

# Yestoryd Revenue Split & Tax Structure Implementation

## Complete Guide: 3-Component Model, LLP Setup & CA Requirements

**Created:** December 18, 2025

**For:** Amit Kumar Rai & Rucha

**Status:** Ready for Implementation

---

## Table of Contents

1. [3-Component Revenue Split Model](#)
  2. [Database Schema](#)
  3. [Admin Interface Design](#)
  4. [API Endpoints](#)
  5. [Payout Processing Logic](#)
  6. [LLP Registration Guide](#)
  7. [TDS Compliance Framework](#)
  8. [CA Selection Criteria](#)
  9. [Implementation Roadmap](#)
- 

## 1. 3-Component Revenue Split Model

### Overview

The new model divides each enrollment fee (₹5,999) into three distinct components with clear accounting separation:

₹5,999 ENROLLMENT FEE

LEAD COST	COACH COST	PLATFORM
FEE		
Configurable	Configurable	Auto-calc
Default: 20%	Default: 50%	Remainder
₹1,200	₹3,000	₹1,799
→ To lead source	→ To coach (always)	→ Retained by LLP

## Split Scenarios

Scenario	Lead Cost (20%)	Coach Cost (50%)	Platform (30%)	Net to Coach	Net to Yestoryd
<b>Yestoryd Lead</b>	₹1,200 → Yestoryd	₹3,000 → Coach	₹1,799 → Yestoryd	₹3,000 (50%)	₹2,999 (50%)
<b>Coach Lead</b>	₹1,200 → Coach	₹3,000 → Coach	₹1,799 → Yestoryd	₹4,200 (70%)	₹1,799 (30%)
<b>Rucha Coaching + Yestoryd Lead</b>	₹1,200 → Yestoryd	₹3,000 → Rucha (internal)	₹1,799 → Yestoryd	₹3,000 (salary/drawing)	₹2,999

## Benefits of 3-Component Model

- Cleaner P&L:** Marketing costs (lead) separated from service costs (coach)
- Coach Incentive:** Clear reward for bringing their own leads
- Scalable Accounting:** Easy to adjust ratios as business evolves
- Tax Clarity:** Platform fee is unambiguous business income

## 2. Database Schema

### New Tables Required

sql

```

-- =====
-- REVENUE SPLIT CONFIGURATION
-- Run in Supabase SQL Editor
-- =====

-- 2.1 REVENUE SPLIT SETTINGS (Admin Configurable)
-- =====

CREATE TABLE IF NOT EXISTS revenue_split_config (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),

    -- Component percentages (must total 100)
    lead_cost_percent DECIMAL(5,2) NOT NULL DEFAULT 20.00,
    coach_cost_percent DECIMAL(5,2) NOT NULL DEFAULT 50.00,
    platform_fee_percent DECIMAL(5,2) NOT NULL DEFAULT 30.00,

    -- TDS configuration
    tds_rate_percent DECIMAL(5,2) NOT NULL DEFAULT 10.00, -- Section 194J
    tds_threshold_annual INTEGER NOT NULL DEFAULT 30000, -- ₹30,000/year

    -- Payout configuration
    payout_frequency TEXT NOT NULL DEFAULT 'monthly', -- 'monthly', 'per_session'
    payout_day_of_month INTEGER DEFAULT 7, -- Pay on 7th of each month

    -- Metadata
    effective_from TIMESTAMP WITH TIME ZONE NOT NULL DEFAULT NOW(),
    created_by TEXT,
    notes TEXT,
    is_active BOOLEAN DEFAULT true,

    -- Validation
    CONSTRAINT valid_percentages CHECK (
        lead_cost_percent + coach_cost_percent + platform_fee_percent = 100.00
    ),
    CONSTRAINT valid_tds CHECK (tds_rate_percent >= 0 AND tds_rate_percent <= 30),
    CONSTRAINT valid_payout_day CHECK (payout_day_of_month >= 1 AND payout_day_of_month <= 28)
);

-- Insert default configuration
INSERT INTO revenue_split_config (
    lead_cost_percent, coach_cost_percent, platform_fee_percent,
    tds_rate_percent, payout_frequency, created_by
) VALUES (20.00, 50.00, 30.00, 10.00, 'monthly', 'system')
ON CONFLICT DO NOTHING;

-- 2.2 ENROLLMENT REVENUE BREAKDOWN

