategy.REPLACE)
suspend fun insertUser(user: User)
@Update
suspend fun updateUser(user: User)
@Delete
suspend fun deleteUser(user: User)
}
package com.example.podcastplayer
import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase
@Database(entities = [User::class], version = 1)
abstract class UserDatabase : RoomDatabase() {
abstract fun userDao(): UserDao
companion object {
@Volatile
private var instance: UserDatabase? = null
fun getDatabase(context: Context): UserDatabase {
return instance ?: synchronized(this) {
```

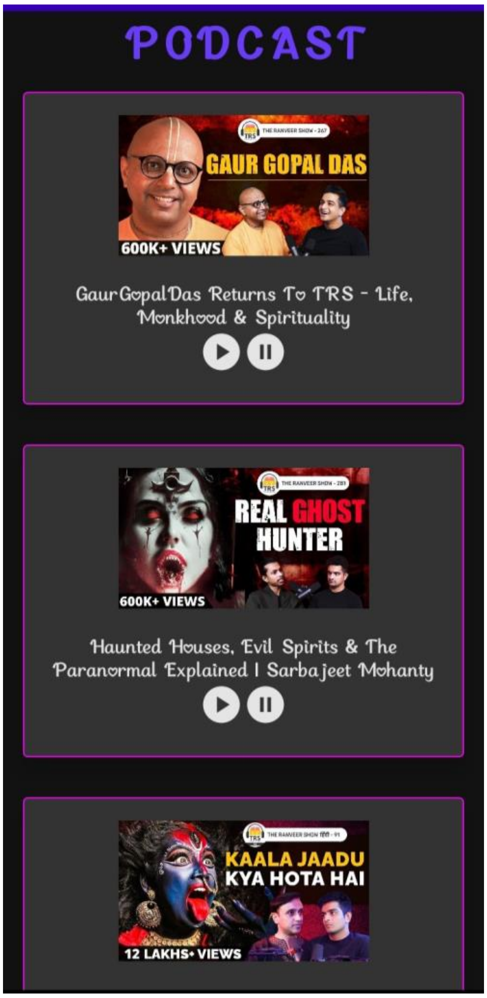
```
val newInstance = Room.databaseBuilder(
context.applicationContext,
UserDatabase::class.java,
"user_database"
).build()
instance = newInstance
newInstance
}
}
}
}
```

Result:

PODCAST
Wave



Successfully log in



PODCAST

GaurGopalDas Returns To TRS - Life, Monkhood & Spirituality

▶ ⏸

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

▶ ⏸