## ****Thunglish Workflow Diagram****

flowchart LR

%% User Action

U[User Payment / Order] --> P[Payment Gateway]

%% Your Module Workflow

P --> F1[Flagged Transaction?]

F1 -- Yes --> A1[Admin Dashboard: Review Pending Flags]

A1 --> D1[Admin Resolves / Clears]

D1 --> N1[Notify User (Email/SMS/Push)]

D1 --> L1[Audit Log Entry]

F1 -- No --> C1[Transaction Completed]

%% Flipkart/Amazon Workflow

P --> F2[Fraud Detection System]

F2 -- Flagged --> A2[Backend Admin / Support Team Review]

A2 --> R2[Approve / Cancel / Refund]

R2 --> N2[Notify User (App / Email / SMS)]

R2 --> L2[Audit Logs / Compliance Records]

F2 -- Not Flagged --> C2[Order Completed / Shipped]

%% Legends

style F1 fill:#ffdddd,stroke:#ff0000

style F2 fill:#ffdddd,stroke:#ff0000

style A1 fill:#ddffdd,stroke:#00aa00

style A2 fill:#ddffdd,stroke:#00aa00

style D1 fill:#ddddff,stroke:#0000ff

style R2 fill:#ddddff,stroke:#0000ff

### ****Thunglish Explanation of Diagram****

1. **User Payment / Order**
   * User flipkart-la order place pannum or unga module-la transaction start.
2. **Flagged Transaction Check**
   * Your module → suspicious transaction detect pannum.
   * Flipkart → fraud detection system automatically detect pannum.
3. **Admin / Support Review**
   * Your module → admin dashboard-la pending flags review.
   * Flipkart → backend support/admin team review.
4. **Resolution / Notification**
   * Your module → admin resolves → notify user → log in audit\_logs.
   * Flipkart → approve/cancel/refund → notify user → logs maintained for compliance.
5. **Normal Transaction**
   * Both systems → transaction/order completed if not flagged.

💡 **Key Thunglish Takeaway**:

unga FastAPI module workflow **mini Flipkart backend workflow** polave:

* Suspicious detect
* Admin review & dispute resolution
* Notifications to user
* Audit logs for compliance

**Flowchart Explanation (Thunglish)**

**1. User Payment / Order**

U[User Payment / Order] --> P[Payment Gateway]

* **U:** User e.g., Flipkart-la order place pannum or unga app-la payment start pannum.
* **P:** Payment gateway (Razorpay / Stripe / PayPal etc.) transaction process pannum.

**2. Your Module Workflow**

P --> F1[Flagged Transaction?]

* **F1:** FastAPI module check pannum “Is this transaction suspicious?”
* Criteria: unusual amount, fraud risk, anomaly

**2.1 If Yes (Suspicious Transaction)**

F1 -- Yes --> A1[Admin Dashboard: Review Pending Flags]

A1 --> D1[Admin Resolves / Clears]

D1 --> N1[Notify User (Email/SMS/Push)]

D1 --> L1[Audit Log Entry]

* **A1:** Admin dashboard-la flagged transaction list show aagum
* **D1:** Admin review panni transaction **resolve / clear** pannum
* **N1:** User-ku notification send pannum (Email, SMS, Push)
* **L1:** Action log save aagum (audit\_logs table)

**2.2 If No (Transaction Normal)**

F1 -- No --> C1[Transaction Completed]

* Normal transaction → directly complete aagum.

**3. Flipkart / Amazon Workflow**

P --> F2[Fraud Detection System]

* **F2:** Flipkart backend fraud detection system automatic-a suspicious order detect pannum

**3.1 If Flagged**

F2 -- Flagged --> A2[Backend Admin / Support Team Review]

A2 --> R2[Approve / Cancel / Refund]

R2 --> N2[Notify User (App / Email / SMS)]

R2 --> L2[Audit Logs / Compliance Records]

* **A2:** Admin/support team review pannum flagged order
* **R2:** Approve / cancel / refund decide pannum
* **N2:** User-ku notify pannum
* **L2:** Audit log maintain pannum for compliance

**3.2 If Not Flagged**

F2 -- Not Flagged --> C2[Order Completed / Shipped]

* Normal orders → directly complete & ship pannum

**4. Color Legend**

| **Color** | **Meaning** |
| --- | --- |
| Red (F1/F2) | Suspicious / flagged transaction |
| Green (A1/A2) | Admin / support review |
| Blue (D1/R2) | Resolution / action taken |

**5. Thunglish Summary**

1. **User Payment → Payment Gateway**
2. **Check Transaction Suspicious?**
   * **Yes → Admin Review → Resolve → Notify → Log**
   * **No → Complete Transaction**
3. **Real-world Example:**
   * Flipkart / Amazon same principle
   * Fraud detection system flag pannum → admin review → approve/refund → notify → audit logs
4. **Difference:**
   * Your module → small-scale FastAPI backend
   * Flipkart → large-scale distributed backend with ML fraud detection

💡 **Key Point:**

Ungal FastAPI Payment Oversight Module workflow **mini Flipkart backend** pola work pannum:

* Suspicious detect
* Admin review & dispute resolution
* Notifications to user
* Audit logs maintain