```

-- Stores the split for each enrollment at time of payment

---

**CREATE TABLE IF NOT EXISTS** enrollment\_revenue (

id UUID **PRIMARY KEY DEFAULT** gen\_random\_uuid(),

enrollment\_id UUID **NOT NULL REFERENCES** enrollments(id) **ON DELETE CASCADE**,

-- Source tracking

lead\_source TEXT **NOT NULL**, -- 'yestoryd', 'coach', 'referral'

lead\_source\_coach\_id UUID **REFERENCES** coaches(id), -- If coach-sourced

-- Original amounts (before TDS)

total\_amount INTEGER **NOT NULL**, -- ₹5,999

lead\_cost\_amount INTEGER **NOT NULL**, -- ₹1,200

coach\_cost\_amount INTEGER **NOT NULL**, -- ₹3,000

platform\_fee\_amount INTEGER **NOT NULL**, -- ₹1,799

-- TDS calculation

tds\_applicable BOOLEAN **DEFAULT** true,

tds\_rate\_applied DECIMAL(5,2), -- 10%

tds\_amount INTEGER **DEFAULT** 0, -- ₹300 (on coach\_cost)

-- Net payable

net\_to\_coach INTEGER **NOT NULL**, -- ₹2,700 (after TDS)

net\_to\_lead\_source INTEGER **NOT NULL**, -- ₹1,200 or ₹0

net\_retained\_by\_platform INTEGER **NOT NULL**, -- Platform fee + TDS collected

-- Split config snapshot (for audit)

config\_snapshot JSONB **NOT NULL**, -- Copy of revenue\_split\_config at time

-- Status

status TEXT **DEFAULT** 'pending', -- 'pending', 'processing', 'completed'

created\_at TIMESTAMP WITH TIME ZONE **DEFAULT** NOW(),

UNIQUE(enrollment\_id)

);

-- Index for faster queries

**CREATE INDEX IF NOT EXISTS** idx\_enrollment\_revenue\_status **ON** enrollment\_revenue(status);

**CREATE INDEX IF NOT EXISTS** idx\_enrollment\_revenue\_lead\_source **ON** enrollment\_revenue(lead\_source);

-- 2.3 COACH PAYOUT SCHEDULE

-- Tracks monthly staggered payouts

---

**CREATE TABLE IF NOT EXISTS** coach\_payouts (

id UUID **PRIMARY KEY DEFAULT** gen\_random\_uuid(),

```
enrollment_revenue_id UUID NOT NULL REFERENCES enrollment_revenue(id),  
coach_id UUID NOT NULL REFERENCES coaches(id),
```

-- Payout details

```
payout_month INTEGER NOT NULL,          -- 1, 2, or 3  
payout_type TEXT NOT NULL,             -- 'coach_cost', 'lead_bonus'
```

-- Amounts

```
gross_amount INTEGER NOT NULL,          -- Before TDS  
tds_amount INTEGER DEFAULT 0,           -- TDS deducted  
net_amount INTEGER NOT NULL,            -- After TDS
```

-- Scheduling

```
scheduled_date DATE NOT NULL,          -- When this payout is due
```

-- Status tracking

```
status TEXT DEFAULT 'scheduled',        -- 'scheduled', 'processing', 'paid', 'failed'  
paid_at TIMESTAMP WITH TIME ZONE,  
payment_reference TEXT,                 -- UTR/Transaction ID  
payment_method TEXT,                   -- 'razorpay_payout', 'bank_transfer', 'manual'
```

-- Metadata

```
notes TEXT,  
created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),  
updated_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()  
);
```

-- Indexes

```
CREATE INDEX IF NOT EXISTS idx_coach_payouts_coach ON coach_payouts(coach_id);  
CREATE INDEX IF NOT EXISTS idx_coach_payouts_status ON coach_payouts(status);  
CREATE INDEX IF NOT EXISTS idx_coach_payouts_scheduled ON coach_payouts(scheduled_date);  
CREATE INDEX IF NOT EXISTS idx_coach_payouts_month ON coach_payouts(payout_month);
```

-- 2.4 TDS LEDGER

-- Track TDS deducted for compliance

```
=====
```

```
CREATE TABLE IF NOT EXISTS tds_ledger (  
id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
```

