**Dance Registration Portal — Workflows and Operations (User + Admin)**

Plain-language, diagram-like layouts optimized for Pages export (no code blocks).

**Legend**

• Green = success

• Orange = pending/action required

• Red = error

**1) Student Workflow (End‑to‑End)**

**1.1 Registration Journey**

**Discover & Select**

1 Open link or scan QR (shared by instructor).

2 Browse available dance series (cards with schedule, price, capacity).

3 Tap “Register Now” on the desired series.

**Fill & Submit**

1 Fill contact info (crew practice switches Instagram to Full Name field).

2 Choose payment option when applicable (full series / per class).

3 Submit. Status is now pending.

**Payment**

1 Mobile: Open Venmo app with amount + note prefilled.

2 Desktop: Scan QR or send manually to instructor Venmo with note.

3 Tap “I’ve Sent the Payment”.

**Confirmation**

1 See on‑screen confirmation summary.

2 Admin will review and mark payment complete.

3 When completed, a confirmation email is sent (if enabled).

**Payment Note Example:** Dance Registration #123 — Dreamers Crew Practice — Sept 12

**2) Admin Operations (Lifecycle)**

**2.1 Create Dance Series & Open Registration**

**Steps**

1 Log in → go to **Courses**.

2 Click **Create Dance Series**.

3 Add **Slot**: capacity, time(s), location.

4 Pricing:

◦ Dance Series: full package + per class

◦ Crew Practice: drop‑in price + practice\_date

5 Save course; go to **Settings** and toggle **Registration Open = ON**.

**Result**

• Course appears on student portal.

• Capacity and price derive from the defined slot(s).

*• schedule\_info* computed from slots for consistent UI.

**2.2 Track Students & Confirm Payments**

**Track**

1 Go to **Registrations**.

2 Filter by course and/or status.

3 Use CSV export for roster consolidation.

**Confirm**

1 When Venmo received, click **Quick Confirm** or **Mark Paid**.

2 Status becomes completed.

3 System sends email confirmation (if enabled).

**Missing Contact?** Use **Assign Student** on a row to link email/name to a registration.

**2.3 Consolidation After Closing Registration**

**Close & Export**

1 Settings → **Registration Open = OFF**.

2 Registrations → filter target course → **Export CSV**.

3 Reconcile pending vs completed; follow up as needed.

**Optional Analytics (SQL)**

SELECT c.name,

SUM(CASE WHEN r.payment\_status='completed' THEN r.payment\_amount ELSE 0 END) AS revenue

FROM registrations r

JOIN courses c ON c.id = r.course\_id

WHERE c.name ILIKE 'Dreamers Crew Practice - Sept 12'

GROUP BY c.name;

**2.4 Reset Between Series**

**Quick Actions**

**• Reset Data (Keep Active Course):** deletes all registrations; deactivates others.

**• Reset & Delete Other Courses:** deletes registrations and other courses.

**Result**

• Dashboard shows $0 revenue and 0 registrations.

• Kept course remains active and ready for next cycle.

**2.5 Fix Wrong or Incomplete Registrations**

**Assign or Delete**

• Assign Student to fill missing contact.

• Delete incorrect entry (via SQL until UI delete is added):  
DELETE FROM registrations WHERE id = 123;

**Best Practices**

• Double‑check IDs before running DELETE.

• Export CSV snapshots prior to bulk changes.

**3) Admin “How Do I” (Cheat‑Sheet)**

**Goal**

**Steps**

Track students for a series

Registrations → filter by course → export CSV

Reconcile payments

Filter by *pending*, confirm received payments; run revenue SQL if needed

Query production DB

Railway → Database → Connect → external DATABASE\_URL → use psql/TablePlus

Reset data

Quick Actions → choose reset option; confirm dialog; verify dashboard

**Missing Contact Report:**

SELECT r.id AS registration\_id, c.name AS course\_name, s.email, s.first\_name, s.last\_name

FROM registrations r

LEFT JOIN students s ON s.id = r.student\_id

LEFT JOIN courses c ON c.id = r.course\_id

WHERE s.id IS NULL

OR COALESCE(TRIM(s.email),'')=''

OR (COALESCE(TRIM(s.first\_name),'')='' AND COALESCE(TRIM(s.last\_name),'')='');

**4) Email & Payment Flow (Readable)**

1 Admin confirms payment in UI.

2 Server updates registration to *completed*.

3 Server computes schedule\_info from slots and prepares email payload.

4 Email sent via Nodemailer. Failures are logged and do not block the update.

**5) Quick Start (New Admins)**

1 Create series + slot (capacity/time/location/pricing).

2 Settings → Registration Open = ON.

3 Share link or QR; watch registrations.

4 Confirm payments; emails will send if enabled.

5 At cutoff: Registration OFF → export CSV → finalize reconciliation.

6 Reset using Quick Actions before next series.