Multi-Organization Payment, Donation & Product Tracking System

This document explains **how to track multiple products, multiple donation campaigns, across many organizations**, and shows **clear diagrams + data relationships** suitable for **FYP documentation and real-world implementation**.

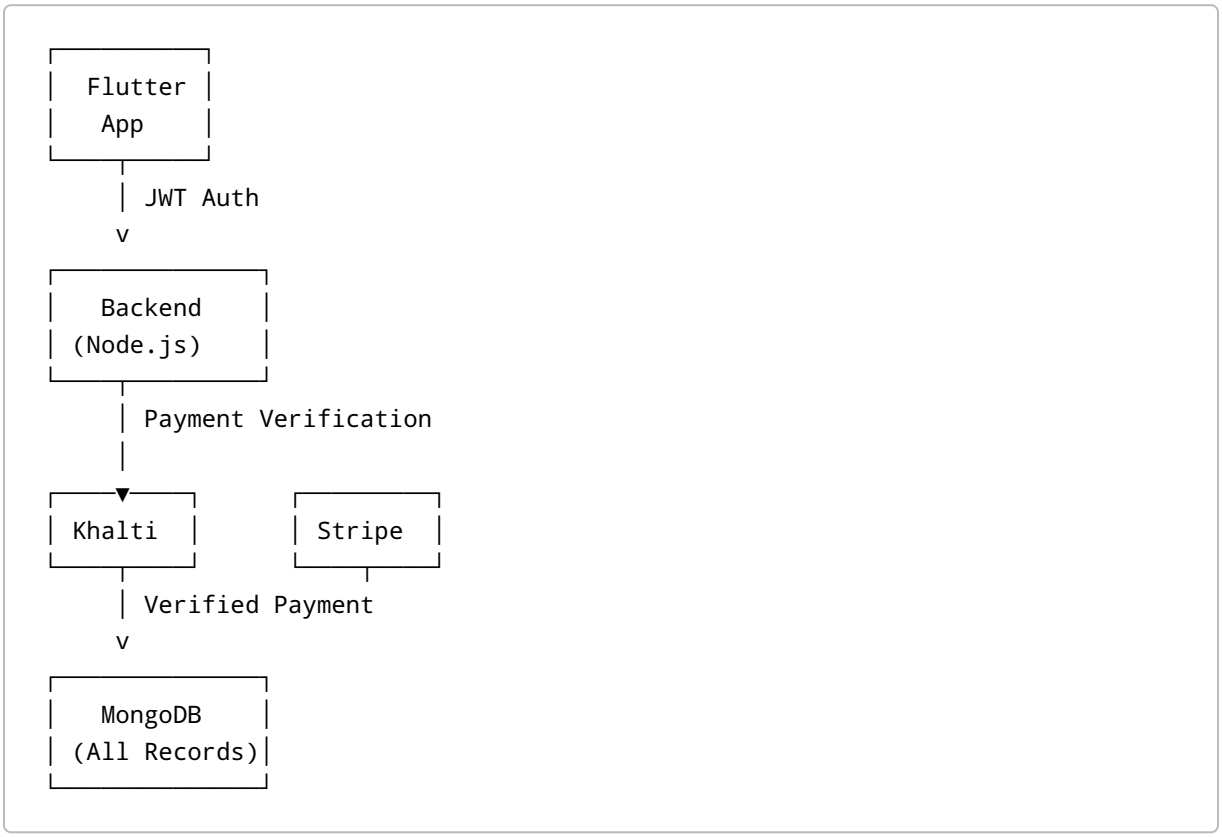
---

1. Core Problem You Are Solving

✔️Many **Organizations** ✔️Each organization has: - Multiple **Donation Campaigns** - Multiple **Products** ✔️ Many **Users** can: - Donate to any campaign - Buy products from any organization ✔️Payments can be done via **Khalti or Stripe** ✔️You must maintain: - User-wise history - Organization-wise collection - Campaign-wise donation totals - Product-wise sales