-- Deductee details

```
coach_id UUID NOT NULL REFERENCES coaches(id),  
coach_pan TEXT,                      -- PAN of coach
```

-- Financial year tracking

```
financial_year TEXT NOT NULL,          -- '2025-26'  
quarter TEXT NOT NULL,                -- 'Q1', 'Q2', 'Q3', 'Q4'
```

```

-- TDS details
section TEXT DEFAULT '194J',           -- TDS section
gross_amount INTEGER NOT NULL,          -- Amount on which TDS deducted
tds_rate DECIMAL(5,2) NOT NULL,
tds_amount INTEGER NOT NULL,

-- Payment reference
payout_id UUID REFERENCES coach_payouts(id),

-- Deposit status
deposited BOOLEAN DEFAULT false,
deposit_date DATE,
challan_number TEXT,

created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

-- Indexes
CREATE INDEX IF NOT EXISTS idx_tds_ledger_coach ON tds_ledger(coach_id);
CREATE INDEX IF NOT EXISTS idx_tds_ledger_fy ON tds_ledger(financial_year, quarter);
CREATE INDEX IF NOT EXISTS idx_tds_ledger_deposited ON tds_ledger(deposited);

-- 2.5 UPDATE COACHES TABLE
-- Add fields for payout
=====

ALTER TABLE coaches ADD COLUMN IF NOT EXISTS bank_account_number TEXT;
ALTER TABLE coaches ADD COLUMN IF NOT EXISTS bank_ifsc TEXT;
ALTER TABLE coaches ADD COLUMN IF NOT EXISTS bank_name TEXT;
ALTER TABLE coaches ADD COLUMN IF NOT EXISTS pan_number TEXT;
ALTER TABLE coaches ADD COLUMN IF NOT EXISTS gst_number TEXT;
ALTER TABLE coaches ADD COLUMN IF NOT EXISTS razorpay_contact_id TEXT; -- For Razorpay Payouts
ALTER TABLE coaches ADD COLUMN IF NOT EXISTS razorpay_fund_account_id TEXT;
ALTER TABLE coaches ADD COLUMN IF NOT EXISTS tds_cumulative_fy INTEGER DEFAULT 0; -- Running total this

-- 2.6 HELPER FUNCTIONS
=====

-- Function to get current revenue split config
CREATE OR REPLACE FUNCTION get_active_revenue_config()
RETURNS revenue_split_config AS $$
SELECT * FROM revenue_split_config
WHERE is_active = true
ORDER BY effective_from DESC
LIMIT 1;

```

```
$$ LANGUAGE SQL STABLE;
```

-- Function to calculate financial year

```
CREATE OR REPLACE FUNCTION get_financial_year(check_date DATE DEFAULT CURRENT_DATE)
RETURNS TEXT AS $$

BEGIN
    IF EXTRACT(MONTH FROM check_date) >= 4 THEN
        RETURN EXTRACT(YEAR FROM check_date)::TEXT || '-' ||
               (EXTRACT(YEAR FROM check_date) + 1)::TEXT;
    ELSE
        RETURN (EXTRACT(YEAR FROM check_date) - 1)::TEXT || '-' ||
               EXTRACT(YEAR FROM check_date)::TEXT;
    END IF;
END;

$$ LANGUAGE plpgsql IMMUTABLE;
```

-- Function to get quarter

```
CREATE OR REPLACE FUNCTION get_quarter(check_date DATE DEFAULT CURRENT_DATE)
RETURNS TEXT AS $$

DECLARE
    month_num INTEGER;
BEGIN
    month_num := EXTRACT(MONTH FROM check_date);
    IF month_num IN (4, 5, 6) THEN RETURN 'Q1';
    ELSIF month_num IN (7, 8, 9) THEN RETURN 'Q2';
    ELSIF month_num IN (10, 11, 12) THEN RETURN 'Q3';
    ELSE RETURN 'Q4';
    END IF;
END;

$$ LANGUAGE plpgsql IMMUTABLE;
```

## Updated enrollments table reference

The existing `enrollments` table will be linked via `enrollment_revenue.enrollment_id`.

