Tab 1

✅ FINAL DRIVER COMMISSION MODEL — LOYAL SAFAR

🎯 OBJECTIVE

🔹 Ride fare ke hisab se **dynamic slab-based commission** lage.
🔹 Har slab mein **type choose kar sakte ho — Fixed ya Percentage**.
🔹 Har area ka apna slab set ho.
🔹 Agar **range match nahi hoti**, toh **area ka default rule** lage.
🔹 Saare calculation **auto backend pe** ho, driver ko hamesha final payout mile.

🔹 1️⃣ TABLES

A) Areas Table

Field	Details
id	Primary Key
name	e.g. "Kolkata", "Howrah"

B) Commission Slabs Table

Field	Details
id	Primary Key
area_id	kis area ke liye hai
min_fare	Minimum fare (inclusive)
max_fare	Maximum fare (inclusive)
commission_type	Fixed / Percentage
commission_value	Fixed amount ya Percentage value
is_default	TRUE / FALSE
active	TRUE / FALSE

C) Rides Table (Main)

Field	Details
id	Ride ID
area_id	Kis area mein ride hui
total_fare	Actual fare
commission_amount	Kitna commission katega
commission_type	Fixed / Percentage
driver_payout	Total payout to driver

🔹 2️⃣ EXAMPLE SLAB

Area	Min Fare	Max Fare	Type	Value	Default?
Kolkata	₹0	₹400	Fixed	₹40	❌
Kolkata	₹401	₹600	Percentage	8%	❌
Kolkata	₹601	₹1000	Fixed	₹100	❌
Kolkata	—	—	Percentage	10%	✅
Howrah	₹0	₹500	Fixed	₹50	❌
Howrah	₹501	₹800	Percentage	12%	❌
Howrah	—	—	Percentage	15%	✅

🔹 3️⃣ KAAM KA TARIKA (WORKFLOW)

✅ **Step 1:**
Ride complete hone par:

- `area_id` aur `total_fare` milega.

✅ **Step 2:**
`commission_slabs` mein dekho:

- **Jo slab ka min-max fare match karta hai**, woh pick karo.
    - e.g. Ride ₹450 hai → Kolkata area → `401–600` slab apply → 8% lagega.

✅ **Step 3:**
Agar koi slab match nahi karta:

- `is_default = TRUE` wala rule use karo us area ka.

✅ **Step 4:**
Commission calculate karo:

- **Fixed:** wohi amount.
- **Percentage:** `(percentage_value/100) * fare`

✅ **Step 5:**
Driver payout:

- `driver_payout = total_fare - commission_amount`

✅ **Step 6:**
Rides table mein save karo:

- `commission_amount`
- `commission_type`
- `driver_payout`

✅ **Step 7:**
Driver wallet mein `driver_payout` credit karo.

🔹 4️⃣ IMPORTANT POINTS

✅ Har area ke liye aap unlimited **slabs** bana sakte ho.
✅ Ek hi area mein **slabs overlap nahi hone chahiye** — developer ko unique check lagana hai.
✅ Har area ke liye **default rule compulsory** — warna system confuse hoga.
✅ Admin panel mein CRUD: Add/Edit/Delete slabs.
✅ Ride complete event pe backend pe auto calculation.
✅ Driver ko payout aur commission report dikhao — transparency.

✅ EXAMPLES

Ride Fare	Area	Slab	Commission	Driver Gets
₹350	Kolkata	₹0–₹400	Fixed ₹40	₹310
₹500	Kolkata	₹401–₹600	8% → ₹40	₹460
₹850	Kolkata	₹601–₹1000	Fixed ₹100	₹750
₹1200	Kolkata	Default 10% → ₹120	₹1080	
₹450	Howrah	₹0–₹500	Fixed ₹50	₹400
₹750	Howrah	₹501–₹800	12% → ₹90	₹660
₹950	Howrah	Default 15% → ₹142.5	₹807.5	

🔹 5️⃣ ADMIN PANEL MEIN DEKHNE KI CHEEZ

✅ Area list
✅ Slab list for each area
✅ Slab ka:

- min_fare, max_fare

- type (Fixed/Percentage)
- value
- default flag
  ✅ Edit / Delete option
  ✅ Duplicate check (overlapping range same area mein na ho)

✅ BENEFITS

✔️ Simple aur powerful logic.
✔️ Area wise full control.
✔️ Low fare, high fare ke liye alag rule.
✔️ Easy to change — no code change.
✔️ Driver ko clear payout.

🔹 CORE RULE

Ride Fare RANGE match = Commission slab apply, else default!

✅ FINAL MODEL

✔️ **Type:** Fixed ya Percentage
✔️ **Fare range:** Min–Max
✔️ **Area-wise:** Alag slabs
✔️ **Default fallback**
✔️ **One slab per range per area, no overlap**

Tab 2

☐Share location option

🚖 Loyal Safar – Ride Tracking Share Feature

🔐 Triggered *only after OTP verification (Ride Start)*

📲 Step-by-Step Flow (App UX):

✅ 1. User books ride & enters OTP

- **Ride officially starts**
- **Backend activates live tracking session (ride ID–based)**

🔗 2. App shows: "Share Your Live Location"

- **Subtitle: *Keep your loved ones updated & safe***
- **Options:**
  - 📱 **WhatsApp**
  - 📧 **SMS**
  - 📞 **Saved Contact (Emergency/Family)**
  - 🔗 **Copy Link**

