parental Dashboard



Code:

@Composable

fun ParentalDashboard() {

Column(

modifier = Modifier

.fillMaxSize()

.background(Color(0xFFF8F4EF))

.padding(16.dp)

) {

Text(

text = "PARENTAL DASHBOARD",

fontSize = 24.sp,

fontWeight = FontWeight.Bold,

color = Color.White,

modifier = Modifier

.fillMaxWidth()

.background(Color(0xFF7E6AD1), shape = RoundedCornerShape(16.dp))

.padding(16.dp)

)

Text(

text = "Kids Educational Game • Age 3–6",

fontSize = 14.sp,

color = Color.White,

modifier = Modifier

.fillMaxWidth()

.background(Color(0xFF7E6AD1))

.padding(bottom = 16.dp)

)

Row(

modifier = Modifier.fillMaxWidth(),

horizontalArrangement = Arrangement.SpaceEvenly

) {

DashboardCard(title = "SCREEN LIMIT", icon = Icons.Default.Timer, content = "1h\n30m", buttonText = "SET", color = Color(0xFFF8972B))

DashboardCard(title = "TRACK PROGRESS", icon = Icons.Default.ShowChart, content = "", buttonText = "VIEW DETAILS", color = Color(0xFF3CBAC8))

DashboardCard(title = "REWARDS", icon = Icons.Default.Star, content = "1350", buttonText = "MANAGE", color = Color(0xFF3FB97E))

}

Spacer(modifier = Modifier.height(24.dp))

Card(

backgroundColor = Color(0xFF7E6AD1),

shape = RoundedCornerShape(16.dp),

modifier = Modifier.fillMaxWidth()

) {

Row(

modifier = Modifier.padding(16.dp),

verticalAlignment = Alignment.CenterVertically

) {

Icon(

imageVector = Icons.Default.Person,

contentDescription = null,

modifier = Modifier.size(48.dp),

tint = Color.White

)

Spacer(modifier = Modifier.width(16.dp))

Column {

Text("ACTIVITY", color = Color.White, fontWeight = FontWeight.Bold)

Text("Learning:", color = Color.White)

LinearProgressIndicator(

progress = 0.7f,

modifier = Modifier

.fillMaxWidth(0.7f)

.height(10.dp),

color = Color.Yellow,

backgroundColor = Color.LightGray

)

}

}

}

}

}

@Composable

fun DashboardCard(title: String, icon: ImageVector, content: String, buttonText: String, color: Color) {

Card(

shape = RoundedCornerShape(16.dp),

elevation = 4.dp,

modifier = Modifier

.width(110.dp)

.padding(8.dp)

) {

Column(

modifier = Modifier

.padding(12.dp),

horizontalAlignment = Alignment.CenterHorizontally

) {

Text(title, fontWeight = FontWeight.Bold, color = color, fontSize = 14.sp)

Spacer(modifier = Modifier.height(8.dp))

Icon(icon, contentDescription = null, tint = color, modifier = Modifier.size(32.dp))

if (content.isNotEmpty()) {

Text(content, fontSize = 20.sp, fontWeight = FontWeight.Bold)

}

Spacer(modifier = Modifier.height(8.dp))

Button(

onClick = {},

colors = ButtonDefaults.buttonColors(backgroundColor = color),

shape = RoundedCornerShape(50)

) {

Text(buttonText, color = Color.White)

}

}

Great! You're building a **comprehensive parental dashboard in Kotlin** for the KinderJoy game — that's an excellent idea for tracking and guiding a child's learning journey.

To get you started, here's a basic **structure** and **Kotlin + Android Jetpack Compose code** example covering the key sections:

**✅ Features to Include:**

1. **Screen Time Limit**
2. **Progress Tracking (Last 10 Days)**
3. **Daily & Weekly Growth Graph**
4. **Time Spent Per Section**
5. **Learning Activity**
6. **Weak Areas in Learning**
7. **Database Integration (Room DB or Firebase)**

**🧩 Kotlin + Jetpack Compose UI Skeleton:**

kotlin

CopyEdit

@Composable

fun ParentalDashboardScreen() {

Column(modifier = Modifier.padding(16.dp)) {

ScreenTimeLimitCard()

Spacer(modifier = Modifier.height(16.dp))

ProgressLast10DaysChart()

Spacer(modifier = Modifier.height(16.dp))

DailyWeeklyGrowthGraph()

Spacer(modifier = Modifier.height(16.dp))

TimeSpentPerSection()

Spacer(modifier = Modifier.height(16.dp))

LearningActivitySection()

Spacer(modifier = Modifier.height(16.dp))

WeakAreasSection()

}

}

**📊 Sample Section: Screen Time Limit**

kotlin

CopyEdit

@Composable

fun ScreenTimeLimitCard() {

Card(

elevation = 4.dp,

shape = RoundedCornerShape(16.dp),

backgroundColor = Color(0xFFFFE0B2)

) {

Column(modifier = Modifier.padding(16.dp)) {

Text("Screen Limit", fontWeight = FontWeight.Bold)

Text("1h 30m", fontSize = 24.sp, fontWeight = FontWeight.Bold)

Button(onClick = { /\* Show time picker \*/ }) {

Text("Set Limit")

}

}

}

}

**📈 Sample: Progress Last 10 Days (Using MPAndroidChart or Accompanist Graphs)**

* You can use MPAndroidChart for Kotlin to show line graphs.
* The graph will show daily scores or activity levels.

**🧠 Weak Areas Detection**

kotlin

CopyEdit

@Composable

fun WeakAreasSection() {

Card(

elevation = 4.dp,

shape = RoundedCornerShape(16.dp),

backgroundColor = Color(0xFFFFCDD2)

) {

Column(modifier = Modifier.padding(16.dp)) {

Text("Weak Learning Areas", fontWeight = FontWeight.Bold)

Text("• Colors\n• Alphabets\n• Puzzles", fontSize = 16.sp)

}

}

}

**💾 Room Database Example (for storing history locally)**

**Entity:**

kotlin

CopyEdit

@Entity(tableName = "activity\_logs")

data class ActivityLog(

@PrimaryKey(autoGenerate = true) val id: Int = 0,

val date: String,

val screenTimeMinutes: Int,

val learningScore: Int,

val weakArea: String?

)

**DAO:**

kotlin

CopyEdit

@Dao

interface ActivityLogDao {

@Insert(onConflict = OnConflictStrategy.REPLACE)

suspend fun insertLog(log: ActivityLog)

@Query("SELECT \* FROM activity\_logs ORDER BY date DESC")

fun getAllLogs(): Flow<List<ActivityLog>>

}

**Database:**

kotlin

CopyEdit

@Database(entities = [ActivityLog::class], version = 1)

abstract class AppDatabase : RoomDatabase() {

abstract fun activityLogDao(): ActivityLogDao

}

Would you like:

* A **Firebase-based implementation** instead of Room?
* A **full working project structure** with navigation and charts?
* Or should I build out each section one-by-one for you?

Bottom of Form