## 3. Admin Interface Design

### 3.1 Revenue Settings Page (`/admin/settings/revenue`)

 Revenue Split Configuration	[Save Changes]
COMPONENT PERCENTAGES	

Lead Cost		Coach Cost		Platform Fee		
[ 20 ] %		[ 50 ] %		30 %		
			(auto-calculated)			
Goes to whoever		Goes to coach		Retained by		
sourced the lead		always		Yestoryd LLP		

⚠ Total must equal 100%. Platform Fee adjusts automatically.

## TDS CONFIGURATION

TDS Rate (Section 194J): [ 10 ] %

Annual Threshold: [ 30000 ] ₹

ℹ️ TDS deducted only on Coach Cost. Applied when coach's cumulative earnings exceed threshold in a financial year.

## PAYOUT SCHEDULE

Frequency:  Monthly (recommended)  Per Session

Payout Day: [ 7 ] th of each month

## PREVIEW (at ₹5,999)

Component	Yestoryd Lead	Coach Lead
Lead Cost	→ Yestoryd	→ Coach
₹1,200	₹1,200	₹1,200

Coach Cost	→ Coach	→ Coach		
₹3,000	₹2,700*	₹2,700*		
Platform Fee	→ Yestoryd	→ Yestoryd		
₹1,799	₹1,799	₹1,799		
TOTAL TO COACH	₹2,700	₹3,900		
TOTAL TO YEST.	₹3,299**	₹2,099**		

## 3.2 Coach Payouts Dashboard ([/admin/payouts](#))

## PAYOUT HISTORY

[\[Export CSV\]](#)

Date	Coach	Amount	TDS	UTR	Status		
Dec 7	Priya S.	₹4,500	₹450	UTR123456789	<input checked="" type="checkbox"/> Paid		
Dec 7	Meera K.	₹2,700	₹270	UTR123456790	<input checked="" type="checkbox"/> Paid		
Nov 7	Priya S.	₹3,600	₹360	UTR123456780	<input checked="" type="checkbox"/> Paid		

## 3.3 TDS Compliance Dashboard ([/admin/tds](#))

### TDS Compliance

FY 2025-26

## QUARTERLY SUMMARY

Quarter	TDS Deducted	Deposited	Pending	Due Date	Status	
Q1	₹12,000	₹12,000	₹0	Jul 7, 25	<input checked="" type="checkbox"/>	
Q2	₹18,500	₹18,500	₹0	Oct 7, 25	<input checked="" type="checkbox"/>	
Q3	₹24,200	₹0	₹24,200	Jan 7, 26		
Q4	₹0	-	-	Apr 30, 26	-	

## COACH-WISE TDS (FY 2025-26)

[\[Download 26Q\]](#)

Coach	PAN	Total Paid	TDS Ded.	TDS Rate		
Priya S.	ABCDE1234F	₹54,000	₹5,400	10%		
Meera K.	FGHIJ5678K	₹36,000	₹3,600	10%		
Rahul T.	PENDING	₹18,000	₹0	N/A*		

\* Below ₹30,000 threshold - TDS not applicable yet

 Rahul T. PAN missing - request before next payout

## 4. API Endpoints

### 4.1 Revenue Configuration APIs

typescript

```
// GET /api/admin/revenue-config  
// Returns active revenue split configuration  
  
// POST /api/admin/revenue-config  
// Updates revenue split configuration  
// Body: { lead_cost_percent, coach_cost_percent, tds_rate_percent, payout_frequency }  
  
// GET /api/admin/revenue-config/preview  
// Preview split calculation for given amount  
// Query: ?amount=5999&lead_source=coach
```

### 4.2 Payout APIs

typescript

```

// GET /api/admin/payouts
// List all payouts with filters
// Query: ?status=pending&month=2025-01&coach_id=xxx

// GET /api/admin/payouts/due
// Get payouts due for processing this month

// POST /api/admin/payouts/process
// Process selected payouts
// Body: { payout_ids: [...], payment_method: 'razorpay_payout' | 'manual' }

// POST /api/admin/payouts/[id]/mark-paid
// Manually mark payout as paid
// Body: { payment_reference: 'UTR123', notes: '...' }

// GET /api/admin/payouts/export
// Export payouts as CSV for bank upload
// Query: ?status=pending&format=hdfc|icici|generic