🌐 Shared Link Details:

Example link:

ruby
CopyEdit

```
https://loyalsafar.com/track/ride/983736?token=abcXYZ123
```

👀 What the Receiver Will See in the Link:

🔒 Live Session View (Bound to Active Ride Session):

👥 Top Section – Rider & Driver Details

yaml
CopyEdit

```
🧍 Rider: Ankit Sharma
📍 Pickup: Connaught Place, Delhi
🎯 Drop: HUDA City Centre, Gurgaon
```

🚖 Driver Profile

- 👤 Driver Name: Rajesh Kumar
- 🚘 Vehicle: White Honda Activa
- 🔢 Plate No.: DL5S AB 1234
- 📞 Contact (masked): Call Driver
- ⭐ Driver Rating: ★ 4.9

🗺️ Live Map

- 🔵 Live driver location icon
- 📍 Route line (same session used by app)
- ✅ Pickup & drop pins

⏱️ Other Info:

- Estimated time of arrival (ETA)
- Distance left (km)
- Ride start time

🧠 Technical Working:

🎯 Session Handling (Backend - Laravel)

- OTP ke baad `ride_id` ke basis par ek session key assign hota hai
- Ye session real-time location Firebase/Socket se sync karta hai
- Shared link use karta hai usi session ka listener, koi naya session nahi banta

📦 Token contains:

- **Ride ID**
- **Start timestamp**
- **Signature or JWT token**
- **Expiry limit (e.g., ride end + 30 minutes)**

✅ Security & Cost Optimization

Feature	Purpose
Single session for rider & viewer	**No duplicate server load**

Masked contact numbers	**Privacy**
Session expires automatically	**Data protection**
One-time encrypted token	**Security**

🔧 Bonus: Admin Panel Control

☐ Show: How many people shared location

☐ Show: Who viewed the link (if possible)

☐ Logs: Which ride links were accessed & when

Tab 3

✅ LOYAL SAFAR COUPON + DRIVER PAYOUT ROADMAP

🔹 1️⃣ Objective

- Jab bhi user coupon use kare:
  - User ko discount mile.
  - Driver ko **full ride amount** mile.
  - Jo discount diya gaya woh **company/marketing fund** se cover ho.
  - Sab ka **proper accounting & reason** ho database mein.

🔹 2️⃣ Required Tables

Developer ko **4 main tables** banana hain ya existing mein columns add karne hain:

1️⃣ rides

- `id, user_id, driver_id, total_fare, coupon_code, coupon_discount, final_fare, payment_status, ride_status`

2️⃣ driver_wallets

- `id, driver_id, ride_id, amount, transaction_type (credit/debit), reason, source, created_at`

3️⃣ coupon_redemptions

- `id, user_id, ride_id, coupon_code, discount_amount, covered_by, created_at`

4️⃣ **company_wallets** (internal ledger)

- `id, ride_id, driver_id, amount, transaction_type (credit/debit), reason, created_at`

🔹 3️⃣ Working Steps

Step 1:

- User ride book karta hai.
- System coupon validate karta hai (expiry, usage limit, amount).

Step 2:

- Fare calculate hota hai:
  - `total_fare` (e.g. ₹500)
  - `coupon_discount` (e.g. ₹100)
  - `final_fare = total_fare - coupon_discount` (₹400)

Step 3:

- User payment gateway se `final_fare` pay karta hai (₹400).
- Discounted `₹100` system mein **Company Wallet** se cover hota hai.

Step 4:

- Jab ride complete hoti hai:
  - `driver_wallets` mein **poora `₹500` credit** hota hai.

    - `reason`: "Ride Fare with Coupon Covered"
    - `source`: "User Paid + Company Coupon"
    - `ride_id` bhi save hota hai.

Step 5:

- `company_wallets` mein **₹100 ka debit** record hota hai:
    - `reason`: "Coupon Redeem for Ride #ID"

Step 6:

- `coupon_redemptions` mein ek record save hota hai:
    - `user_id, ride_id, coupon_code, discount_amount, covered_by`.

🔹 4️⃣ Dashboard & Report

- Developer ko ek **admin panel view** banana hoga:
    - Total Coupons Given.
    - Total Company Wallet Debit for Coupons.

    - Driver Wallet Entries with Source = "Coupon Covered".
    - Ride wise coupon redemption history.

🔹 5️⃣ Accounting Point

- Ye model **driver commission** se independent hai.
- Driver ko hamesha **full ride fare** milega.
- Aapko bas company ya marketing cost mein ye discount dikhana hai.
- Cash flow mismatch nahi hoga — user se kam paisa aayega, baaki aapki company fund se jaayega.

✅ Team Instructions

Developer Tasks:

1. New fields/tables migrate karo.
2. Payment success hone par coupon ka effect save karo.
3. Ride complete hone par driver payout auto credit karo.
4. Coupon amount company wallet se auto debit karo.
5. Proper reports bana do — so koi dispute na ho.

Testing Checklist:
✔️ Coupon Valid/Invalid.
✔️ Multiple coupons restriction.
✔️ Driver ko galat amount na mile.
✔️ Duplicate payout entry na bane.
✔️ Reports cross-check.

🟢 Summary:

User Pays: Final Fare after discount
Driver Gets: Full Fare
Company Bears: Coupon discount
Records: Proper linked `ride_id`, `reason`, `source`.