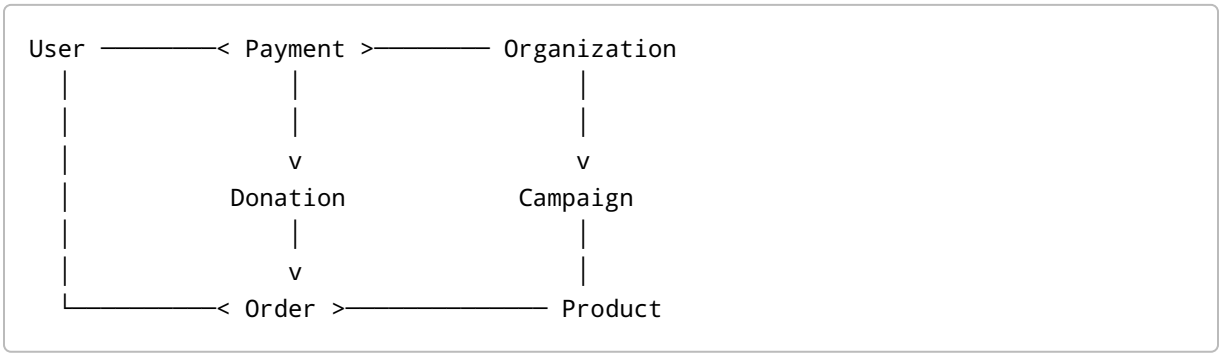
---

2. High-Level System Architecture Diagram



---

3. Entity Relationship Diagram (ERD – Textual)



Legend: - `A > B` → One-to-Many

---

4. Detailed Data Models & Relationships

4.1 User

```
User
 ├── _id
 ├── name
 ├── email
 ├── role (user | org | admin)
```

4.2 Organization

```
Organization
 ├── _id
 ├── name
 ├── totalDonationsCollected
 ├── totalProductRevenue
```

✔️Updated automatically from payments

4.3 Campaign (Donation Campaign)

```
Campaign
 ├── _id
 ├── organizationId
```

```
  ├── title
  ├── targetAmount
  ├── collectedAmount
  ├── status (active | completed)
```

✔️One organization → many campaigns

4.4 Product

```
Product
  ├── _id
  ├── organizationId
  ├── name
  ├── price
  ├── stock
```

✔️One organization → many products

4.5 Payment (MOST IMPORTANT)

This is the **single source of truth**.

```
Payment
  ├── _id
  ├── userId
  ├── organizationId
  ├── gateway (khalti | stripe)
  ├── purpose (donation | purchase)
  ├── amount
  ├── status (success | failed)
  ├── transactionId
  ├── createdAt
```

✔️Every donation & purchase MUST have a payment

4.6 Donation

```
Donation
  ├── _id
  ├── userId
```

```
  ├── organizationId
  ├── campaignId
  ├── paymentId
  ├── amount
  ├── createdAt
```

✔️Used for campaign tracking

4.7 Order (Product Purchase)

```
Order
  ├── _id
  ├── userId
  ├── organizationId
  ├── products[]
  ├── totalAmount
  ├── paymentId
  ├── status
```

✔️Used for product revenue tracking

---

5. User Donation Flow (Sequence Diagram – Detailed)

```
User
  │
  │ Select Campaign
  │──────────────►
  │
  │ Choose Amount + Payment Gateway
  │──────────────► Flutter App
  │
  │ Pay via Khalti/Stripe SDK
  │──────────────► Gateway
  │
  │ Payment Token
  │◄──────────────
  │
  │ POST /payments/verify (JWT)
  │──────────────► Backend
  │
  │ Verify with Gateway
  │──────────────► Khalti / Stripe
```

```
  │
  │ Success Response
  │◀──────────────
  │
  │ Save Payment + Donation
  │ Update Campaign Collected Amount
  │ Update Organization Total
  │
  │◀────────────── Success
```

---

6. Product Purchase Flow (Sequence Diagram)

```
User
  │
  │ Select Products
  │──────────────▶ Flutter App
  │
  │ Checkout
  │──────────────▶ Gateway
  │
  │ Payment Token
  │◀──────────────
  │
  │ POST /payments/verify
  │──────────────▶ Backend
  │
  │ Verify Payment
  │
  │ Save Payment + Order
  │ Reduce Product Stock
  │ Update Organization Revenue
  │
  │◀────────────── Order Confirmed
```

---

7. How History is Tracked (KEY EXPLANATION)

User History

- Query by `userId`

```
Payments → Donations → Orders
```

Organization History

- Query by `organizationId`

```
Payments → Campaigns → Products
```

Campaign History

- Query by `campaignId`

```
All donations linked to campaign
```

---

8. Suggested API Endpoints (Clean & Scalable)

```
POST   /payments/verify
GET    /users/me/donations
GET    /users/me/orders
GET    /organizations/:id/campaigns
GET    /organizations/:id/donations
GET    /organizations/:id/sales
```

---

9. Strong FYP-Level Suggestions (EXTRA MARKS)

✔️Unified payment table (very important) ✔️Do NOT store amount from frontend blindly ✔️Keep raw gateway response ✔️Anonymous donation option ✔️Campaign auto-close when target reached ✔️Admin analytics dashboard

---

10. Next Steps (I Can Help You With)

- draw.io **XML diagrams (ready to import)**
- ER diagram image
- Database indexes & optimization
- Stripe webhook full implementation
- FYP documentation (chapter-wise)

Just tell me what you want next.