```

### 4.3 TDS APIs

typescript

```

// GET /api/admin/tds/summary
// TDS summary by quarter and financial year

// GET /api/admin/tds/coach/[id]
// Coach-specific TDS details

// POST /api/admin/tds/mark-deposited
// Mark TDS as deposited
// Body: { quarter: 'Q3', financial_year: '2025-26', challan_number: '...', deposit_date: '...' }

// GET /api/admin/tds/26q
// Generate 26Q data for filing
// Query: ?quarter=Q3&fy=2025-26

```

### 4.4 Enrollment Revenue Calculation

typescript

```
// POST /api/enrollment/calculate-revenue
// Called after successful payment to calculate and store revenue breakdown

interface CalculateRevenueRequest {
    enrollment_id: string;
    total_amount: number;
    lead_source: 'yestoryd' | 'coach';
    lead_source_coach_id?: string; // If coach-sourced
    coach_id: string;
}

interface CalculateRevenueResponse {
    enrollment_revenue_id: string;
    breakdown: {
        lead_cost: number;
        coach_cost: number;
        platform_fee: number;
        tds_amount: number;
        net_to_coach: number;
        net_to_lead_source: number;
    };
    payouts_scheduled: {
        month_1: { date: string; amount: number };
        month_2: { date: string; amount: number };
        month_3: { date: string; amount: number };
    };
}
```

## 5. Payout Processing Logic

### 5.1 On Enrollment (Payment Verified)

typescript

```

async function processEnrollmentRevenue(enrollmentId: string, paymentData: PaymentData) {
    // 1. Get active revenue config
    const config = await getActiveRevenueConfig();

    // 2. Calculate split
    const totalAmount = paymentData.amount;
    const leadCost = Math.round(totalAmount * config.lead_cost_percent / 100);
    const coachCost = Math.round(totalAmount * config.coach_cost_percent / 100);
    const platformFee = totalAmount - leadCost - coachCost;

    // 3. Determine TDS applicability
    const coach = await getCoach(paymentData.coach_id);
    const coachYTD = coach.tds_cumulative_fy + coachCost;
    const tdsApplicable = coachYTD > config.tds_threshold_annual;
    const tdsAmount = tdsApplicable ? Math.round(coachCost * config.tds_rate_percent / 100) : 0;

    // 4. Calculate net amounts
    const netToCoach = coachCost - tdsAmount;
    const netToLeadSource = paymentData.lead_source === 'coach' ? leadCost : 0;

    // 5. Store enrollment revenue
    const enrollmentRevenue = await createEnrollmentRevenue({
        enrollment_id: enrollmentId,
        lead_source: paymentData.lead_source,
        total_amount: totalAmount,
        lead_cost_amount: leadCost,
        coach_cost_amount: coachCost,
        platform_fee_amount: platformFee,
        tds_applicable: tdsApplicable,
        tds_rate_applied: tdsApplicable ? config.tds_rate_percent : null,
        tds_amount: tdsAmount,
        net_to_coach: netToCoach,
        net_to_lead_source: netToLeadSource,
        config_snapshot: config,
    });

    // 6. Create staggered payout schedule
    const payoutSchedule = calculatePayoutDates(config.payout_day_of_month);

    // Month 1, 2, 3 coach cost payouts
    for (let month = 1; month <= 3; month++) {
        const monthlyCoachCost = Math.round(coachCost / 3);
        const monthlyTds = Math.round(tdsAmount / 3);

        await createCoachPayout({
            enrollment_revenue_id: enrollmentRevenue.id,

```

```

coach_id: paymentData.coach_id,
payout_month: month,
payout_type: 'coach_cost',
gross_amount: monthlyCoachCost,
tds_amount: monthlyTds,
net_amount: monthlyCoachCost - monthlyTds,
scheduled_date: payoutSchedule[month - 1],
});

}

// If coach-sourced, also schedule lead bonus payouts
if(paymentData.lead_source === 'coach') {
  for(let month = 1; month <= 3; month++) {
    const monthlyLeadBonus = Math.round(leadCost / 3);
    const monthlyTds = Math.round(monthlyLeadBonus * config.tds_rate_percent / 100);

    await createCoachPayout({
      enrollment_revenue_id: enrollmentRevenue.id,
      coach_id: paymentData.lead_source_coach_id,
      payout_month: month,
      payout_type: 'lead_bonus',
      gross_amount: monthlyLeadBonus,
      tds_amount: tdsApplicable ? monthlyTds : 0,
      net_amount: monthlyLeadBonus - (tdsApplicable ? monthlyTds : 0),
      scheduled_date: payoutSchedule[month - 1],
    });
  }
}

// 7. Update coach cumulative TDS tracker
await updateCoachTdsCumulative(paymentData.coach_id, coachCost);

return enrollmentRevenue;
}

```

## 5.2 Monthly Payout Processing (Cron/Manual)

typescript

```
async function processMonthlyPayouts() {
  const today = new Date();

  // Get all payouts due up to today
  const duePayouts = await getDuePayouts(today);

  // Group by coach for batch processing
  const payoutsByCoach = groupBy(duePayouts, 'coach_id');

  const results = [];

  for (const [coachId, payouts] of Object.entries(payoutsByCoach)) {
    const coach = await getCoach(coachId);

    // Calculate total for this batch
    const totalNet = payouts.reduce((sum, p) => sum + p.net_amount, 0);
    const totalTds = payouts.reduce((sum, p) => sum + p.tds_amount, 0);

    // Option 1: Razorpay Payouts API
    if (coach.razorpay_fund_account_id) {
      const transfer = await razorpayPayouts.create({
        fund_account_id: coach.razorpay_fund_account_id,
        amount: totalNet * 100, // paise
        currency: 'INR',
        mode: 'IMPS',
        purpose: 'payout',
        notes: { coach_id: coachId, month: today.toISOString().slice(0, 7) },
      });
    }

    // Update payout status
    for (const payout of payouts) {
      await updatePayoutStatus(payout.id, {
        status: 'paid',
        paid_at: new Date(),
        payment_reference: transfer.utr,
        payment_method: 'razorpay_payout',
      });
    }

    // Record in TDS ledger
    if (payout.tds_amount > 0) {
      await createTdsLedgerEntry({
        coach_id: coachId,
        financial_year: getFinancialYear(),
        quarter: getQuarter(),
        section: '194J',
        gross_amount: payout.gross_amount,
      });
    }
  }
}
```

```

        tds_rate: payout.tds_rate || 10,
        tds_amount: payout.tds_amount,
        payout_id: payout.id,
    });
}

}

results.push({ coachId, status: 'success', amount: totalNet, utr: transfer.utr });
}

// Option 2: Manual (generate bank file)
else {
    results.push({ coachId, status: 'manual_required', amount: totalNet, bankDetails: coach });
}
}

return results;
}

```

## 6. LLP Registration Guide

### Why LLP for Yestoryd

Factor	Proprietorship	LLP	Pvt Ltd
Tax on ₹10L profit	₹3L (30% slab)	₹2.4L (optimized)*	₹2.5L + dividend tax
Liability	Unlimited personal	Limited	Limited
Compliance cost	₹5K/year	₹20K/year	₹50K/year
Setup cost	₹0	₹8K	₹15K
Investor-ready	No	Convertible	Yes

\*With proper remuneration structure to Rucha

### LLP Registration Steps

#### Step 1: Digital Signature Certificate (DSC)

- Apply for Class 3 DSC for both partners (Amit + Rucha)
- Cost: ₹1,500 per person
- Time: 1-2 days

- Provider: eMudhra, Capricorn, Sify

## **Step 2: Designated Partner Identification Number (DPIN)**

- Apply on MCA portal
- Cost: ₹100 per partner
- Time: 2-3 days

## **Step 3: Name Reservation**

- Apply for "Yestoryd LLP" on MCA
- Alternatives: "Yestoryd Education LLP", "Yestoryd Learning LLP"
- Cost: ₹200
- Time: 2-5 days

## **Step 4: LLP Agreement**

- Draft LLP Agreement covering:
  - Capital contribution (can be ₹10,000 each minimum)
  - Profit sharing ratio (e.g., 50:50 or as decided)
  - Remuneration to working partner (Rucha)
  - Decision-making authority
- Get it notarized
- Cost: ₹1,000-2,000

## **Step 5: Incorporation Filing**

- File Form FiLLiP on MCA
- Attach: LLP Agreement, Address proof, Partner ID proofs
- Cost: Government fees ~₹1,500
- Time: 5-7 days

## **Step 6: PAN & TAN**

- Apply immediately after incorporation
- Required for opening bank account and TDS compliance
- Cost: ₹0 (included)
- Time: 7-10 days

## Total LLP Setup

Item	Cost
2x DSC	₹3,000
2x DPIN	₹200
Name Reservation	₹200
Government Fees	₹1,500
Professional Fees (CA)	₹3,000
<b>Total</b>	<b>₹7,900</b>
<b>Time</b>	<b>15-20 days</b>

## LLP Tax Optimization Strategy

Scenario: ₹10L Annual Profit

LLP TAX STRUCTURE	
Gross Profit (before remuneration)	₹10,00,000
(-) Remuneration to Rucha (Designated Partner doing day-to-day work) [Allowed: Higher of ₹3L or 60% of book profit]	₹ 6,00,000
(=) LLP Taxable Profit	₹ 4,00,000
TAX CALCULATION:	
LLP Tax (30% of ₹4L)	₹ 1,20,000
Rucha's Personal Tax: - ₹6L remuneration (say 20% effective)	₹ 1,20,000
TOTAL TAX	₹ 2,40,000
COMPARISON:	
As Proprietorship (₹10L @ 30%)	₹ 3,00,000
As LLP (optimized)	₹ 2,40,000
ANNUAL SAVINGS	₹ 60,000

**Note:** Savings increase as profit grows. At ₹25L profit, savings can be ₹1.5L+.

## 7. TDS Compliance Framework

### TDS Section 194J - Professional Fees

Parameter	Requirement
Section	194J (Fees for Professional Services)

Parameter	Requirement
<b>Rate</b>	10%
<b>Threshold</b>	₹30,000 per financial year per coach
<b>Deposit Due</b>	7th of following month
<b>Return</b>	Quarterly (26Q)

## Quarterly Calendar

Quarter	Period	TDS Deposit By	26Q Filing By
Q1	Apr-Jun	7th Jul	31st Jul
Q2	Jul-Sep	7th Oct	31st Oct
Q3	Oct-Dec	7th Jan	31st Jan
Q4	Jan-Mar	30th Apr*	31st May

\*Q4 has extended deadline

## Compliance Checklist

### Monthly:

- Deduct TDS on coach payouts (if above threshold)
- Maintain deduction records in TDS ledger
- Deposit TDS by 7th of next month (via NSDL/TRACES)

### Quarterly:

- Reconcile TDS deducted vs deposited
- File Form 26Q on TRACES
- Issue Form 16A to coaches

### Annually:

- Reset coach cumulative counters (April 1)
- Verify all 26Q filings submitted
- Ensure all Form 16A issued

## Coach Onboarding Checklist (for TDS)

Before first payout, collect from each coach:

- PAN card copy
  - Bank account details (for payout)
  - Cancelled cheque or passbook copy
  - Address proof
  - GST number (if registered - not required for most coaches)
- 

## 8. CA Selection Criteria

### What You Need from a CA

Service	Importance	Notes
LLP Registration	High	One-time setup
GST Filing	High	Monthly returns
TDS Compliance	Critical	Quarterly 26Q filing, monthly deposits
Annual ITR (LLP)	High	Form ITR-5
Annual ITR (Personal)	Medium	Rucha's return with remuneration
Book-keeping	Optional	Can use software like Zoho Books
Advisory	Medium	Tax planning, structure optimization

### Questions to Ask Potential CAs

1. **Experience with EdTech/coaching businesses?** (Understanding of coach payouts, split models)
2. **Familiar with LLP remuneration rules?** (Section 40(b) limits)
3. **Handle TDS compliance in-house?** (26Q filing, TRACES)
4. **Can you set up Zoho Books/Tally?** (For your own tracking)
5. **Response time for queries?** (Important for payment gateway issues)
6. **Fee structure?** (Monthly retainer vs per-task)

### Recommended CA Profile

Look for:

- **Firm with 2-5 partners** (not too small, not too large)

- **Age 30-45** (tech-savvy, accessible via WhatsApp)
- **Has startup/D2C clients** (understands fast-moving businesses)
- **Based in your city** (for physical meetings if needed, though optional)

## Expected CA Costs

Service	Monthly	Annual
GST Filing (GSTR-1, 3B)	₹1,500	₹18,000
TDS (26Q, deposits)	₹1,000	₹12,000
LLP Annual Return	-	₹5,000
LLP ITR Filing	-	₹5,000
Rucha's Personal ITR	-	₹3,000
Ad-hoc Advisory	₹500	₹6,000
<b>Total</b>	<b>~₹3,000/month</b>	<b>₹49,000/year</b>

## How to Find a Good CA

### Option 1: Referrals

- Ask founders in your network
- Check IndiaMART/JustDial reviews (filter noise)

### Option 2: Platforms

- ClearTax (has CA network)
- Zoho Books (has CA partners)
- LegalZoom/Vakilsearch (for LLP registration, can continue)

### Option 3: Direct

- ICAI member search: [https://www.icai.org/new\\_post.html?post\\_id=4278](https://www.icai.org/new_post.html?post_id=4278)
- Filter by city, firm type

## Interview 2-3 CAs Before Deciding

**Template message:**

Hi [CA Name],

I'm looking for a CA for my EdTech startup "Yestoryd" - an AI-powered reading coaching platform for children.

Current structure: Proprietorship (planning to convert to LLP)

Expected revenue: ₹6-7L/month initially

Key needs:

1. LLP registration and compliance
2. Monthly GST filing
3. Quarterly TDS (194J) for coach payouts
4. Tax planning/optimization

We have a tech-first business with multiple coaches paid monthly via bank transfer. Need someone comfortable with digital tools and responsive on WhatsApp.

Could we schedule a 30-min call to discuss? Happy to share more details.

Thanks,

Amit

## 9. Implementation Roadmap

### Phase 1: LLP Setup (Week 1-3)

Day	Action	Owner
1-2	Interview 2-3 CAs, finalize one	Amit
3-5	Apply for DSC (both partners)	CA
5-7	Apply for DPIN	CA
7-10	LLP Name reservation	CA
10-12	Draft & notarize LLP Agreement	CA
12-15	File FiLLiP, incorporate LLP	CA
15-20	Receive Certificate, apply PAN/TAN	CA
20-25	Open LLP bank account	Amit + Rucha
25-30	GST registration	CA

## Phase 2: Database & Backend (Week 2-4)

Day	Action	Owner
1	Run SQL schema in Supabase	Dev
2-3	Build revenue config API	Dev
4-5	Build payout scheduling logic	Dev
6-7	Build TDS calculation logic	Dev
8-10	Integrate with enrollment webhook	Dev
11-12	Test with sample enrollments	Dev

## Phase 3: Admin Interface (Week 3-5)

Day	Action	Owner
1-3	Revenue Settings page	Dev
4-6	Coach Payouts dashboard	Dev
7-8	TDS Compliance dashboard	Dev
9-10	Payout processing workflow	Dev
11-12	CSV export for bank upload	Dev

## Phase 4: Go-Live (Week 5-6)

Day	Action	Owner
1-2	Update Razorpay to LLP bank account	Amit
3	Configure revenue split (20-50-30)	Admin portal
4	Collect coach bank details & PAN	Operations
5	Test end-to-end with real enrollment	All
6	Go live!	

## Phase 5: Cashfree Integration (Post LLP, Week 7-8)

Day	Action	Owner
1	Sign up Cashfree with LLP details	Amit
2-3	Build Cashfree payment integration	Dev
4	Build Cashfree payouts integration	Dev
5	Test as backup gateway	Dev
6	Document failover procedure	Dev

## Summary: Your Action Items

### Immediate (This Week)

1. **Finalize CA** - Interview 2-3, select one
2. **Start LLP process** - DSC application
3. **Review this document** - Confirm 20-50-30 split is correct

### Next 2 Weeks

4. **Complete LLP registration**
5. **Open LLP bank account**
6. **GST registration**

### Next 4 Weeks

7. **Database schema deployment**
8. **Admin interface development**
9. **Update payment gateway to LLP account**
10. **Collect coach PAN/bank details**

---

**Questions?** Let me know if you want me to:

- Model different percentage splits

- Create the actual React components for admin pages
- Draft the LLP Agreement points
- Prepare coach onboarding checklist

This document will be your north star for the next 6-8 weeks of implementation.